SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
Institute of Information Engineering, Automation, and Mathematics
Faculty of Chemical and Food Technology

Full Papers

18TH INTERNATIONAL CONFERENCE ON PROCESS CONTROL '11

June 14 – 17, 2011, Tatranská Lomnica, Slovakia

Sponsors:

EPP Group in the European Parliament

Honeywell

invensys Operations Management

Media Partners:

ProCS PROCESS CONTROL SYSTEMS

AUTOMA

CHEMagazín Časopis pro chemicko-technologickou a laboratorní praxi

AT&P JOURNAL

TECHNIKA V PRAXI

Co-organisers:

Department of Process Control, Faculty of Electrical Engineering and Informatics
University of Pardubice

Slovak Society for Cybernetics and Informatics
National Member Organisation of IFAC

http://www.kirp.chtf.stuba.sk/pc11

ISBN 978-80-227-3517-9

***The aim of the conference** is to exchange the recent advances and experience in the various areas of Control Theory between the researchers from industry, research institutes, project organisations, academies of sciences, and universities.*

***The program of the conference will be focused on** all aspects of Control and Systems, and ranges from fundamental research to applications in process control. Topics of interest include linear and non-linear control, optimisation, robust, adaptive and intelligent control, identification, modelling and simulations, real-time systems, new trends in application of industrial computer control, and education of qualified experts.*

INTERNATIONAL PROGRAM COMMITTEE

Chairman: Fikar M. (SK)

Albertos P.	(ESP)	Krokavec D.	(SK)
Bars R.	(HU)	Kučera V.	(CZ)
Bobál V.	(CZ)	Latifi M. A.	(F)
Dostál P.	(CZ)	Masár I.	(D)
Gerke M.	(D)	Mészáros A.	(SK)
Haber R.	(D)	Mikleš J.	(SK)
Henrion D.	(F)	Ogonowski Z.	(PL)
Hippe P.	(D)	Prokop R.	(CZ)
Huba M.	(SK)	Veselý V.	(SK)
Hulkó G.	(SK)	Strmčnik S.	(SLO)
Jørgensen S. B.	(DK)	Šebek M.	(CZ)
Jurišica L.	(SK)	Taufer I.	(CZ)
Koštial I.	(SK)	Vrancic D.	(SLO)
Kozák Š.	(SK)	Zítek P.	(CZ)

NATIONAL ORGANISING COMMITTEE

Chairman: Kvasnica M. (SK)

Čirka Ľ.	(SK)	Karšaiová M.	(SK)
Fikar M.	(SK)	Taufer I.	(CZ)
Kalmárová A.	(SK)	Vagač S.	(SK)

EDITORS

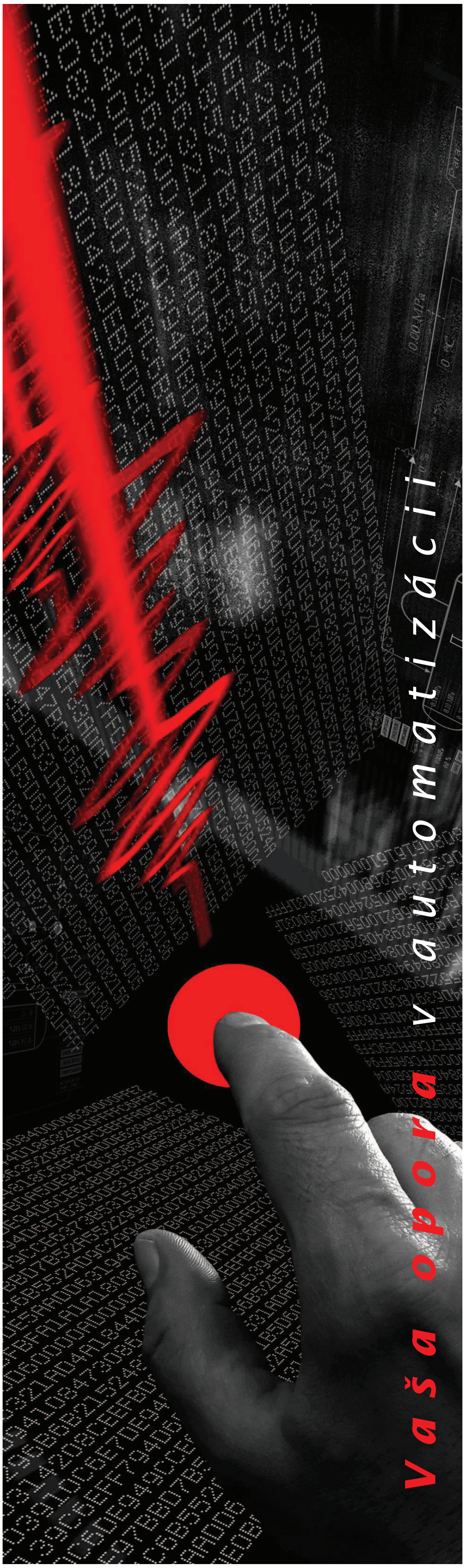
Fikar M.
Kvasnica M.

EPP Group
in the European Parliament

The 18th Conference on Process Control is sponsored by

Alajos Mészáros

from the Europeans People's Party Group in the European Parliament


Vaša opora v automatizácii

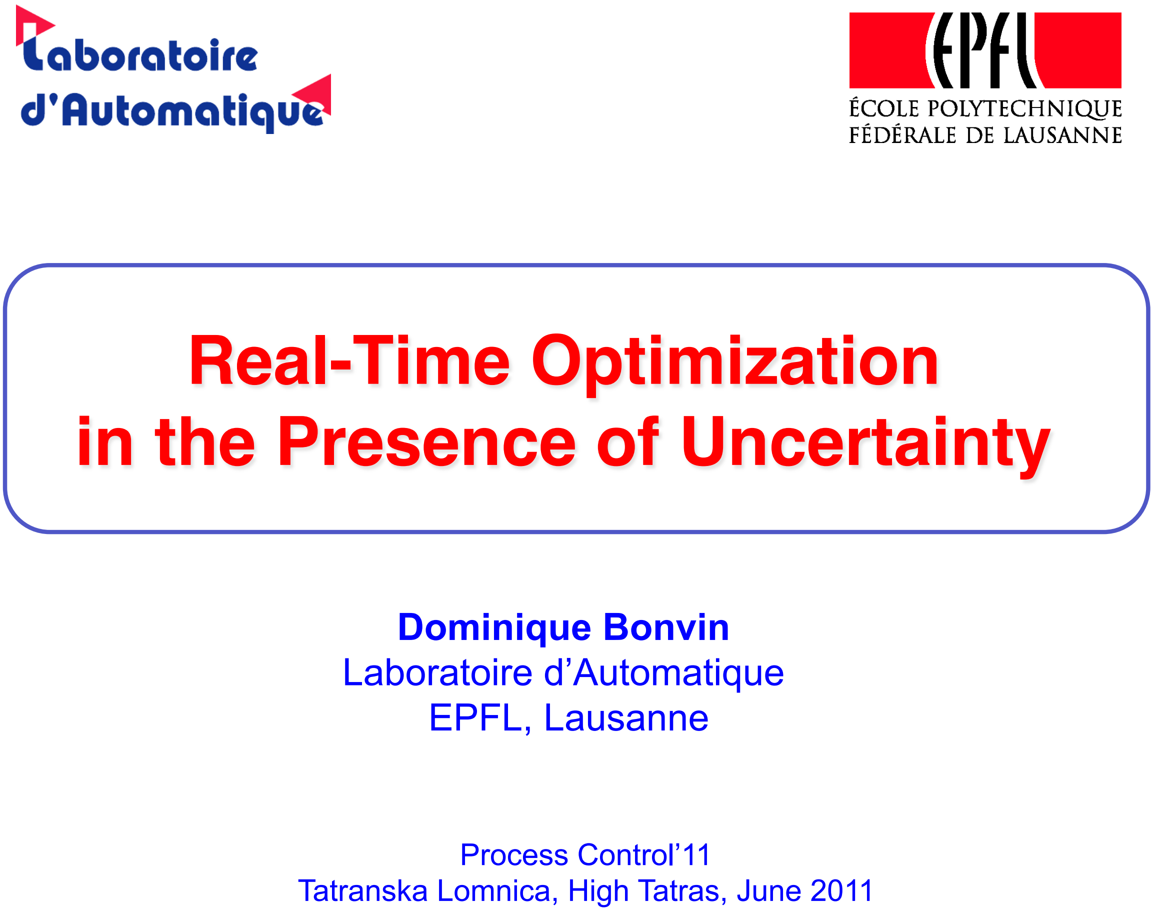
Program

Tuesday

Wednesday

Thursday

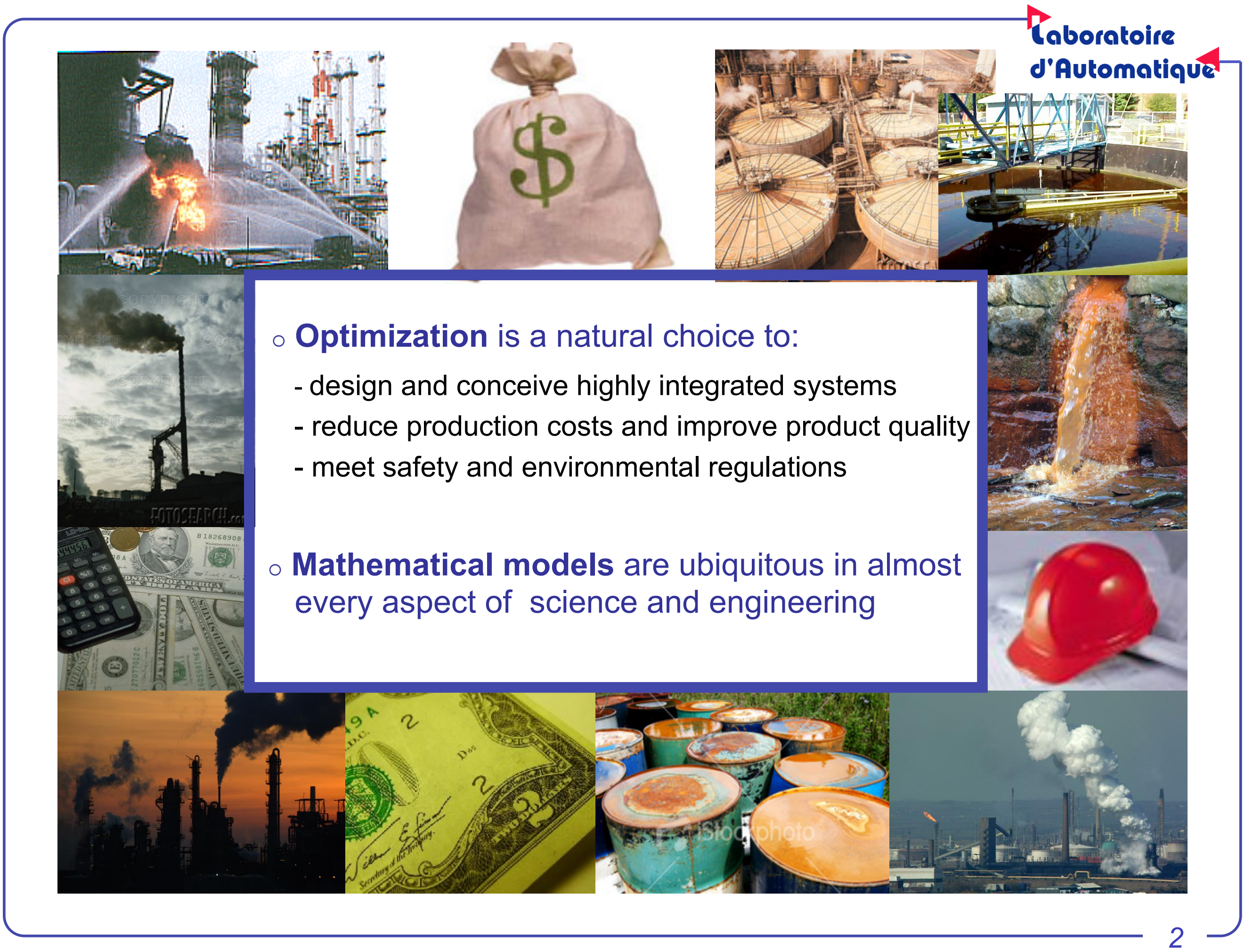
Friday



Real-Time Optimization in the Presence of Uncertainty

Dominique Bonvin
Laboratoire d'Automatique
EPFL, Lausanne

Process Control'11
Tatranska Lomnica, High Tatras, June 2011

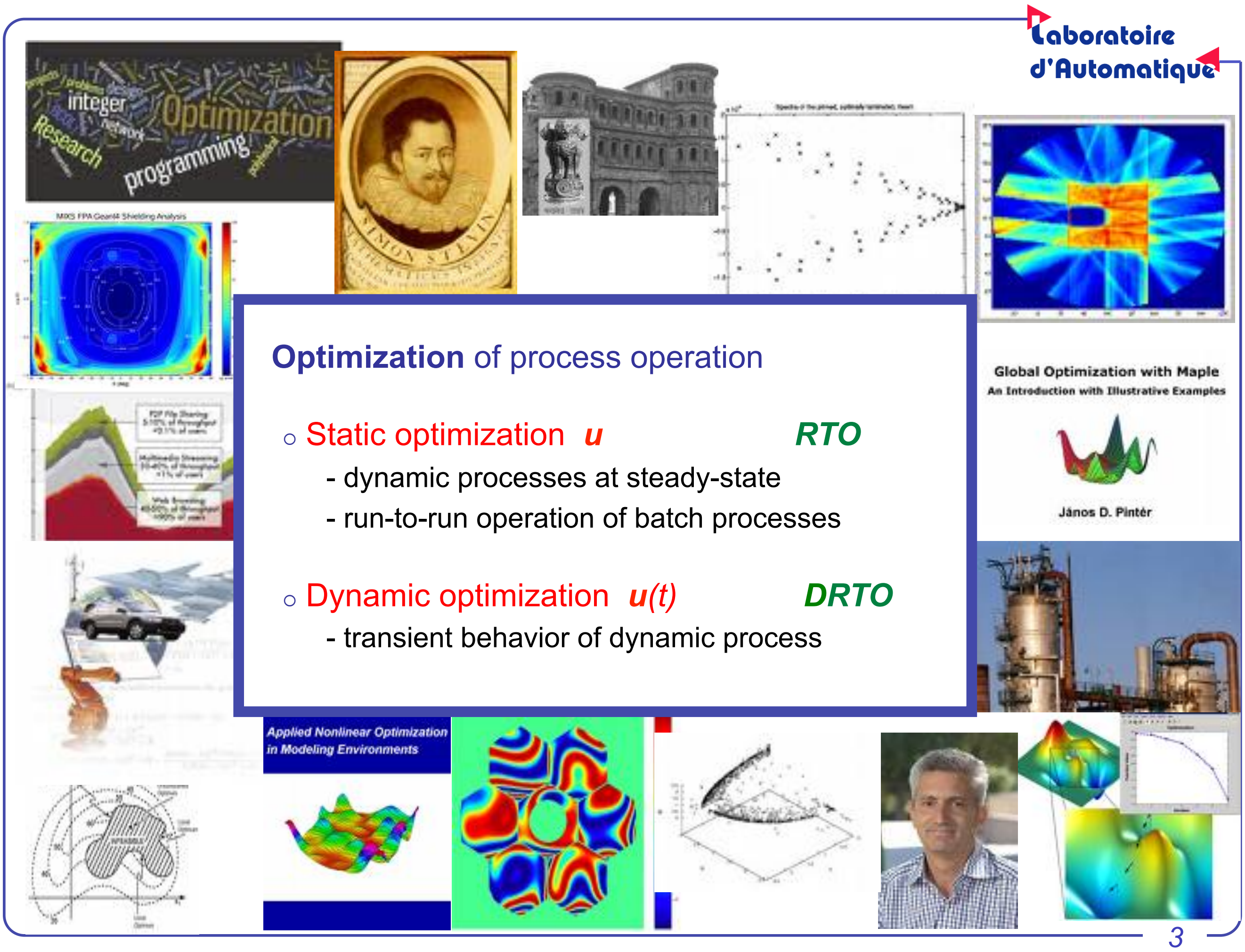


Laboratoire d'Automatique

- **Optimization** is a natural choice to:
  - design and conceive highly integrated systems
  - reduce production costs and improve product quality
  - meet safety and environmental regulations

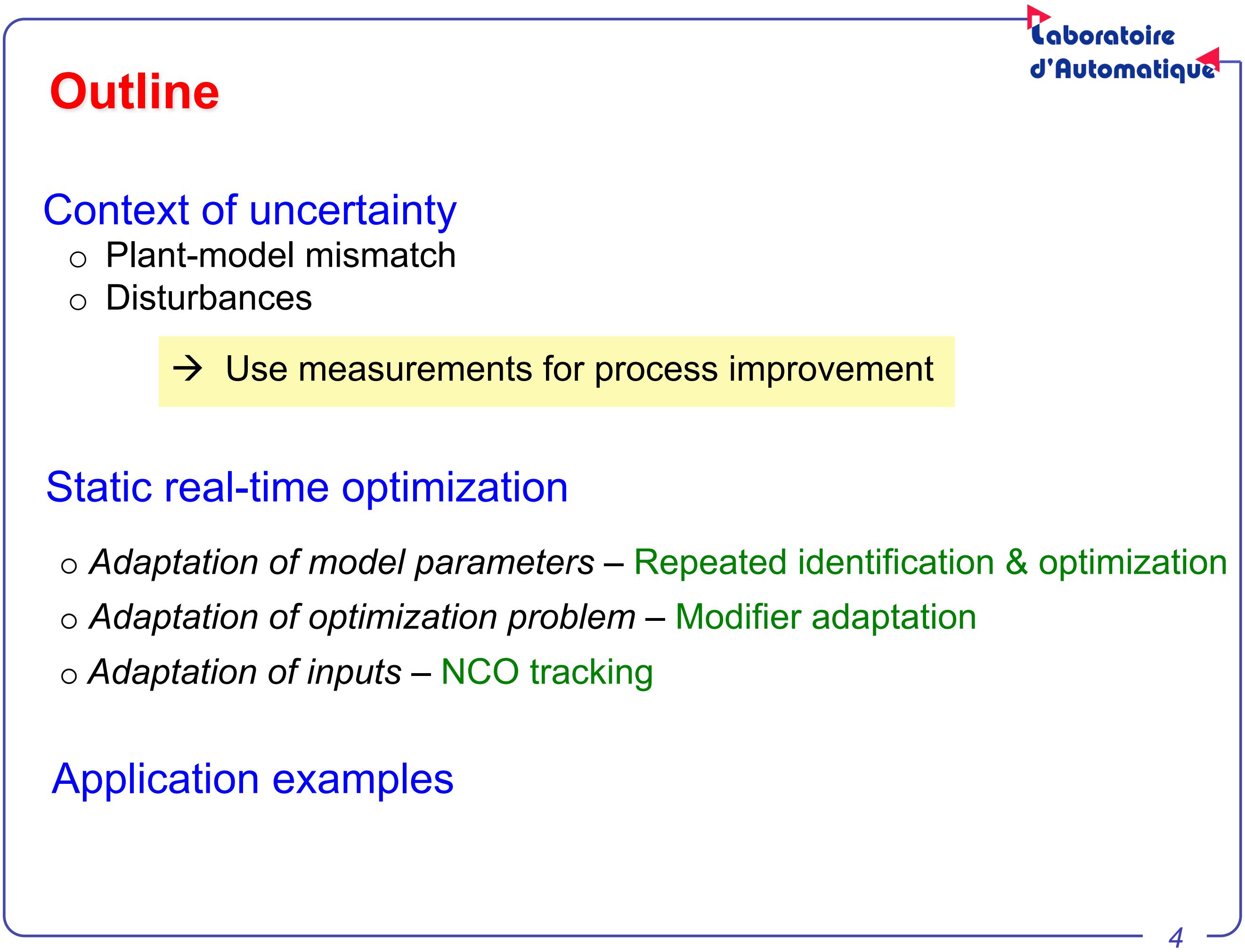
- **Mathematical models** are ubiquitous in almost every aspect of science and engineering

2


Laboratoire
d'Automatique
integer
Optimization
Research
programming
Global Optimization with Maple
An Introduction with Illustrative Examples
János D. Pintér
Applied Nonlinear Optimization
in Modeling Environments
3

Optimization of process operation

- Static optimization *u* ***RTO***
  - dynamic processes at steady-state
  - run-to-run operation of batch processes
- Dynamic optimization *u(t)* ***DRTO***
  - transient behavior of dynamic process



Outline

Context of uncertainty

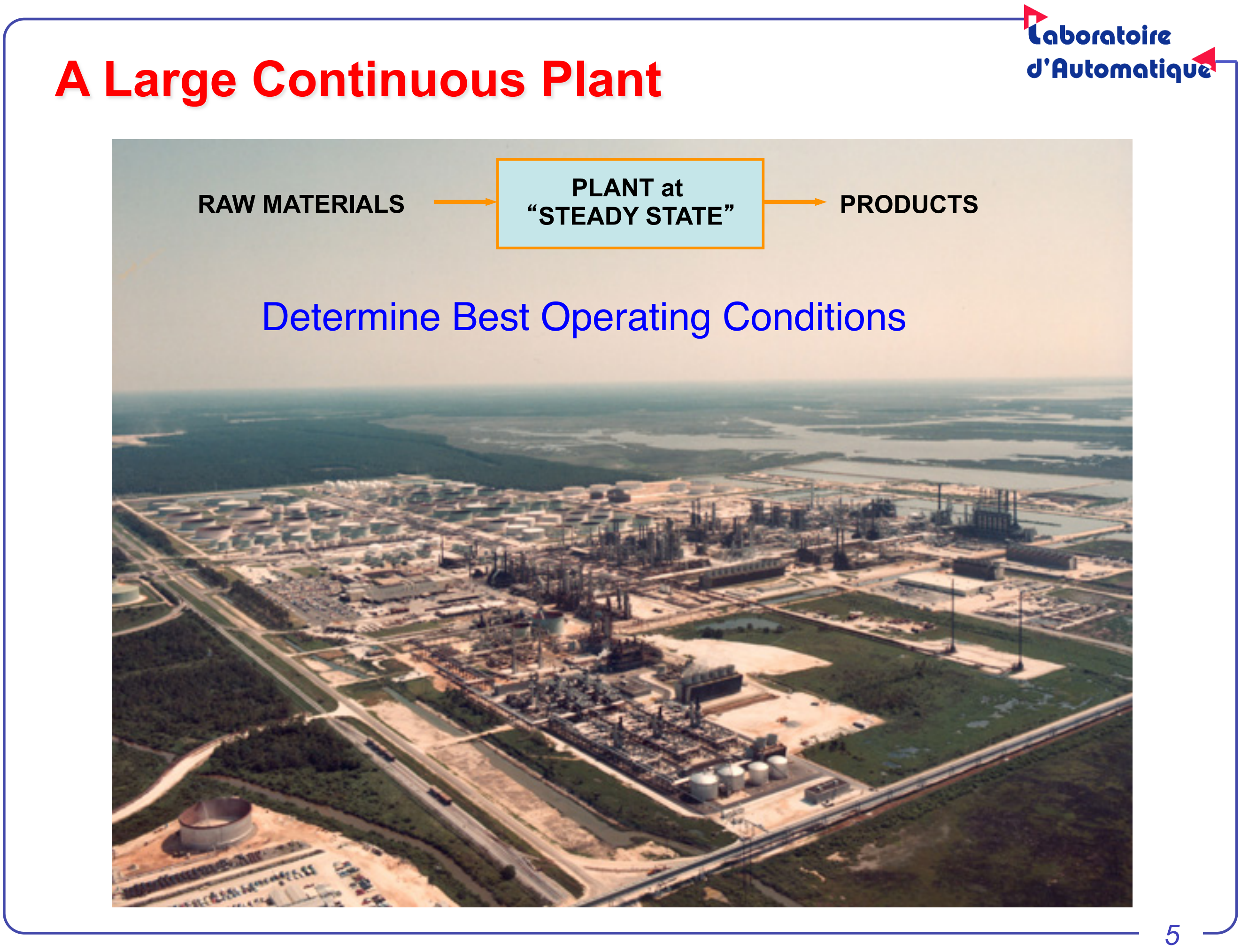
- ○ Plant-model mismatch
- ○ Disturbances

→ Use measurements for process improvement

Static real-time optimization

- ○ *Adaptation of model parameters* – Repeated identification & optimization
- ○ *Adaptation of optimization problem* – Modifier adaptation
- ○ *Adaptation of inputs* – NCO tracking

Application examples



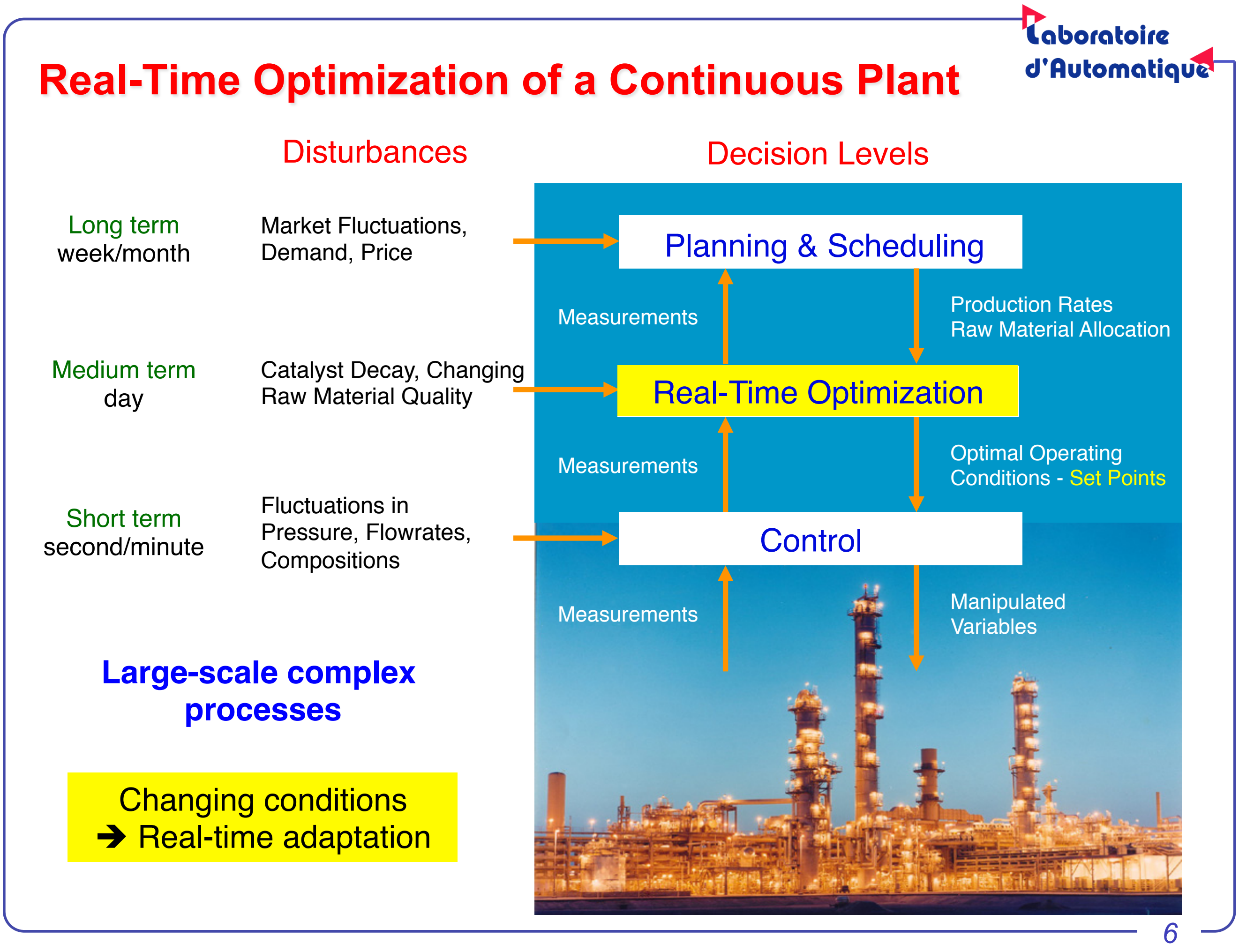
Laboratoire
d'Automatique

A Large Continuous Plant

RAW MATERIALS → PLANT at "STEADY STATE" → PRODUCTS

Determine Best Operating Conditions

5



Laboratoire d'Automatique

Real-Time Optimization of a Continuous Plant

Disturbances

Decision Levels

Long term
week/month

Market Fluctuations, Demand, Price

Medium term
day

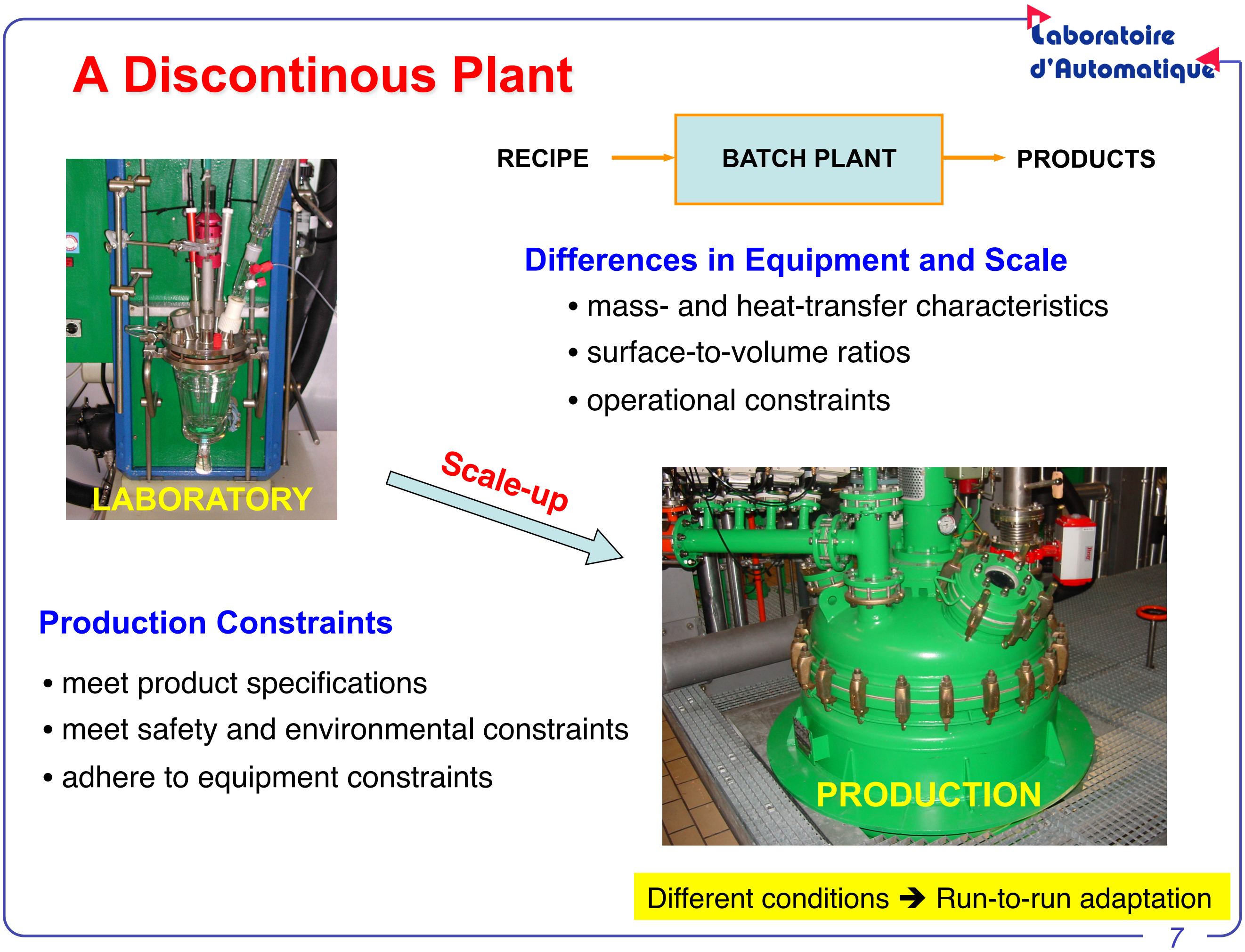
Catalyst Decay, Changing Raw Material Quality

Short term
second/minute

Fluctuations in Pressure, Flowrates, Compositions

Large-scale complex processes

Changing conditions
➔ Real-time adaptation



Laboratoire d'Automatique

A Discontinous Plant

RECIPE → BATCH PLANT → PRODUCTS

Differences in Equipment and Scale

- mass- and heat-transfer characteristics
- surface-to-volume ratios
- operational constraints

LABORATORY

Scale-up

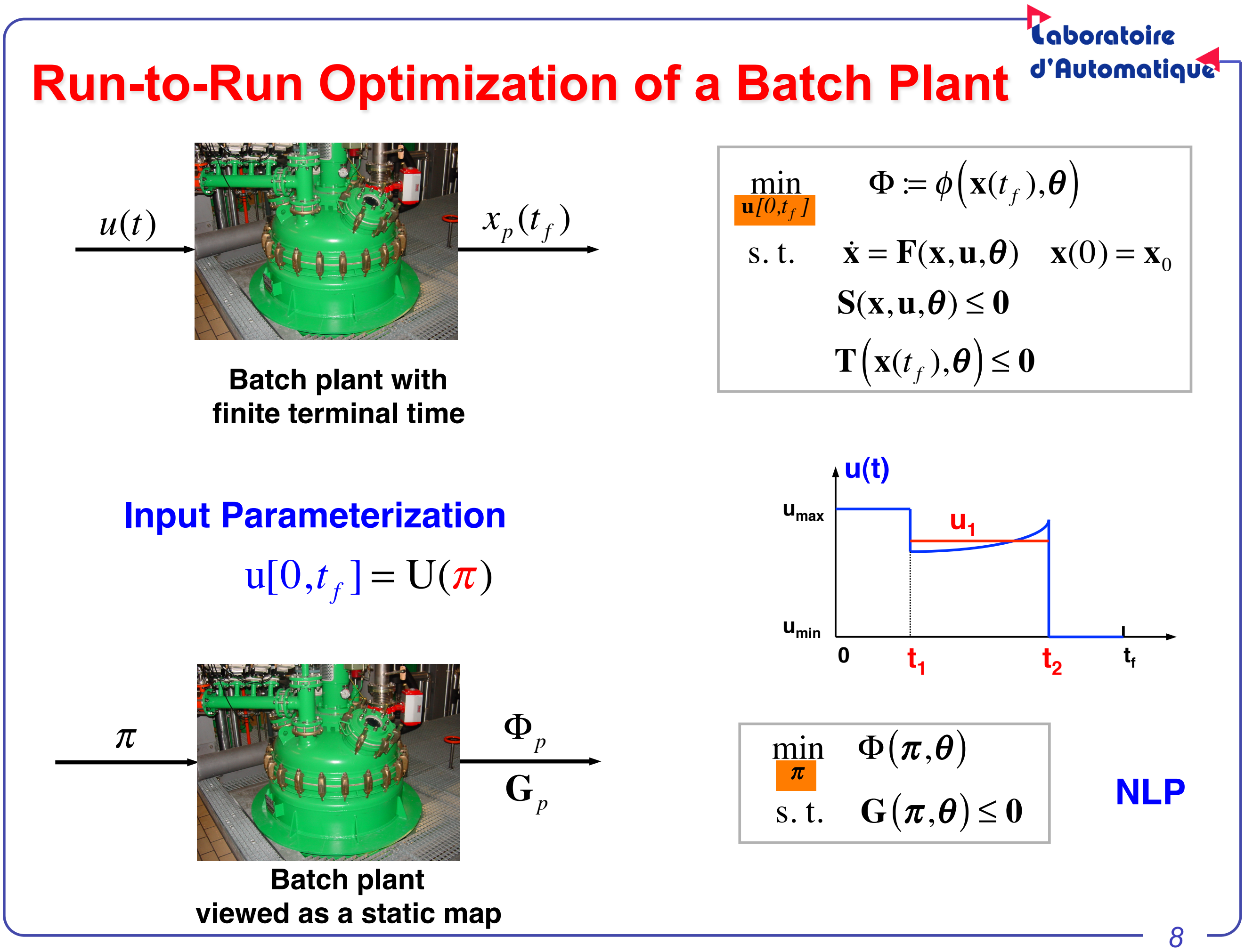
PRODUCTION

Production Constraints

- meet product specifications
- meet safety and environmental constraints
- adhere to equipment constraints

Different conditions ➔ Run-to-run adaptation

7



Laboratoire d'Automatique

Run-to-Run Optimization of a Batch Plant

$u(t)$ → $x_p(t_f)$

Batch plant with finite terminal time

$$\min_{\mathbf{u}[0,t_f]} \quad \Phi := \phi\big(\mathbf{x}(t_f),\boldsymbol{\theta}\big)$$
$$\text{s. t.} \quad \dot{\mathbf{x}} = \mathbf{F}(\mathbf{x},\mathbf{u},\boldsymbol{\theta}) \quad \mathbf{x}(0) = \mathbf{x}_0$$
$$\mathbf{S}(\mathbf{x},\mathbf{u},\boldsymbol{\theta}) \le \mathbf{0}$$
$$\mathbf{T}\big(\mathbf{x}(t_f),\boldsymbol{\theta}\big) \le \mathbf{0}$$

Input Parameterization

$$\mathrm{u}[0,t_f] = \mathrm{U}(\pi)$$

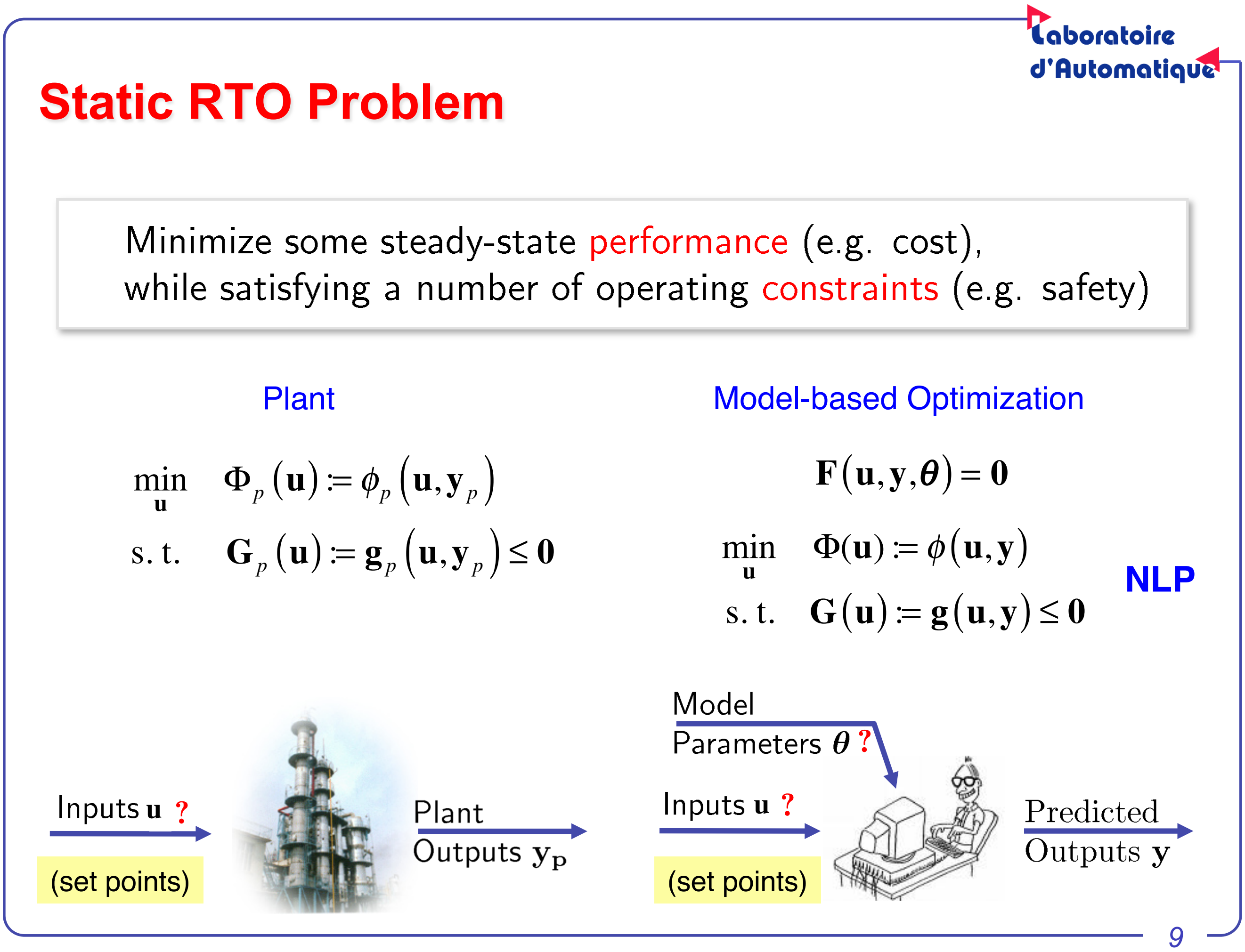
π → Φ_p, $\mathbf{G}_p$

Batch plant viewed as a static map

$$\min_{\pi} \quad \Phi(\boldsymbol{\pi},\boldsymbol{\theta})$$
$$\text{s. t.} \quad \mathbf{G}(\boldsymbol{\pi},\boldsymbol{\theta}) \le \mathbf{0}$$

NLP

8



Laboratoire d'Automatique

Static RTO Problem

Minimize some steady-state performance (e.g. cost),
while satisfying a number of operating constraints (e.g. safety)

Plant

$$\min_{\mathbf{u}} \quad \Phi_p(\mathbf{u}) := \phi_p(\mathbf{u},\mathbf{y}_p)$$

$$\text{s.t.} \quad \mathbf{G}_p(\mathbf{u}) := \mathbf{g}_p(\mathbf{u},\mathbf{y}_p) \le \mathbf{0}$$

Model-based Optimization

$$\mathbf{F}(\mathbf{u},\mathbf{y},\boldsymbol{\theta}) = \mathbf{0}$$

$$\min_{\mathbf{u}} \quad \Phi(\mathbf{u}) := \phi(\mathbf{u},\mathbf{y})$$

$$\text{s.t.} \quad \mathbf{G}(\mathbf{u}) := \mathbf{g}(\mathbf{u},\mathbf{y}) \le \mathbf{0}$$

NLP

Inputs $\mathbf{u}$?

(set points)

Plant Outputs $\mathbf{y}_\mathbf{P}$

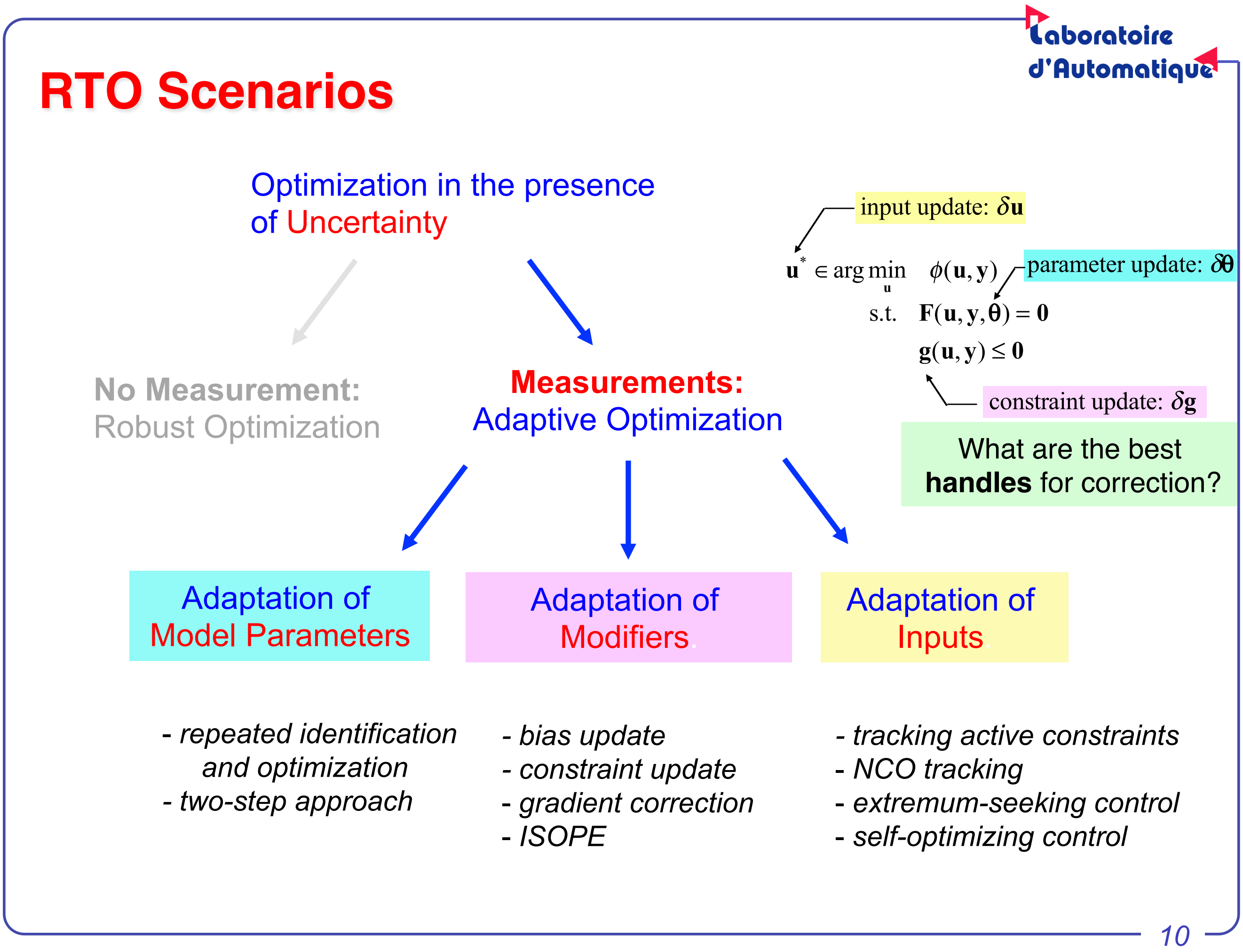
Model Parameters $\boldsymbol{\theta}$?

Inputs $\mathbf{u}$?

(set points)

Predicted Outputs $\mathbf{y}$

9



Laboratoire d'Automatique

RTO Scenarios

Optimization in the presence of Uncertainty

No Measurement: Robust Optimization

Measurements: Adaptive Optimization

$$\mathbf{u}^* \in \arg\min_{\mathbf{u}} \ \phi(\mathbf{u},\mathbf{y}) \quad \text{s.t.} \quad \mathbf{F}(\mathbf{u},\mathbf{y},\boldsymbol{\theta}) = \mathbf{0}, \quad \mathbf{g}(\mathbf{u},\mathbf{y}) \le \mathbf{0}$$

input update: $\delta\mathbf{u}$

parameter update: $\delta\boldsymbol{\theta}$

constraint update: $\delta\mathbf{g}$

What are the best **handles** for correction?

Adaptation of Model Parameters

- *repeated identification and optimization*
- *two-step approach*

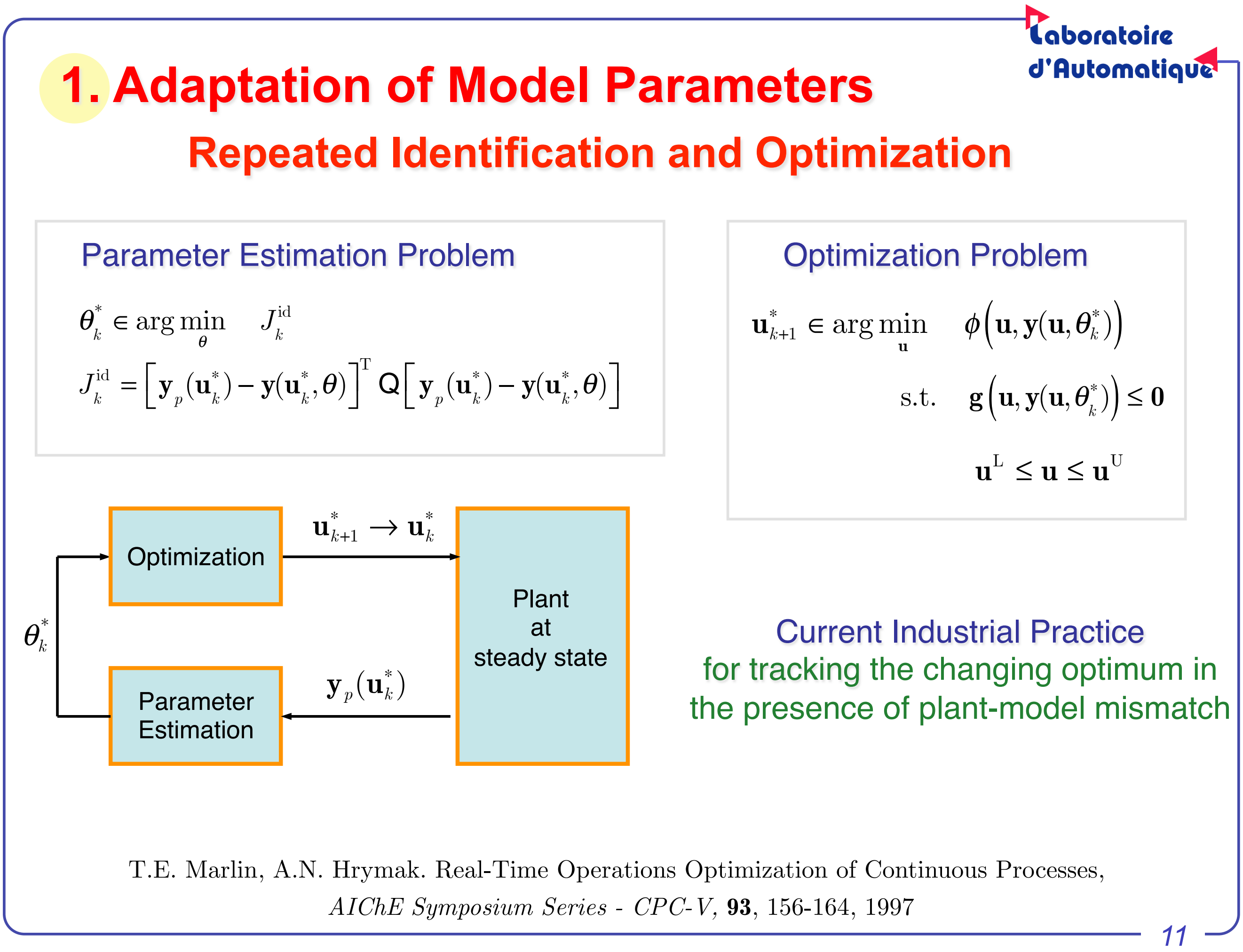
Adaptation of Modifiers

- *bias update*
- *constraint update*
- *gradient correction*
- *ISOPE*

Adaptation of Inputs

- *tracking active constraints*
- *NCO tracking*
- *extremum-seeking control*
- *self-optimizing control*

10



Laboratoire
d'Automatique

1. Adaptation of Model Parameters

Repeated Identification and Optimization

Parameter Estimation Problem

$$\theta_k^* \in \arg\min_{\theta} \quad J_k^{\text{id}}$$

$$J_k^{\text{id}} = \left[\mathbf{y}_p(\mathbf{u}_k^*) - \mathbf{y}(\mathbf{u}_k^*,\theta)\right]^{\mathrm{T}} \mathsf{Q} \left[\mathbf{y}_p(\mathbf{u}_k^*) - \mathbf{y}(\mathbf{u}_k^*,\theta)\right]$$

Optimization Problem

$$\mathbf{u}_{k+1}^* \in \arg\min_{\mathbf{u}} \quad \phi\left(\mathbf{u},\mathbf{y}(\mathbf{u},\theta_k^*)\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{u},\mathbf{y}(\mathbf{u},\theta_k^*)\right) \leq \mathbf{0}$$

$$\mathbf{u}^{\mathrm{L}} \leq \mathbf{u} \leq \mathbf{u}^{\mathrm{U}}$$

Optimization

$\mathbf{u}_{k+1}^* \rightarrow \mathbf{u}_k^*$

Plant
at
steady state

$\mathbf{y}_p(\mathbf{u}_k^*)$

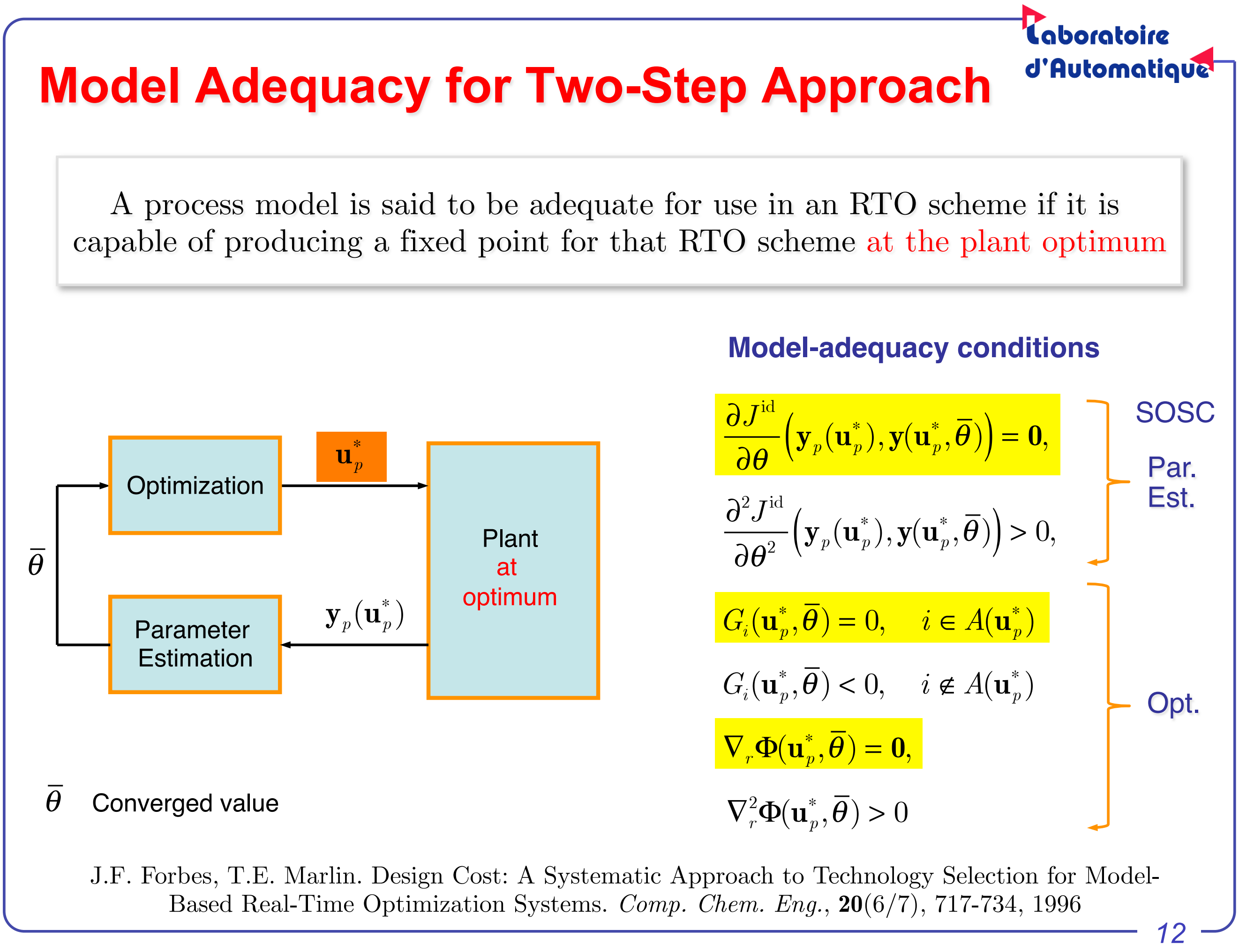
Parameter
Estimation

θ_k^*

Current Industrial Practice
for tracking the changing optimum in
the presence of plant-model mismatch

T.E. Marlin, A.N. Hrymak. Real-Time Operations Optimization of Continuous Processes,
AIChE Symposium Series - CPC-V, **93**, 156-164, 1997

11



Model Adequacy for Two-Step Approach

A process model is said to be adequate for use in an RTO scheme if it is capable of producing a fixed point for that RTO scheme at the plant optimum

Optimization
$\mathbf{u}_p^*$
Plant
at
optimum
$\bar{\theta}$
Parameter Estimation
$\mathbf{y}_p(\mathbf{u}_p^*)$

$\bar{\theta}$ Converged value

Model-adequacy conditions

SOSC

Par. Est.

$$\frac{\partial J^{\text{id}}}{\partial \theta}\left(\mathbf{y}_p(\mathbf{u}_p^*), \mathbf{y}(\mathbf{u}_p^*, \bar{\theta})\right) = \mathbf{0},$$

$$\frac{\partial^2 J^{\text{id}}}{\partial \theta^2}\left(\mathbf{y}_p(\mathbf{u}_p^*), \mathbf{y}(\mathbf{u}_p^*, \bar{\theta})\right) > 0,$$

Opt.

$$G_i(\mathbf{u}_p^*, \bar{\theta}) = 0, \quad i \in A(\mathbf{u}_p^*)$$

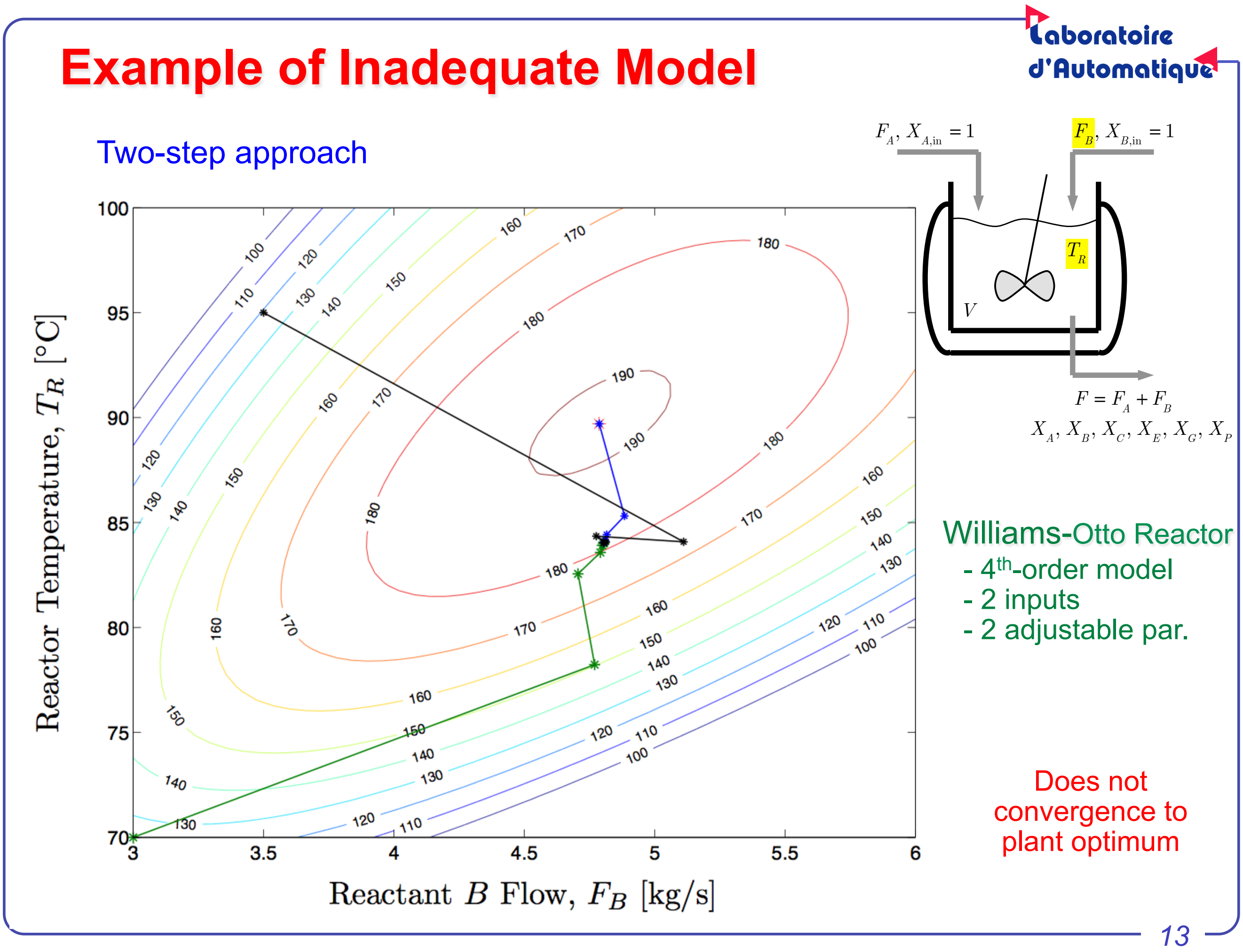
$$G_i(\mathbf{u}_p^*, \bar{\theta}) < 0, \quad i \notin A(\mathbf{u}_p^*)$$

$$\nabla_r \Phi(\mathbf{u}_p^*, \bar{\theta}) = \mathbf{0},$$

$$\nabla_r^2 \Phi(\mathbf{u}_p^*, \bar{\theta}) > 0$$

J.F. Forbes, T.E. Marlin. Design Cost: A Systematic Approach to Technology Selection for Model-Based Real-Time Optimization Systems. *Comp. Chem. Eng.*, **20**(6/7), 717-734, 1996

12



Example of Inadequate Model

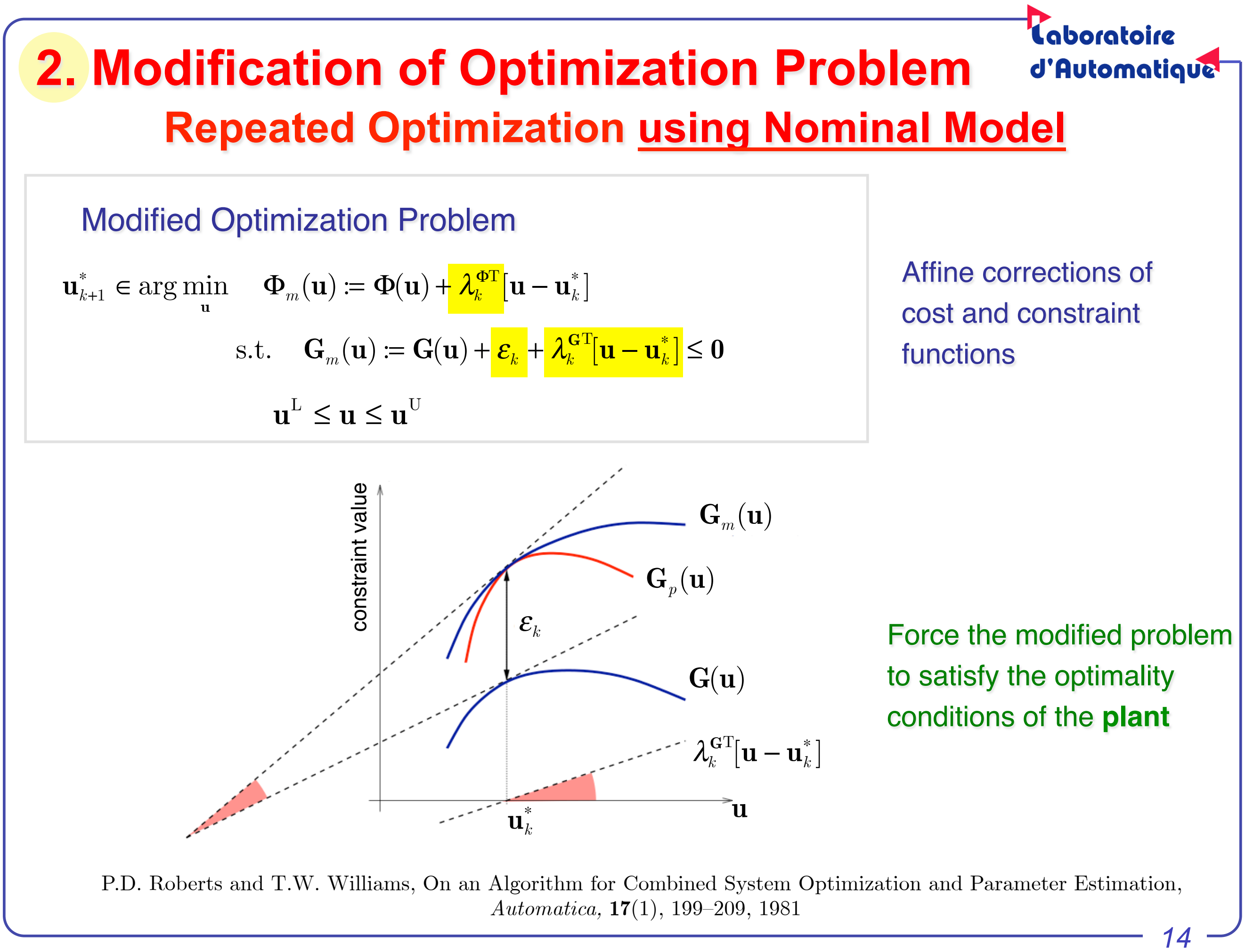
Two-step approach

Williams-Otto Reactor

- 4th-order model
- 2 inputs
- 2 adjustable par.

Does not convergence to plant optimum

13



Laboratoire d'Automatique

2. Modification of Optimization Problem

Repeated Optimization <u>using Nominal Model</u>

Modified Optimization Problem

$$\mathbf{u}_{k+1}^* \in \arg\min_{\mathbf{u}} \quad \Phi_m(\mathbf{u}) := \Phi(\mathbf{u}) + \lambda_k^{\Phi\mathrm{T}}[\mathbf{u}-\mathbf{u}_k^*]$$

$$\text{s.t.} \quad \mathbf{G}_m(\mathbf{u}) := \mathbf{G}(\mathbf{u}) + \varepsilon_k + \lambda_k^{\mathrm{GT}}[\mathbf{u}-\mathbf{u}_k^*] \leq \mathbf{0}$$

$$\mathbf{u}^{\mathrm{L}} \leq \mathbf{u} \leq \mathbf{u}^{\mathrm{U}}$$

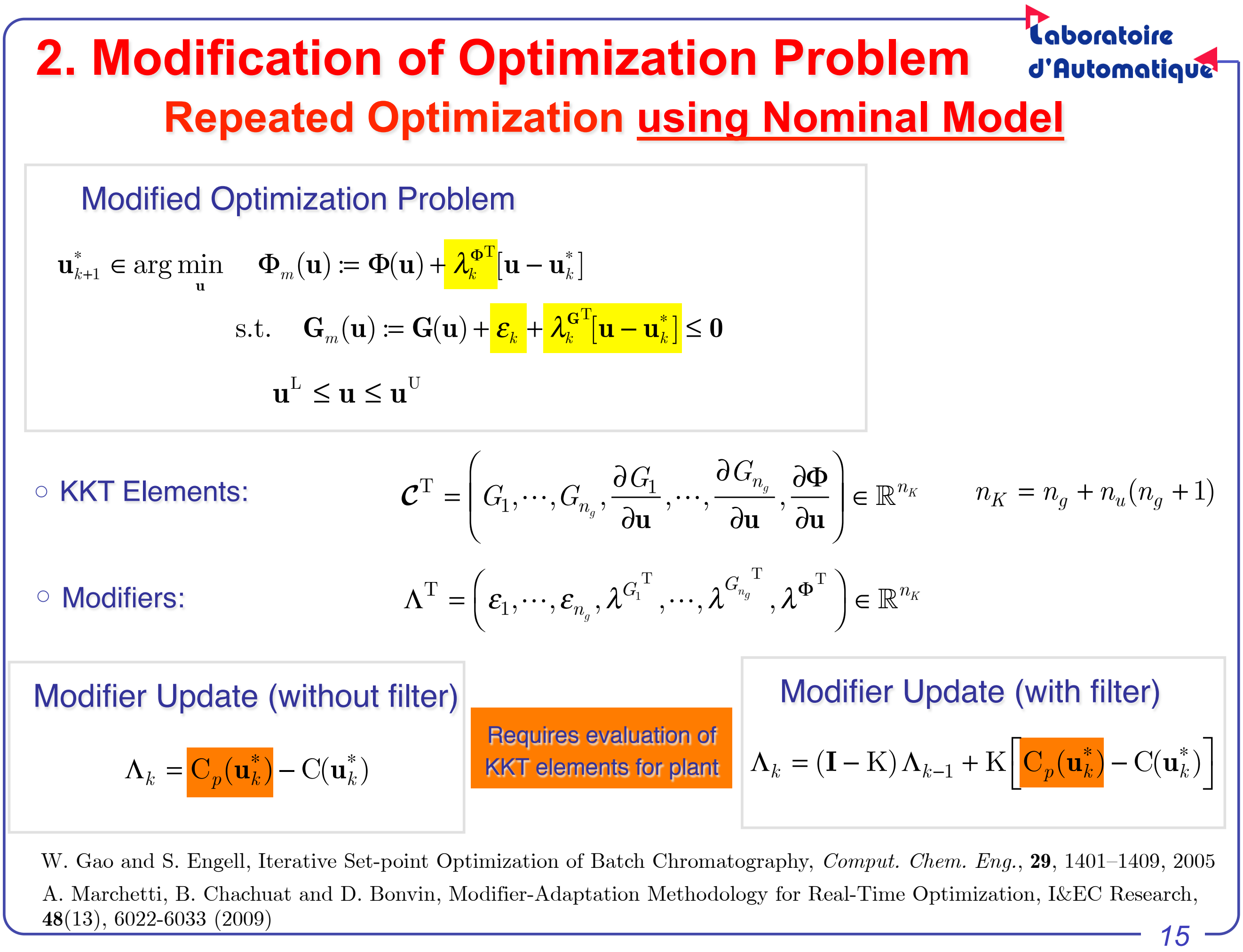
Affine corrections of cost and constraint functions

constraint value
$\mathbf{G}_m(\mathbf{u})$
$\mathbf{G}_p(\mathbf{u})$
ε_k
$\mathbf{G}(\mathbf{u})$
$\lambda_k^{\mathrm{GT}}[\mathbf{u}-\mathbf{u}_k^*]$
$\mathbf{u}_k^*$
$\mathbf{u}$

Force the modified problem to satisfy the optimality conditions of the **plant**

P.D. Roberts and T.W. Williams, On an Algorithm for Combined System Optimization and Parameter Estimation, *Automatica*, **17**(1), 199–209, 1981

14



2. Modification of Optimization Problem

Repeated Optimization using Nominal Model

Modified Optimization Problem

$$\mathbf{u}_{k+1}^* \in \arg\min_{\mathbf{u}} \quad \Phi_m(\mathbf{u}) := \Phi(\mathbf{u}) + \lambda_k^{\Phi\mathrm{T}}[\mathbf{u} - \mathbf{u}_k^*]$$

$$\text{s.t.} \quad \mathbf{G}_m(\mathbf{u}) := \mathbf{G}(\mathbf{u}) + \varepsilon_k + \lambda_k^{\mathrm{G}\mathrm{T}}[\mathbf{u} - \mathbf{u}_k^*] \le \mathbf{0}$$

$$\mathbf{u}^{\mathrm{L}} \le \mathbf{u} \le \mathbf{u}^{\mathrm{U}}$$

○ KKT Elements: $\mathcal{C}^{\mathrm{T}} = \left(G_1, \cdots, G_{n_g}, \frac{\partial G_1}{\partial \mathbf{u}}, \cdots, \frac{\partial G_{n_g}}{\partial \mathbf{u}}, \frac{\partial \Phi}{\partial \mathbf{u}} \right) \in \mathbb{R}^{n_K}$ $\quad n_K = n_g + n_u(n_g + 1)$

○ Modifiers: $\Lambda^{\mathrm{T}} = \left(\varepsilon_1, \cdots, \varepsilon_{n_g}, \lambda^{G_1^{\mathrm{T}}}, \cdots, \lambda^{G_{n_g}^{\mathrm{T}}}, \lambda^{\Phi^{\mathrm{T}}} \right) \in \mathbb{R}^{n_K}$

Modifier Update (without filter)

$$\Lambda_k = \mathrm{C}_p(\mathbf{u}_k^*) - \mathrm{C}(\mathbf{u}_k^*)$$

Requires evaluation of KKT elements for plant

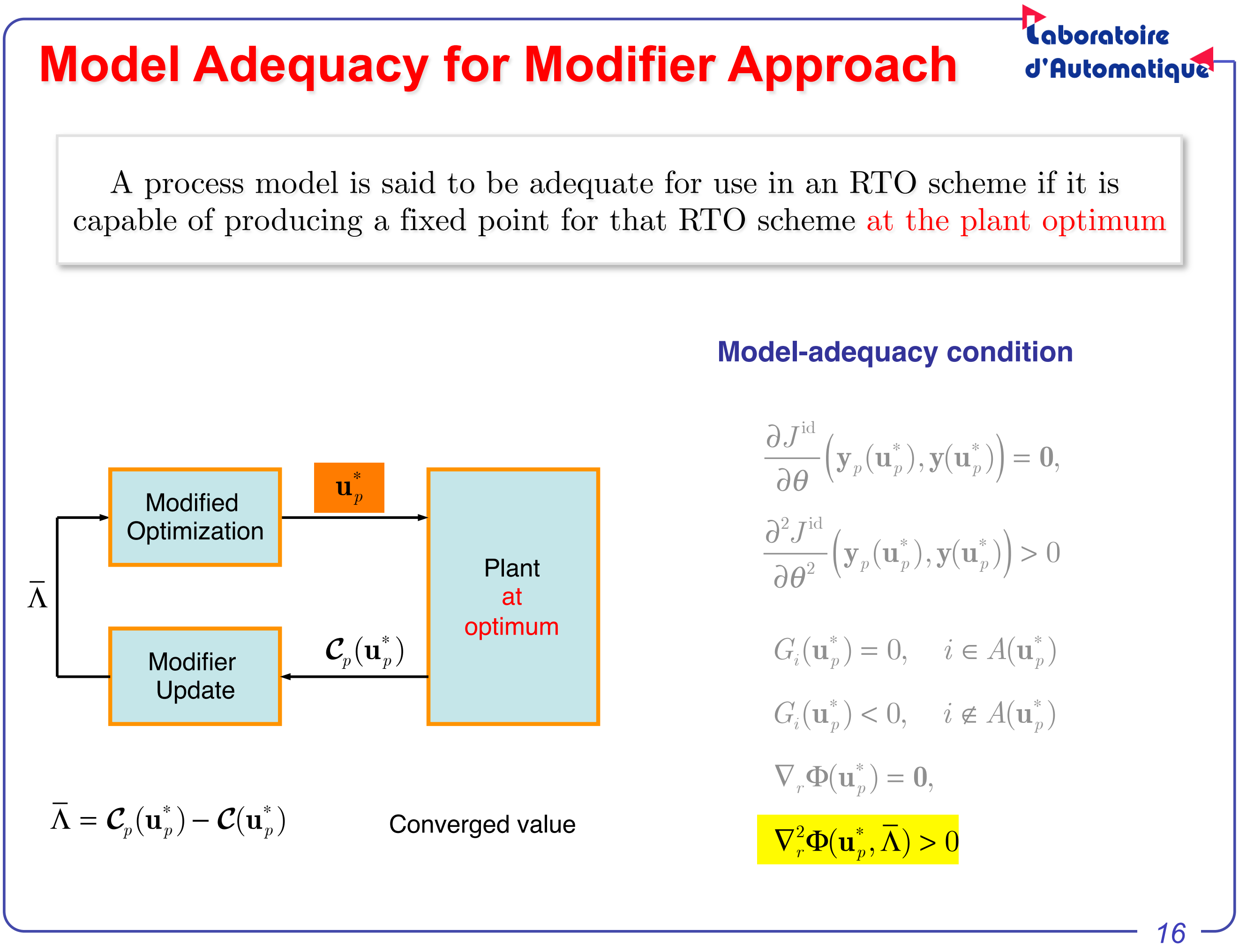
Modifier Update (with filter)

$$\Lambda_k = (\mathbf{I} - \mathrm{K})\,\Lambda_{k-1} + \mathrm{K}\left[\mathrm{C}_p(\mathbf{u}_k^*) - \mathrm{C}(\mathbf{u}_k^*)\right]$$

W. Gao and S. Engell, Iterative Set-point Optimization of Batch Chromatography, *Comput. Chem. Eng.*, **29**, 1401–1409, 2005

A. Marchetti, B. Chachuat and D. Bonvin, Modifier-Adaptation Methodology for Real-Time Optimization, I&EC Research, **48**(13), 6022-6033 (2009)

15



Model Adequacy for Modifier Approach

A process model is said to be adequate for use in an RTO scheme if it is capable of producing a fixed point for that RTO scheme at the plant optimum

Modified Optimization

$\mathbf{u}_p^*$

Plant at optimum

$\bar{\Lambda}$

Modifier Update

$\mathcal{C}_p(\mathbf{u}_p^*)$

$\bar{\Lambda} = \mathcal{C}_p(\mathbf{u}_p^*) - \mathcal{C}(\mathbf{u}_p^*)$ Converged value

Model-adequacy condition

$$\frac{\partial J^{\mathrm{id}}}{\partial \theta}\left(\mathbf{y}_p(\mathbf{u}_p^*), \mathbf{y}(\mathbf{u}_p^*)\right) = \mathbf{0},$$

$$\frac{\partial^2 J^{\mathrm{id}}}{\partial \theta^2}\left(\mathbf{y}_p(\mathbf{u}_p^*), \mathbf{y}(\mathbf{u}_p^*)\right) > 0$$

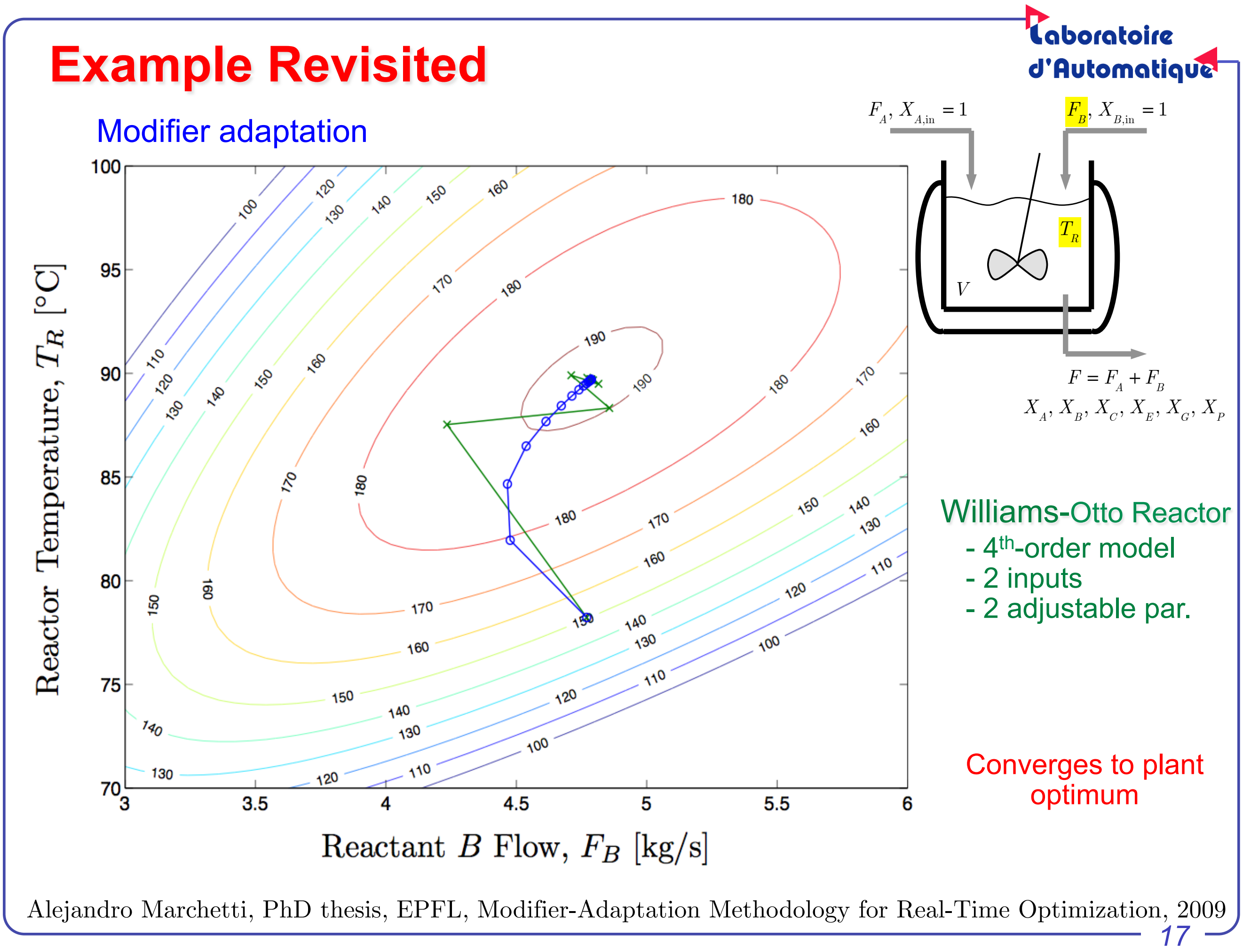
$$G_i(\mathbf{u}_p^*) = 0, \quad i \in A(\mathbf{u}_p^*)$$

$$G_i(\mathbf{u}_p^*) < 0, \quad i \notin A(\mathbf{u}_p^*)$$

$$\nabla_r \Phi(\mathbf{u}_p^*) = \mathbf{0},$$

$$\nabla_r^2 \Phi(\mathbf{u}_p^*, \bar{\Lambda}) > 0$$

16



Example Revisited

Modifier adaptation

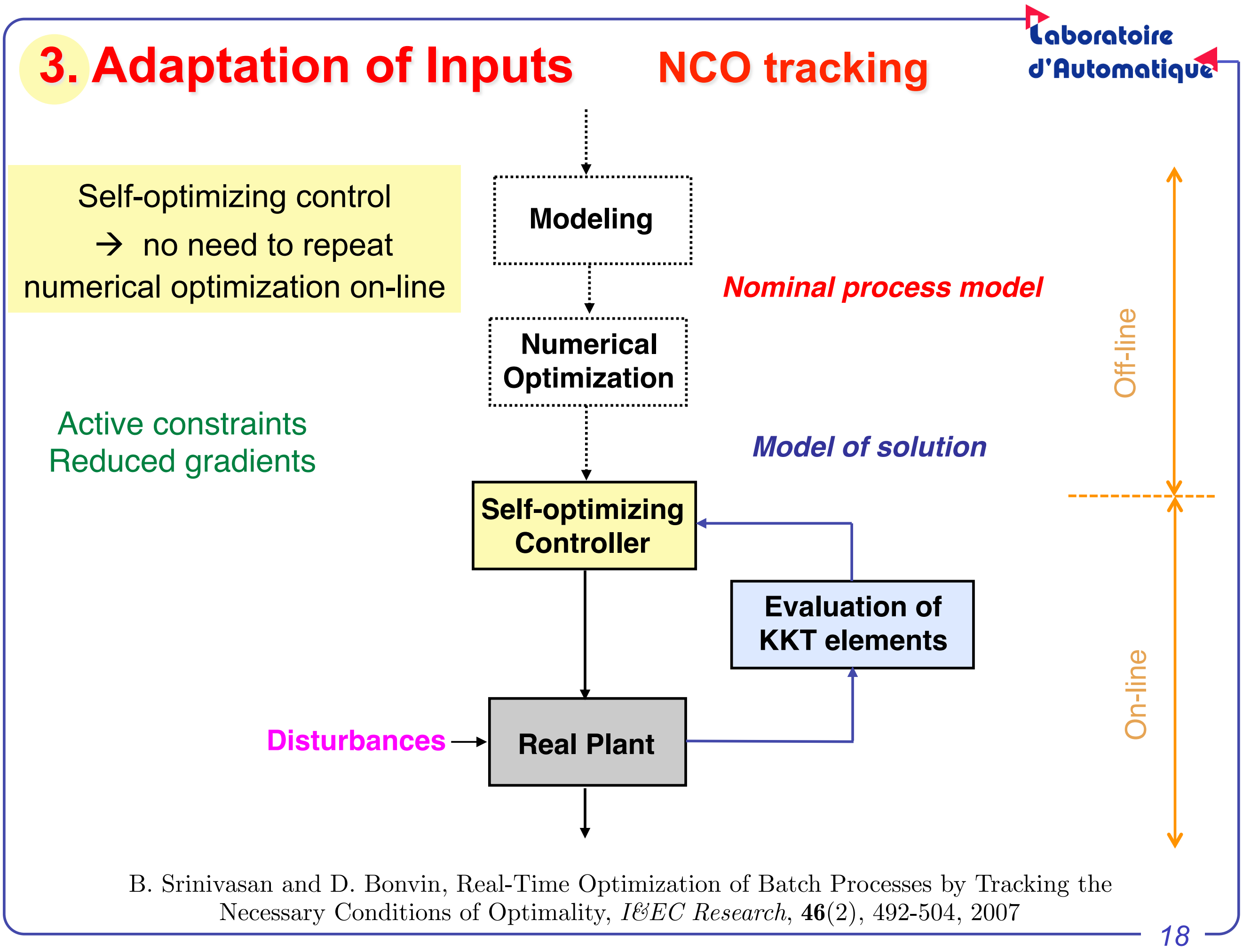
Williams-Otto Reactor

- 4th-order model
- 2 inputs
- 2 adjustable par.

Converges to plant optimum

Alejandro Marchetti, PhD thesis, EPFL, Modifier-Adaptation Methodology for Real-Time Optimization, 2009

17



3. Adaptation of Inputs

NCO tracking

Laboratoire d'Automatique

Self-optimizing control
→ no need to repeat numerical optimization on-line

Modeling

Nominal process model

Numerical Optimization

Active constraints
Reduced gradients

Model of solution

Self-optimizing Controller

Evaluation of KKT elements

Disturbances

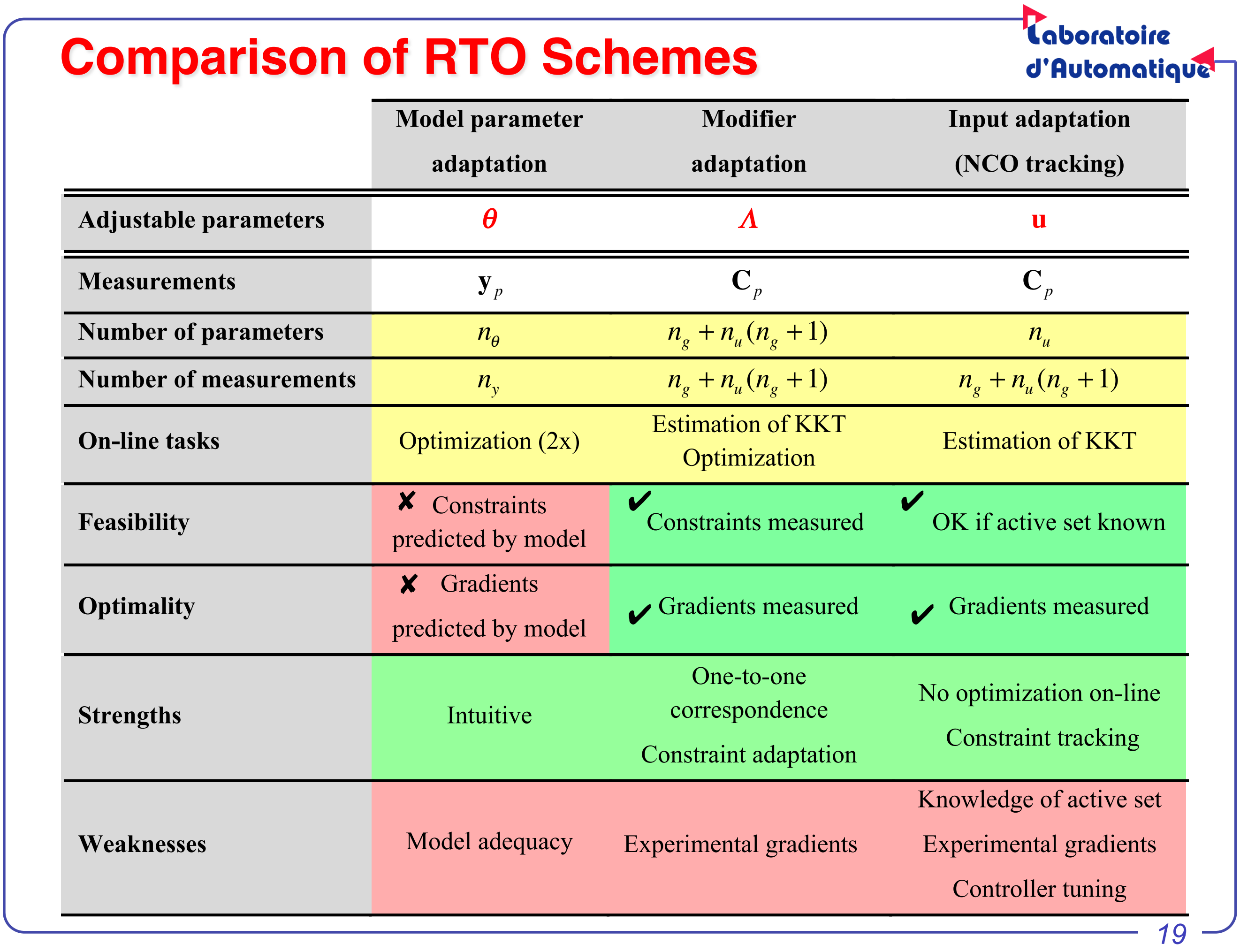
Real Plant

Off-line

On-line

B. Srinivasan and D. Bonvin, Real-Time Optimization of Batch Processes by Tracking the Necessary Conditions of Optimality, *I&EC Research*, **46**(2), 492-504, 2007

18

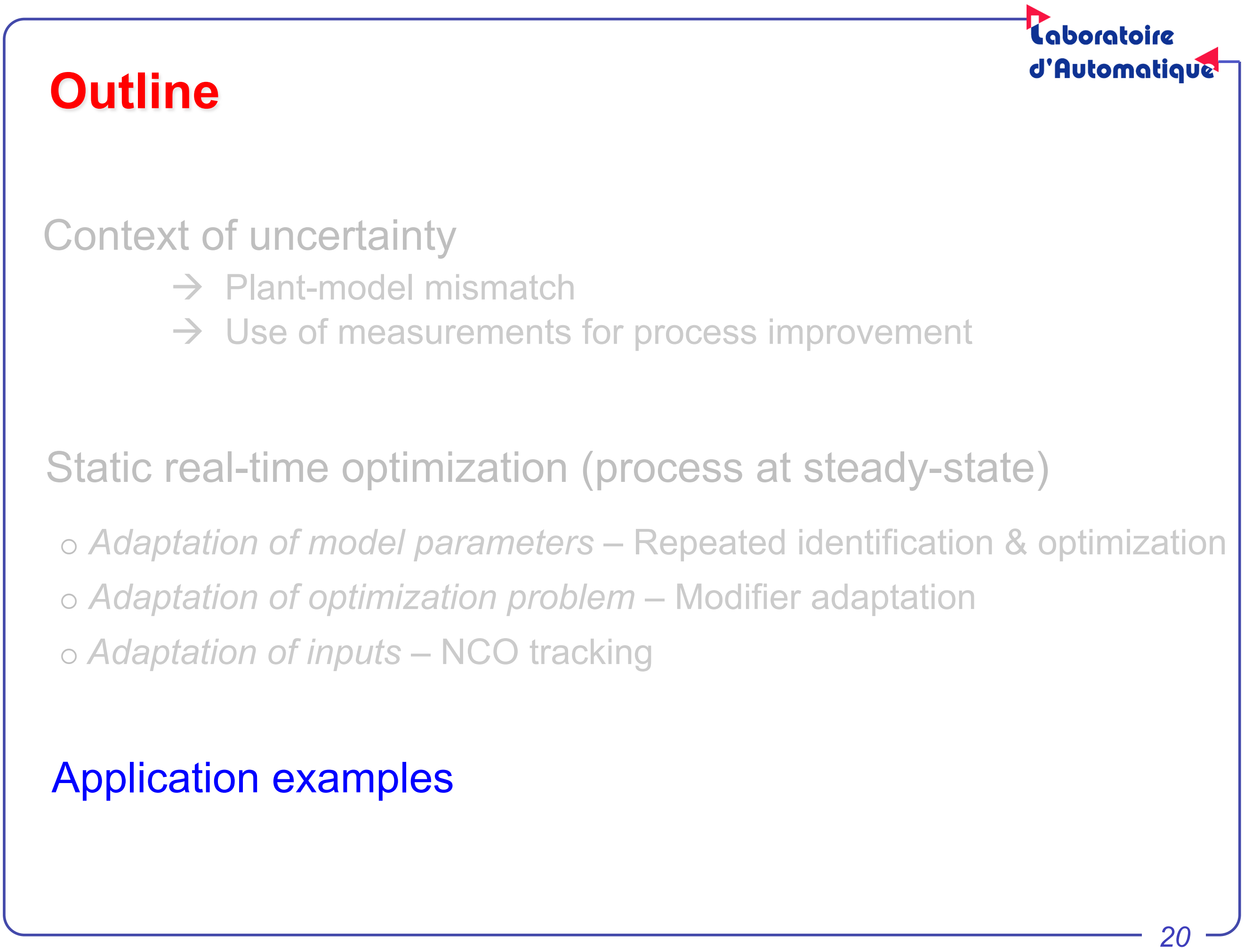


Comparison of RTO Schemes

Laboratoire d'Automatique

	Model parameter adaptation	Modifier adaptation	Input adaptation (NCO tracking)
Adjustable parameters	$\boldsymbol{\theta}$	$\boldsymbol{\Lambda}$	$\mathbf{u}$
Measurements	$\mathbf{y}_p$	$\mathbf{C}_p$	$\mathbf{C}_p$
Number of parameters	n_θ	$n_g + n_u(n_g + 1)$	n_u
Number of measurements	n_y	$n_g + n_u(n_g + 1)$	$n_g + n_u(n_g + 1)$
On-line tasks	Optimization (2x)	Estimation of KKT Optimization	Estimation of KKT
Feasibility	✘ Constraints predicted by model	✔ Constraints measured	✔ OK if active set known
Optimality	✘ Gradients predicted by model	✔ Gradients measured	✔ Gradients measured
Strengths	Intuitive	One-to-one correspondence Constraint adaptation	No optimization on-line Constraint tracking
Weaknesses	Model adequacy	Experimental gradients	Knowledge of active set Experimental gradients Controller tuning

19



Laboratoire
d'Automatique

Outline

Context of uncertainty

→ Plant-model mismatch

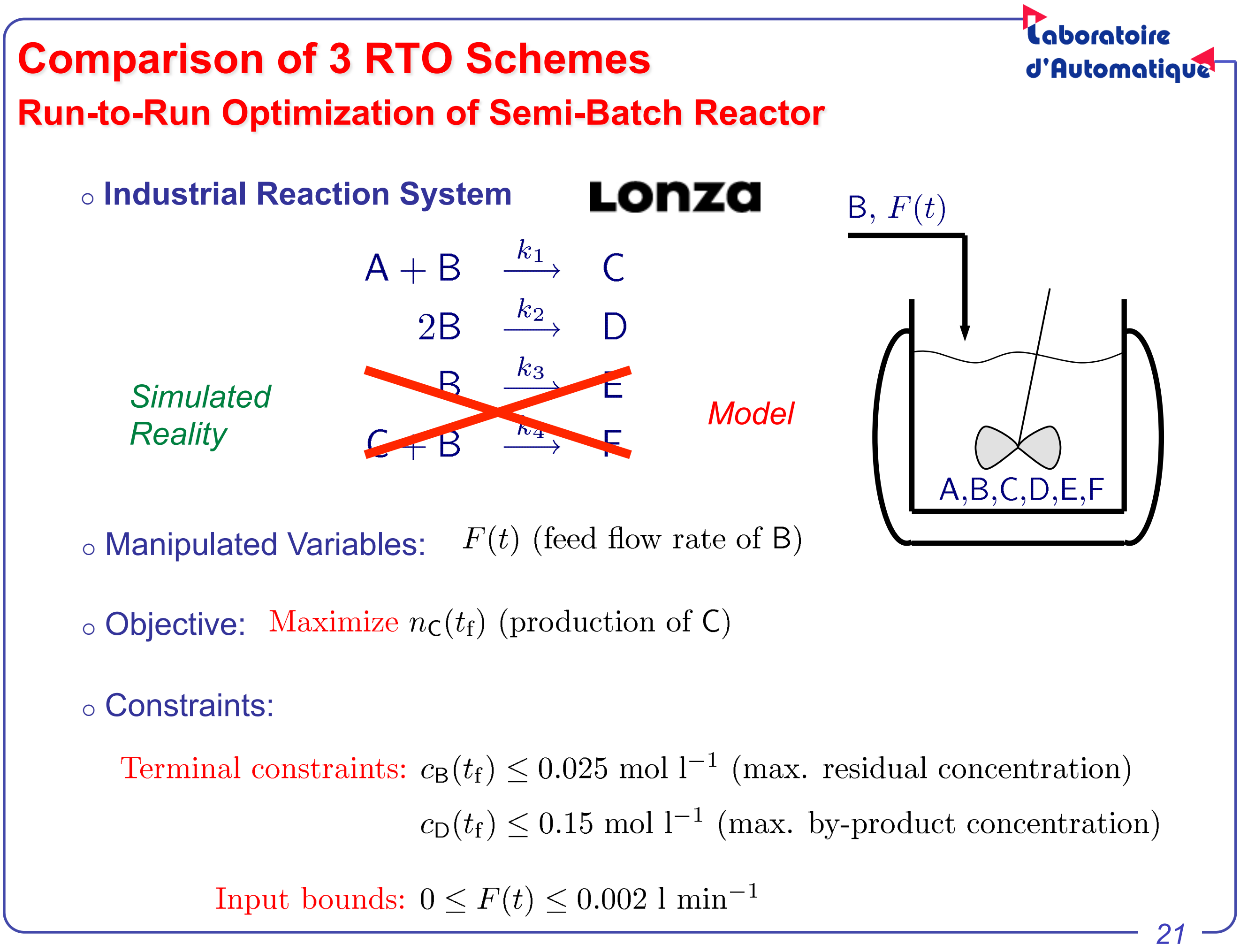
→ Use of measurements for process improvement

Static real-time optimization (process at steady-state)

- ○ *Adaptation of model parameters* – Repeated identification & optimization
- ○ *Adaptation of optimization problem* – Modifier adaptation
- ○ *Adaptation of inputs* – NCO tracking

Application examples

20



Laboratoire d'Automatique

Comparison of 3 RTO Schemes

Run-to-Run Optimization of Semi-Batch Reactor

- **Industrial Reaction System** Lonza

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\xrightarrow{k_1} \mathrm{C} \\ 2\mathrm{B} &\xrightarrow{k_2} \mathrm{D} \\ \cancel{\mathrm{B}} &\cancel{\xrightarrow{k_3} \mathrm{E}} \\ \cancel{\mathrm{C} + \mathrm{B}} &\cancel{\xrightarrow{k_4} \mathrm{F}} \end{aligned}$$

Simulated Reality

Model

- Manipulated Variables: $F(t)$ (feed flow rate of B)
- Objective: Maximize $n_{\mathrm{C}}(t_{\mathrm{f}})$ (production of C)
- Constraints:

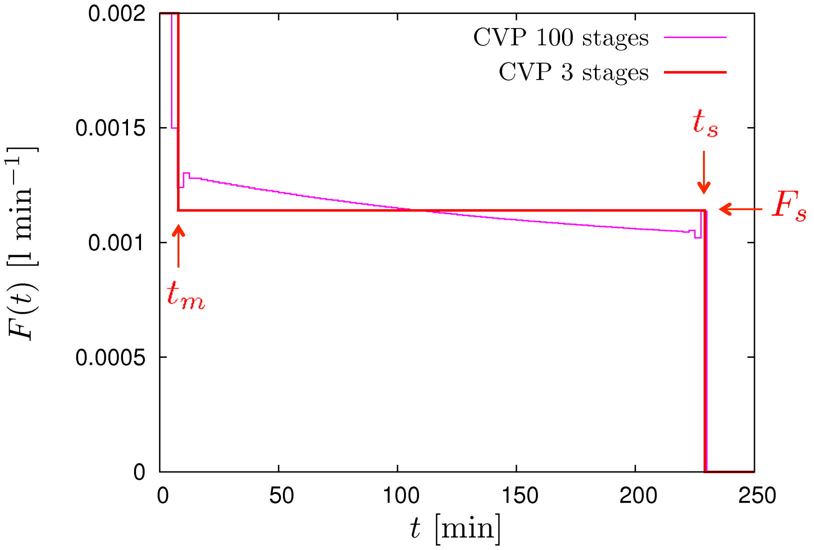
Terminal constraints: $c_{\mathrm{B}}(t_{\mathrm{f}}) \leq 0.025$ mol l^{-1} (max. residual concentration)

$c_{\mathrm{D}}(t_{\mathrm{f}}) \leq 0.15$ mol l^{-1} (max. by-product concentration)

Input bounds: $0 \leq F(t) \leq 0.002$ l min^{-1}

Laboratoire d'Automatique

Nominal Input Trajectory



◦ **Optimal Solution**

3 arcs, 2 active terminal constraints

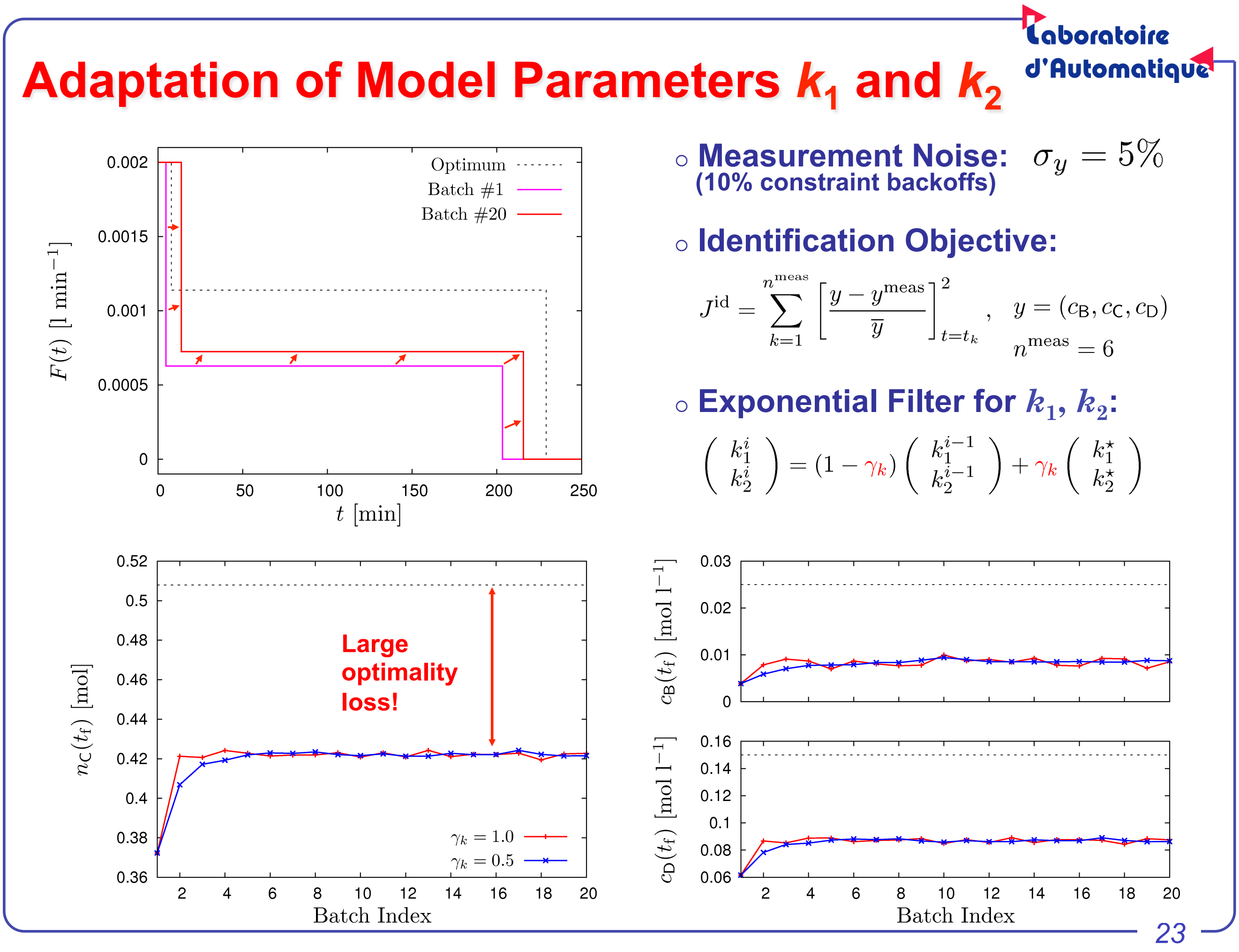
$J^\star \approx 0.5081$ mol

◦ **Approximate Solution**

Parameterization: $\mathbf{u} = (t_m, t_s, F_s)$

$J^\star \approx 0.5079$ mol

22



Adaptation of Model Parameters k_1 and k_2

Large optimality loss!

- **Measurement Noise:** $\sigma_y = 5\%$
  (10% constraint backoffs)

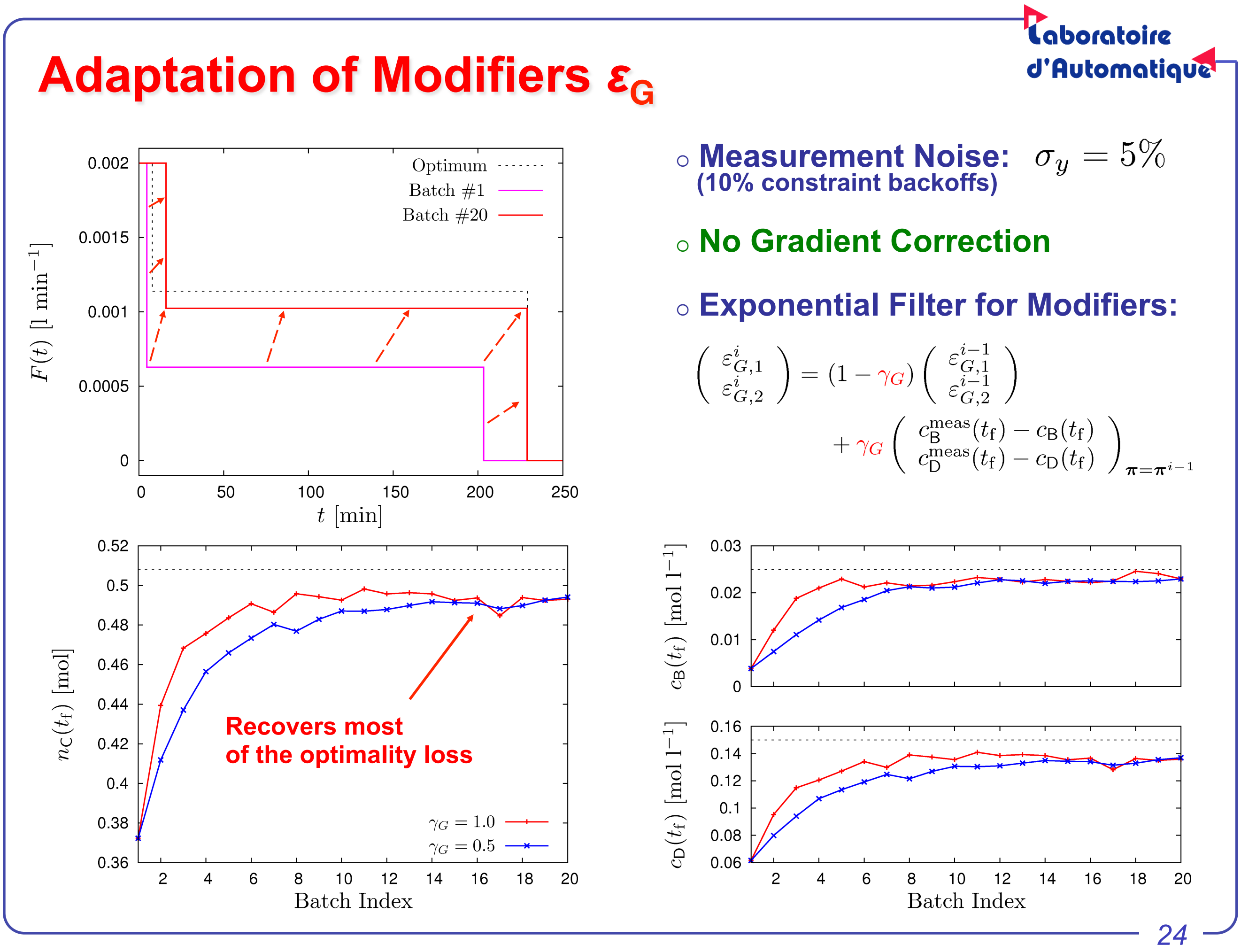
- **Identification Objective:**

$$J^{\text{id}} = \sum_{k=1}^{n^{\text{meas}}} \left[\frac{y - y^{\text{meas}}}{\overline{y}} \right]^2_{t=t_k}, \quad y = (c_B, c_C, c_D) \quad n^{\text{meas}} = 6$$

- **Exponential Filter for k_1, k_2:**

$$\begin{pmatrix} k_1^i \\ k_2^i \end{pmatrix} = (1 - \gamma_k) \begin{pmatrix} k_1^{i-1} \\ k_2^{i-1} \end{pmatrix} + \gamma_k \begin{pmatrix} k_1^\star \\ k_2^\star \end{pmatrix}$$

23



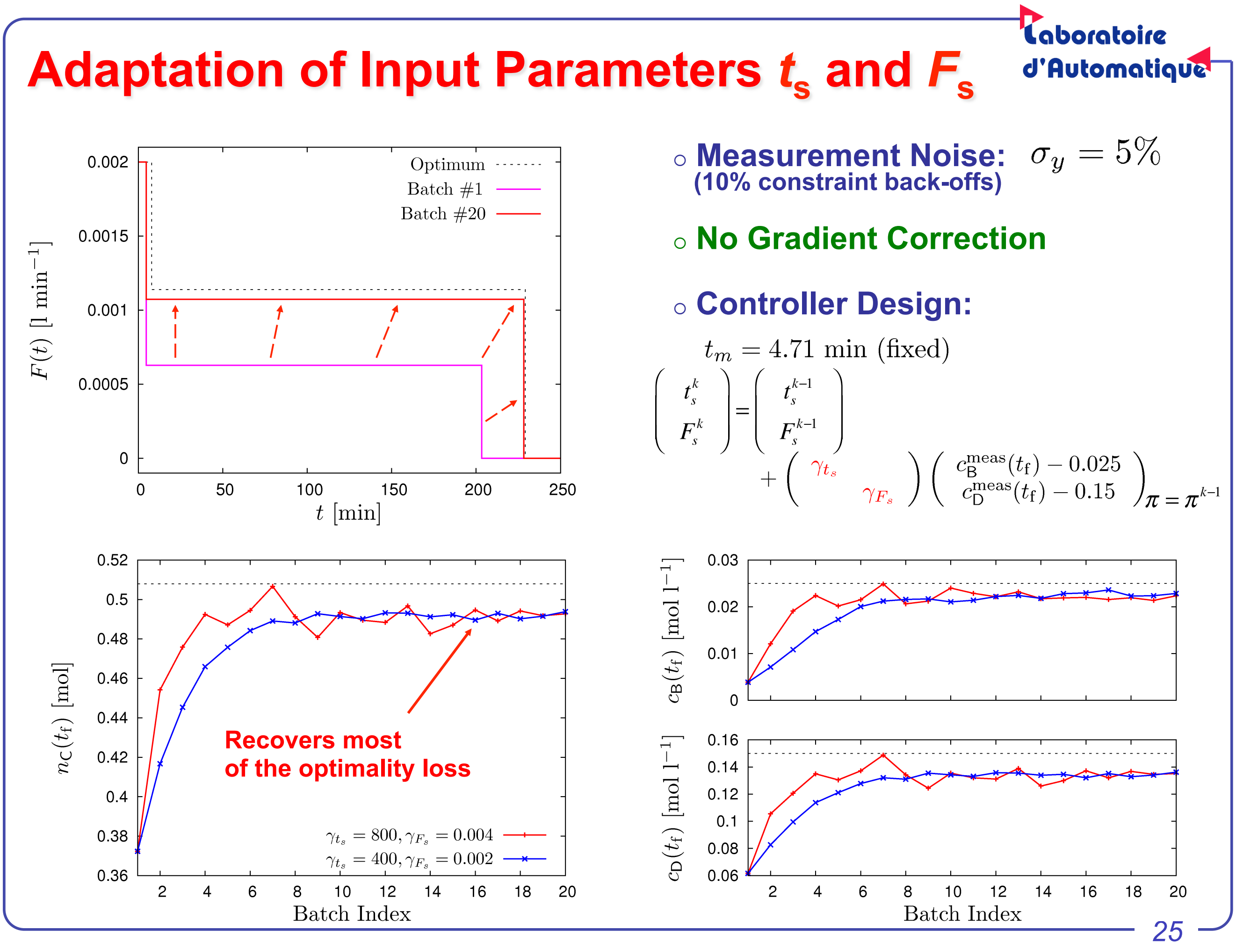
Adaptation of Modifiers ε_G

Laboratoire d'Automatique

- ○ **Measurement Noise:** $\sigma_y = 5\%$
  (10% constraint backoffs)
- ○ **No Gradient Correction**
- ○ **Exponential Filter for Modifiers:**

$$\begin{pmatrix} \varepsilon_{G,1}^{i} \\ \varepsilon_{G,2}^{i} \end{pmatrix} = (1-\gamma_G)\begin{pmatrix} \varepsilon_{G,1}^{i-1} \\ \varepsilon_{G,2}^{i-1} \end{pmatrix} + \gamma_G \begin{pmatrix} c_{\mathrm{B}}^{\mathrm{meas}}(t_{\mathrm{f}}) - c_{\mathrm{B}}(t_{\mathrm{f}}) \\ c_{\mathrm{D}}^{\mathrm{meas}}(t_{\mathrm{f}}) - c_{\mathrm{D}}(t_{\mathrm{f}}) \end{pmatrix}_{\boldsymbol{\pi}=\boldsymbol{\pi}^{i-1}}$$

Recovers most of the optimality loss



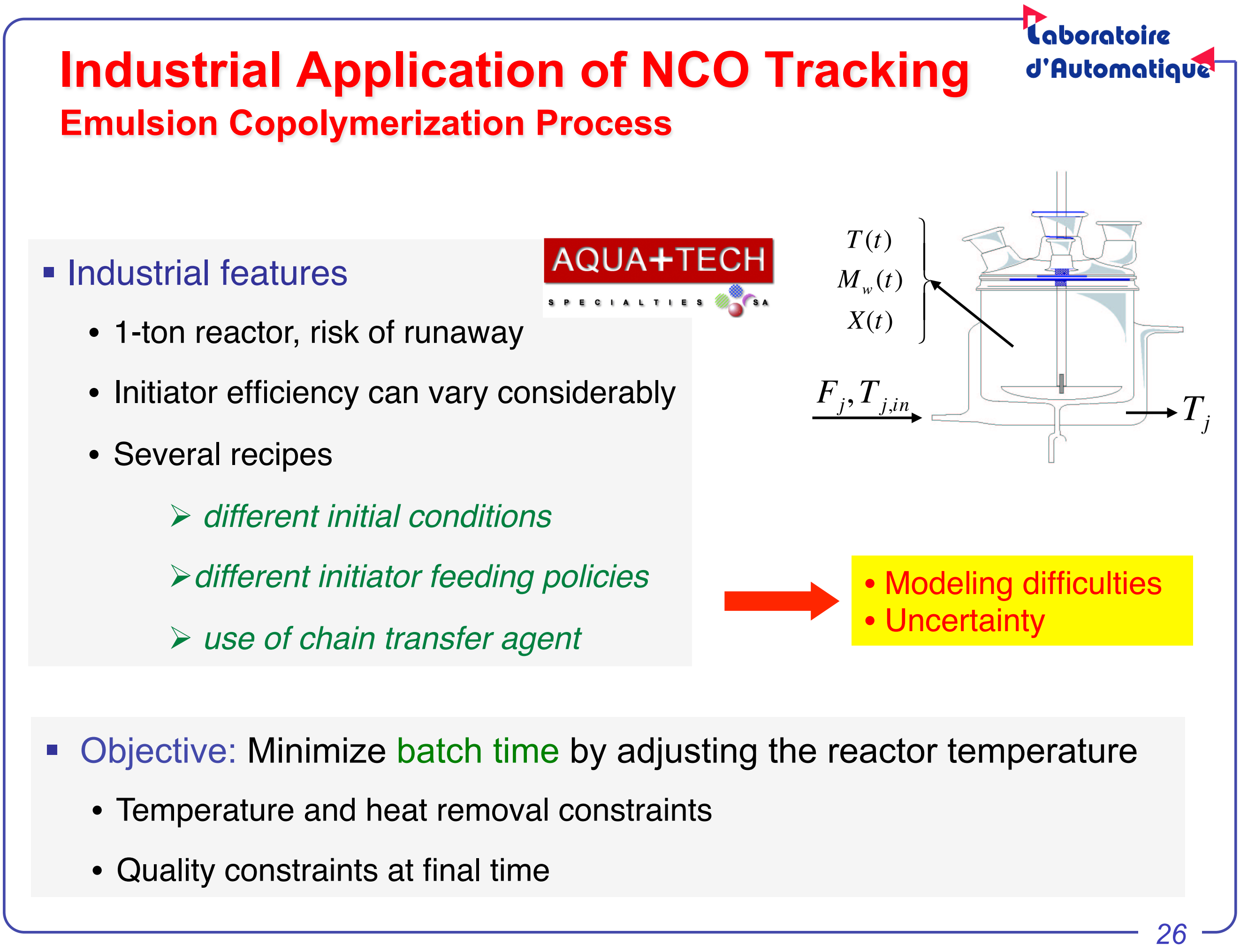
Adaptation of Input Parameters t_s and F_s

Recovers most
of the optimality loss

- **Measurement Noise:** $\sigma_y = 5\%$
  (10% constraint back-offs)
- **No Gradient Correction**
- **Controller Design:**

$t_m = 4.71$ min (fixed)

$$\begin{pmatrix} t_s^k \\ F_s^k \end{pmatrix} = \begin{pmatrix} t_s^{k-1} \\ F_s^{k-1} \end{pmatrix} + \begin{pmatrix} \gamma_{t_s} & \\ & \gamma_{F_s} \end{pmatrix} \begin{pmatrix} c_{\mathrm{B}}^{\mathrm{meas}}(t_{\mathrm{f}}) - 0.025 \\ c_{\mathrm{D}}^{\mathrm{meas}}(t_{\mathrm{f}}) - 0.15 \end{pmatrix}_{\pi = \pi^{k-1}}$$



Laboratoire d'Automatique

Industrial Application of NCO Tracking

Emulsion Copolymerization Process

AQUA+TECH SPECIALTIES SA

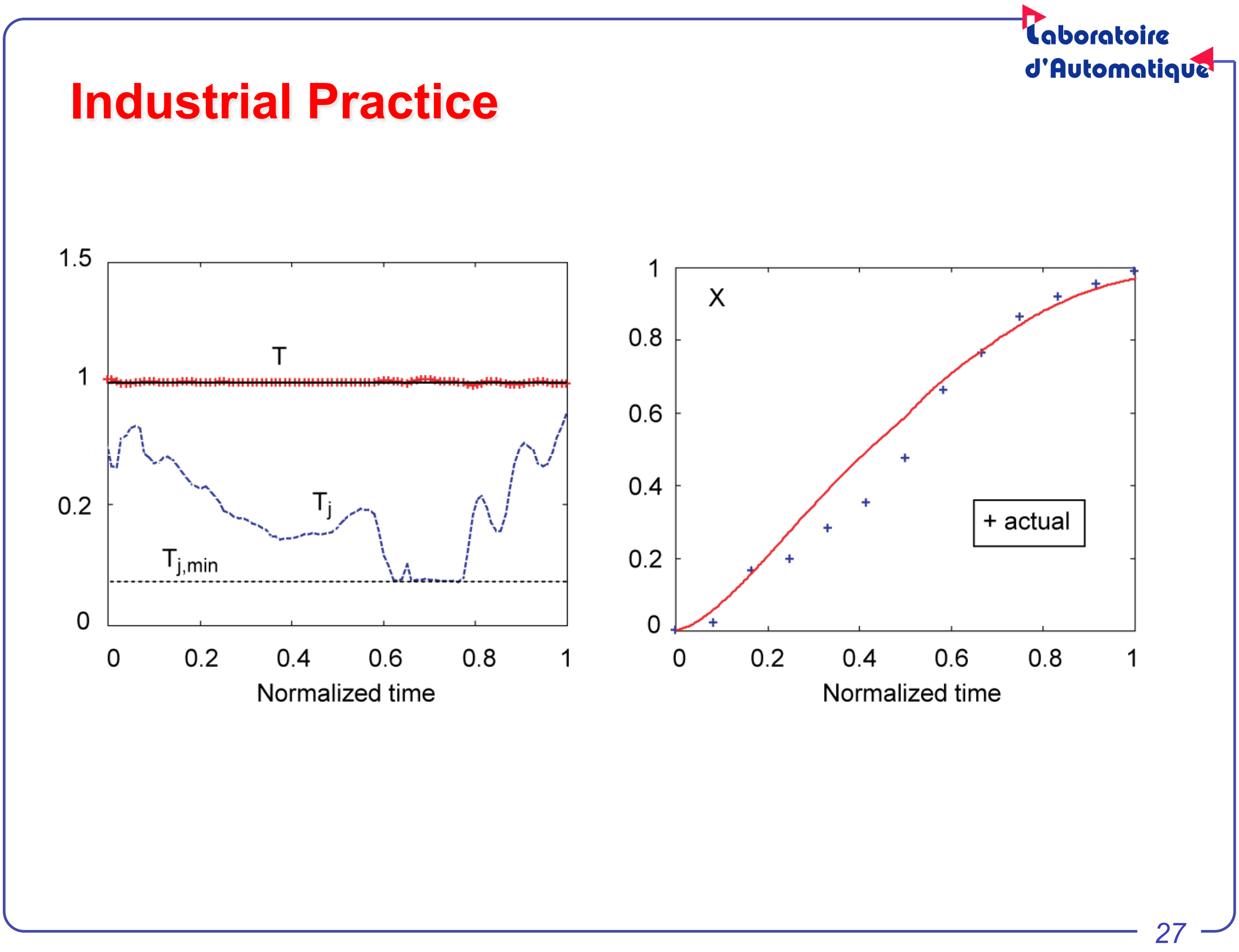
- Industrial features
  - 1-ton reactor, risk of runaway
  - Initiator efficiency can vary considerably
  - Several recipes
    - *different initial conditions*
    - *different initiator feeding policies*
    - *use of chain transfer agent*

$T(t)$
$M_w(t)$
$X(t)$
$F_j, T_{j,in}$
T_j

- Modeling difficulties
- Uncertainty

- Objective: Minimize batch time by adjusting the reactor temperature
  - Temperature and heat removal constraints
  - Quality constraints at final time

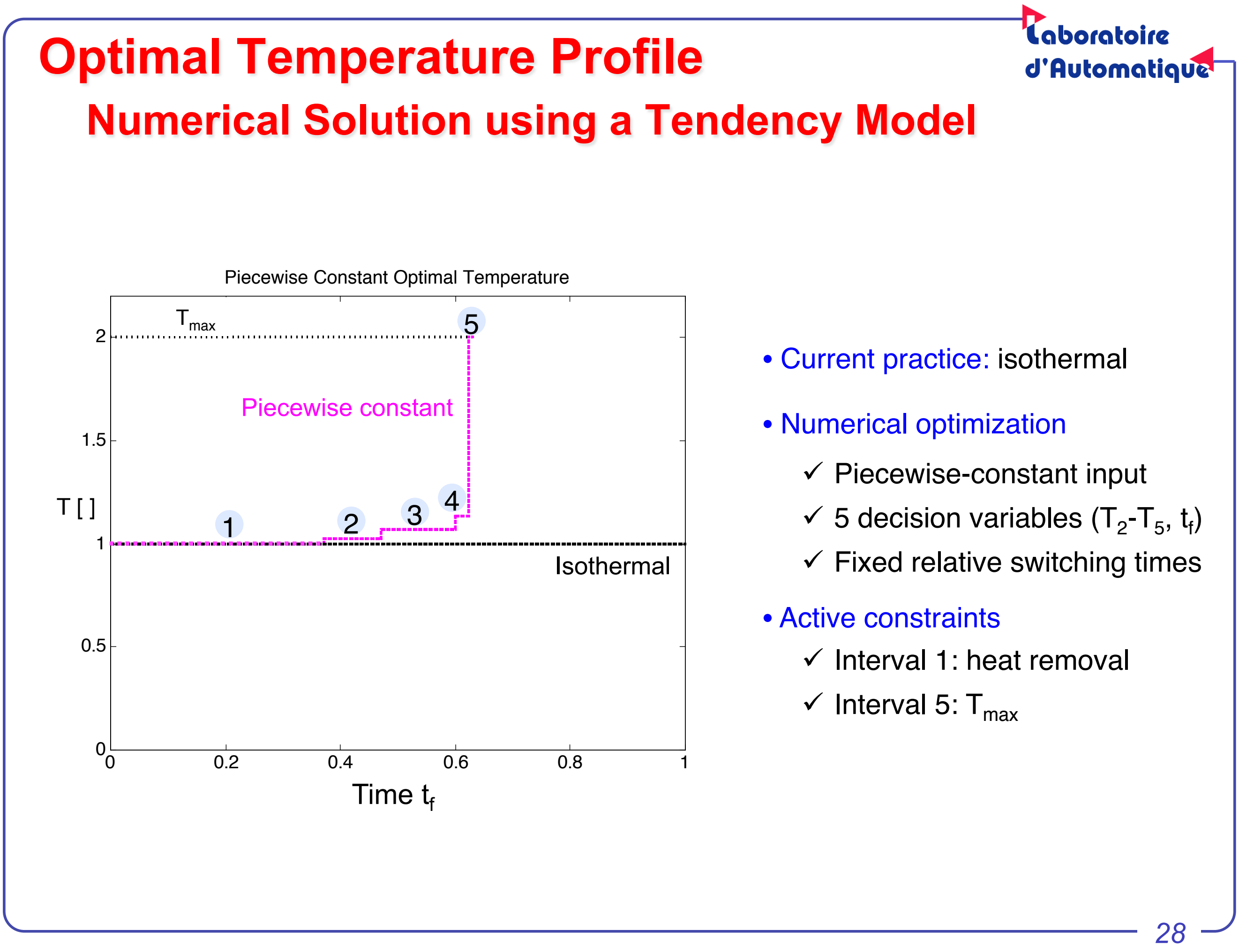
26



Laboratoire d'Automatique

Industrial Practice

27



Laboratoire d'Automatique

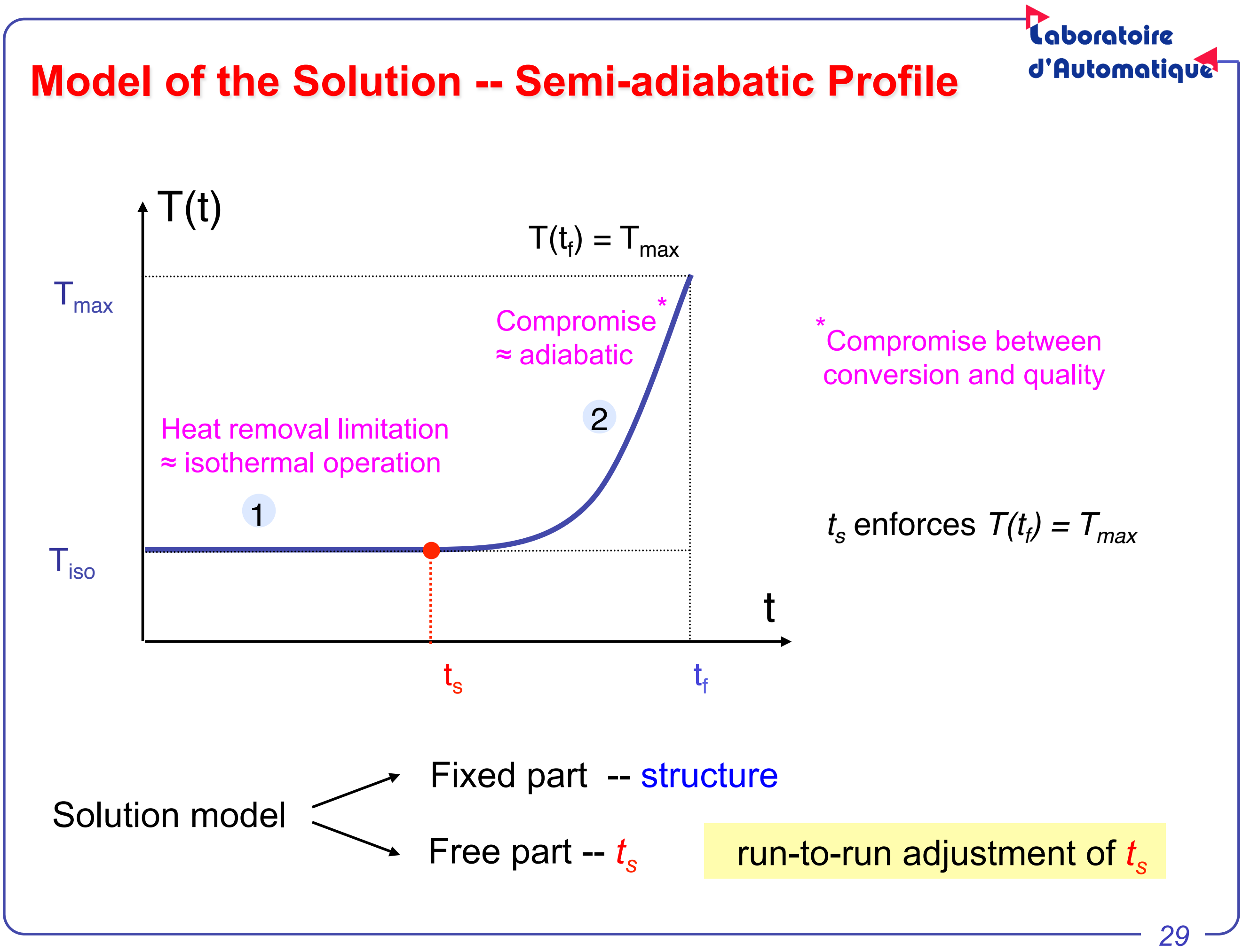
Optimal Temperature Profile

Numerical Solution using a Tendency Model

Piecewise Constant Optimal Temperature

- Current practice: isothermal
- Numerical optimization
  - ✓ Piecewise-constant input
  - ✓ 5 decision variables (T_2-T_5, t_f)
  - ✓ Fixed relative switching times
- Active constraints
  - ✓ Interval 1: heat removal
  - ✓ Interval 5: T_{max}

28



Laboratoire d'Automatique

Model of the Solution -- Semi-adiabatic Profile

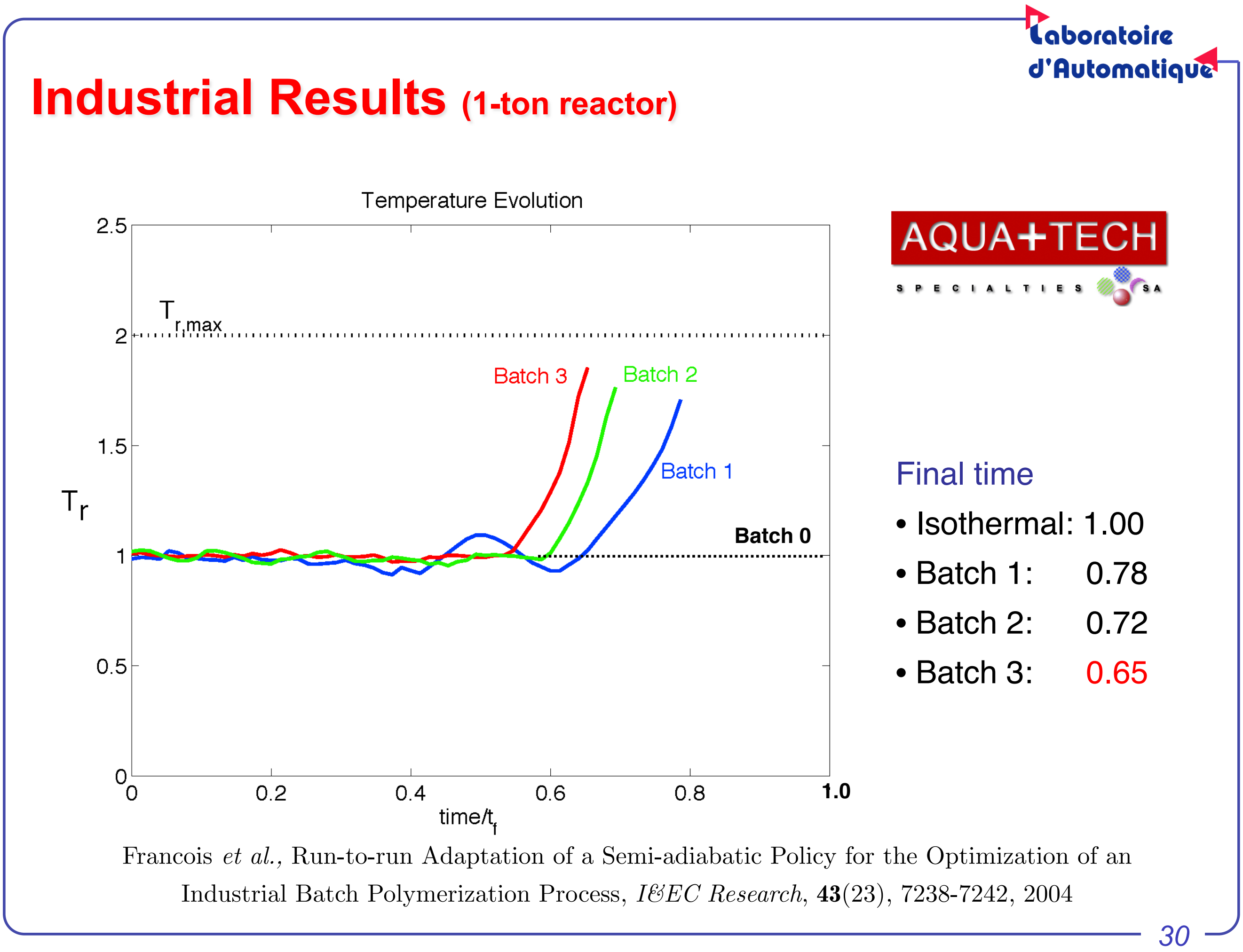
*Compromise between conversion and quality

t_s enforces $T(t_f) = T_{max}$

Solution model → Fixed part -- structure

Solution model → Free part -- t_s

run-to-run adjustment of t_s



Laboratoire d'Automatique

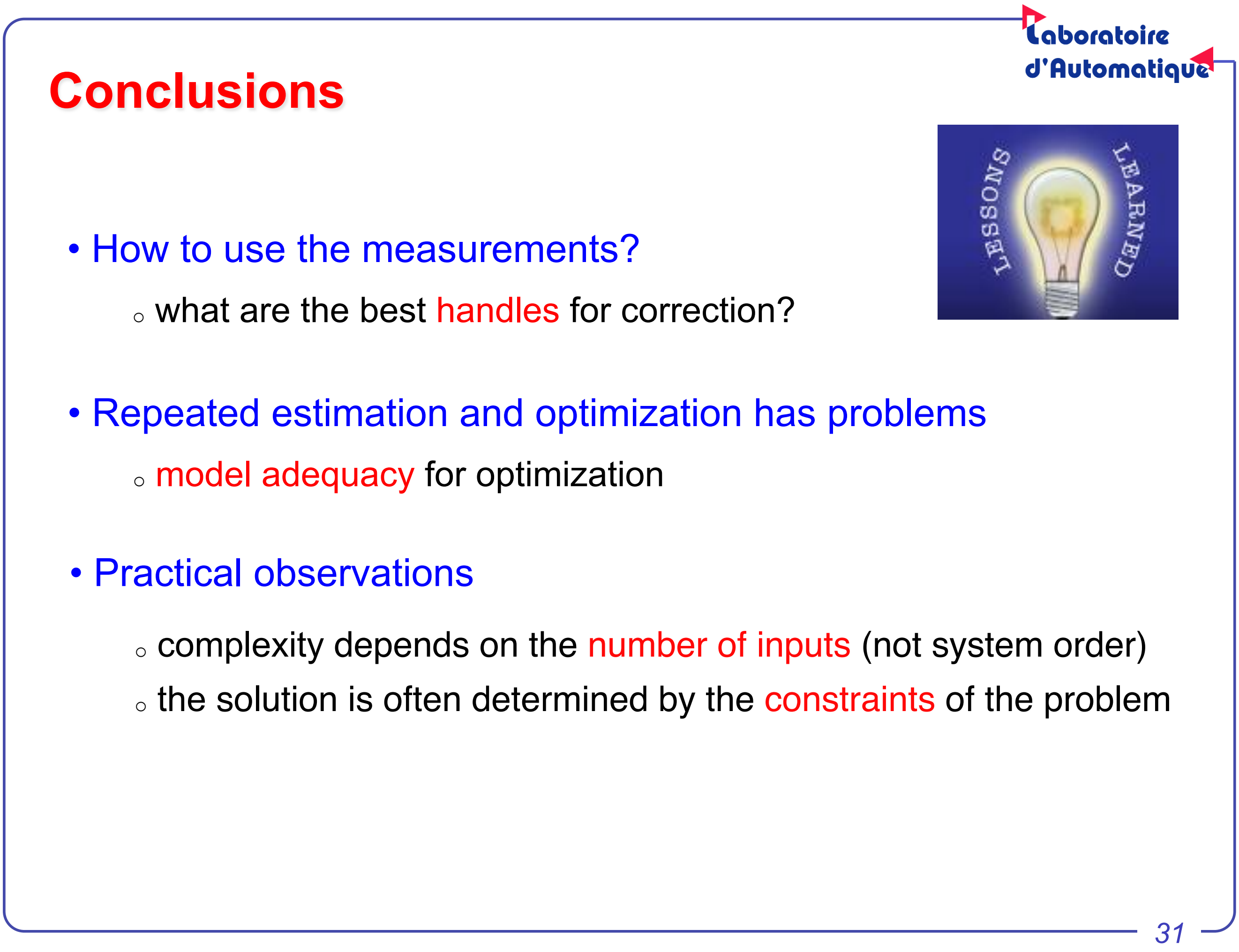
Industrial Results (1-ton reactor)

Final time

- Isothermal: 1.00
- Batch 1: 0.78
- Batch 2: 0.72
- Batch 3: 0.65

Francois *et al.*, Run-to-run Adaptation of a Semi-adiabatic Policy for the Optimization of an Industrial Batch Polymerization Process, *I&EC Research*, **43**(23), 7238-7242, 2004

30



Laboratoire
d'Automatique

Conclusions

LESSONS
LEARNED

- How to use the measurements?
  - what are the best handles for correction?
- Repeated estimation and optimization has problems
  - model adequacy for optimization
- Practical observations
  - complexity depends on the number of inputs (not system order)
  - the solution is often determined by the constraints of the problem

Regular design equations for the reduced-order Kalman filter

P. Hippe

Peter Hippe, Lehrstuhl für Regelungstechnik, Universität Erlangen-Nürnberg
(Tel : +49-9131-8527135 ; e-mail : peter.hippe@rt.eei.uni-erlangen.de)

Abstract: Reduced-order Kalman filters yield an optimal state estimate for linear dynamical systems, where parts of the outputs are not corrupted by noise. The design of such filters can either be carried out in the time domain or in the frequency domain. Different from the full-order case with all measurements corrupted by noise, the design equations of the reduced-order filter are not regular, due to the rank deficient measurement covariance matrix. This can cause problems when using standard software for the solution of the Riccati equations in the time domain. In the frequency domain spectral factorization of the non-regular polynomial matrix causes no problems. The known proof of optimality of the factorization result, however, also requires a regular measurement covariance matrix. This paper presents regular (reduced-order) design equations for such filters in the time and in the frequency domains for linear continuous-time systems and it is shown, that the existing solutions obtained by spectral factorization of a non-regular polynomial equation are indeed optimal.

Keywords: Optimal estimation, polynomials, multivariable systems, continuous-time systems.

1. INTRODUCTION

If the system is completely observable the dynamics of a state observer can be assigned arbitrarily. In the absence of disturbances the observer generates a state estimate $\hat{x}$ that converges towards the real state x of the system. In the presence of stochastic disturbances, however, persistent observation errors occur. Then, a state estimate is of interest such that the observation error $\hat{x} - x$ has the smallest mean square. Given Gaussian white noise with zero mean, such an estimate is generated by a stationary Kalman filter (Anderson and Moore (1979), Kwakernaak and Sivan (1972)) whose order coincides with the order n of the plant.

If parts of the measurements are not corrupted by noise, the order of the optimal filter is reduced. The optimal estimation problem in the presence of noise-free measurements is one of the well researched fields in automatic control. Since the original work of Bryson and Johansen (1965) a considerable amount of contributions has been published on the subject (see, *e.g.*, the books by Sage and Melsa (1971), Gelb (1996), O'Reilly (1983) and Hippe and Deutscher (2009), or the references in O'Reilly (1982) and Fairman and Luk (1985)). The time-domain design of the reduced-order filter amounts to solving an algebraic Riccati equation (ARE).

The equivalent frequency domain version of the reduced-order Kalman filter is parameterized by a polynomial matrix $\tilde{\bar{D}}(s)$, which can be obtained by spectral factorization of a polynomial matrix equation. This polynomial matrix equation is determined from a version of the ARE introduced by Bryson and Johansen (1965) or Gelb (1996). This Riccati equation is formulated for a full-order covariance matrix $\bar{P}$ which, however, is singular. There have been papers presenting regular reduced-order Riccati equations yielding a regular covariance matrix $\bar{P}_r$ of reduced order, but they cannot be used to develop an equivalent frequency domain formulation of the filtering problem.

Standard software cannot be used to design the reduced-order Kalman filter, because the basic requirement, namely a measurement covariance matrix which is positive definite, is not fulfilled in the presence of undisturbed measurements. To obtain a well-defined order of the reduced-order filter it is assumed here that the random signals, which disturb the artificial output consisting of the noisy measurements and the time derivatives of the undisturbed outputs, have a regular covariance. This is a standard assumption in nearly all investigations on reduced-order Kalman filters (see, *e.g.*, Bryson and Johansen (1965), O'Reilly (1983), Haddad and Bernstein (1987), Hippe (1989)).

After a formulation of the underlying problem in the time domain in Section 2 the existing solution for the optimal filter is presented. By a reformulation of the Riccati equation for the artificial output, one obtains a regular measurement covariance. In the continuous-time case standard software still does not work because the Hamiltonian of this ARE has eigenvalues at $s = 0$. By an adequate state transformation of the state equations of the system this Riccati equation can be subdivided into a regular part and a vanishing part. The regular part is solvable by standard software. This regular part also allows to derive the conditions for the optimal filter to be stable, and it is shown how these conditions translate into conditions on the original system.

The known polynomial matrix equation for the design of the reduced-order Kalman filter in the frequency domain is based on the left MFD of the full-order system whereas the polynomial matrix $\tilde{\bar{D}}(s)$, resulting from the spectral factorization of this polynomial matrix characterizes a system of reduced order. This is a consequence of the rank deficient measurement covariance matrix multiplying the denominator matrix of the system. Unfortunately, proofs for the optimality of the spectral factor are only known in the case, where the measurement covariance is not singular. In Hippe and Deutscher (2009) it has been observed that, on the one hand, optimality of the result can only be checked by computing the corresponding time domain results and, on the other hand, that all examples investigated so far have shown that the resulting $\tilde{\bar{D}}(s)$ is indeed optimal.

In Section 3 it is shown, that the polynomial matrix $\tilde{\bar{D}}(s)$ resulting from the non-regular polynomial equation is identical to that, which can be obtained from a "regular" polynomial matrix equation. This regular polynomial matrix equation is derived from the reduced regular ARE in the time domain and it allows the design of a full-order filter for a reduced-order system. As an additional result, the conditions for the stability of the filter are presented.

Concluding remarks are presented in Section 4

2. THE FILTER DESIGN IN THE TIME DOMAIN

We consider linear time-invariant systems of the order n, with p inputs u, q stochastic inputs w and m measured outputs y, where the first $m-\kappa$ outputs y_1 are corrupted by noise and the remaining κ outputs y_2 are free of noise, described by

$$\dot{x}(t) = Ax(t) + Bu(t) + Gw(t) \tag{1}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x(t) + \begin{bmatrix} v_1(t) \\ 0 \end{bmatrix} \tag{2}$$

where the abbreviation

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = C \tag{3}$$

will be used in the sequel. It is assumed that the system is controllable both from the input u and from the input w and that it is observable.

The stochastic inputs $w \in \mathbb{R}^q$ and $v_1 \in \mathbb{R}^{m-\kappa}$ are independent, zero-mean, stationary Gaussian white noises with

$$\mathrm{E}\{w(t)w^T(\tau)\} = \bar{Q}\delta(t-\tau) \tag{4}$$

$$\mathrm{E}\{v_1(t)v_1^T(\tau)\} = \bar{R}_1\delta(t-\tau) \tag{5}$$

where $\mathrm{E}\{\cdot\}$ denotes the mathematical expectation and $\delta(t)$ is the Dirac delta function.

The covariance matrices $\bar{Q}$ and $\bar{R}_1$ are real and symmetric, where $\bar{Q}$ is positive-semidefinite and $\bar{R}_1$ is positive-definite. The initial state $x(0) = x_0$ is not correlated with the disturbances, *i.e.*, $\mathrm{E}\{x_0 w^T(t)\} = 0$ and $\mathrm{E}\{x_0 v_1^T(t)\} = 0$ for all $t \geq 0$.

It is assumed that the covariance matrix

$$\Phi = C_2 G\bar{Q}G^T C_2^T = G_2\bar{Q}G_2^T \tag{6}$$

is positive definite. It characterizes the influence of the input noise on the time derivative of the undisturbed measurement y_2.

The reduced-order Kalman filter for such systems is described by

$$\begin{aligned} \dot{\hat{\zeta}}(t) = {} & T(A - L_1C_1)\Theta\hat{\zeta}(t) + \\ & [TL_1 \quad T(A - L_1C_1)\Psi_2]\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + TBu(t) \end{aligned} \tag{7}$$

$$\hat{x}(t) = \Theta\hat{\zeta}(t) + \Psi_2 y_2(t) \tag{8}$$

(see Gelb (1996), Hippe and Deutscher (2009)). The optimal estimate $\hat{\zeta}(t)$ results if the matrices L_1 and Ψ_2 are chosen such that

$$L_1 = \bar{P}C_1^T\bar{R}_1^{-1} \tag{9}$$

and

$$\Psi_2 = \left(\bar{P}A^TC_2^T + G\bar{Q}G^TC_2^T\right)\Phi^{-1} \tag{10}$$

with Φ as in (6) and $\bar{P} = \bar{P}(\infty)$ defined by

$$\bar{P}(t) = \mathrm{E}\{(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T\} \tag{11}$$

The stationary covariance $\bar{P}$ satisfies the ARE

$$\begin{aligned} A\bar{P} + \bar{P}A^T - [\, L_1 \quad \Psi_2\,]\begin{bmatrix} \bar{R}_1 & 0 \\ 0 & \Phi \end{bmatrix}\begin{bmatrix} L_1^T \\ \Psi_2^T \end{bmatrix} + \\ G\bar{Q}G^T = 0 \end{aligned} \tag{12}$$

(Hippe and Deutscher (2009)) which is the basis for deriving the equivalent frequency-domain solution (see Section 3). This ARE, however, is not in a standard form to be solved for $\bar{P}$.

Inserting the optimal solutions (9) and (10) in (12) one obtains

$$\tilde{A}\bar{P} + \bar{P}\tilde{A}^T - \bar{P}\tilde{C}^T\tilde{R}^{-1}\tilde{C}\bar{P} + G\tilde{Q}G^T = 0 \tag{13}$$

with

$$\tilde{A} = A - G\bar{Q}G^TC_2^T\Phi^{-1}C_2A \tag{14}$$

$$\tilde{C} = \begin{bmatrix} C_1 \\ C_2A \end{bmatrix} \tag{15}$$

$$\tilde{R} = \begin{bmatrix} \bar{R}_1 & 0 \\ 0 & \Phi \end{bmatrix} \tag{16}$$

and

$$\tilde{Q} = \bar{Q} - \bar{Q}G^TC_2^T\Phi^{-1}C_2G\bar{Q} \tag{17}$$

The ARE (13) is in the standard form with a regular $\tilde{R} > 0$. Standard software as, *e.g.*, the function *lqe* in MATLAB®, however, does not yield the solution $\bar{P}$, because the Hamiltonian related to the ARE (13) has eigenvalues at $s = 0$. This is due to the fact that $\operatorname{rank}\bar{P} = n - \kappa$.

By a regular state transformation $z(t) = \bar{T}x(t)$ with

$$\bar{T} = \begin{bmatrix} C \\ * \end{bmatrix} \tag{18}$$

the state equations (1)–(2) of the system can always be transformed into

$$\dot z(t) = \bar A z(t) + \bar B u(t) + \bar G w(t) \tag{19}$$

$$y(t) = \bar C z(t) + \begin{bmatrix} v_1(t) \\ 0 \end{bmatrix} \tag{20}$$

with

$$\bar A = \bar T A \bar T^{-1},\ \bar B = \bar T B,\ \bar G = \bar T G,\ \bar C = C \bar T^{-1} \tag{21}$$

or in components

$$\begin{bmatrix} \dot z_1 \\ \dot z_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{22} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u + \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} w \tag{22}$$

$$y_1 = \bar C_1 z_1 + v_1 \tag{23}$$

$$y_2 = z_2 \tag{24}$$

with $z_1 \in \mathbb{R}^{(n-\kappa)}$, $0 < \kappa \leq m$, $z_2 \in \mathbb{R}^{\kappa}$.

If the transformed matrices (21) are inserted in (13)–(17) the solution $\bar P_z = \bar T \bar P \bar T^{-1}$ of this modified ARE (13) has the form

$$\bar P_z = \begin{bmatrix} \bar P_r & 0 \\ 0 & 0_\kappa \end{bmatrix} \tag{25}$$

and the ARE (13) then consists of a regular (upper left) part

$$A_r \bar P_r + \bar P_r A_r^T - \bar P_r C_r^T \tilde R^{-1} C_r \bar P_r + G_r \tilde Q G_r^T = 0 \tag{26}$$

while the rest is vanishing. The matrices in (26) are defined by

$$A_r = A_{11} - G_1 \bar Q G_2^T \Phi^{-1} A_{21} \tag{27}$$

$$G_r = G_1 \tag{28}$$

and

$$C_r = \begin{bmatrix} \bar C_1 \\ A_{21} \end{bmatrix} \tag{29}$$

so that the reduced-order Kalman filter can be regarded as a regular full-order filter for the reduced system (A_r, G_r, C_r). The feedback matrix L_r is defined by

$$L_r = \bar P_r C_r^T \tilde R^{-1} = \bar P_r \begin{bmatrix} \bar C_1^T \bar R_1^{-1} & A_{21}^T \Phi^{-1} \end{bmatrix} \tag{30}$$

The ARE (26) has two advantages. First, it can be used to obtain $\bar P_r$ and consequently also $\bar P$ by standard software. Second, it defines the conditions which guarantee a stable filter. It is known that the full-order Kalman filter for the reduced system (A_r, G_r, C_r) is stable if the pair $(A_r, G_r \tilde Q_0)$ has no uncontrollable eigenvalues on the imaginary axis, where

$$\tilde Q = \tilde Q_0\, \tilde Q_0^T \tag{31}$$

(Goodwin et al. (2001)). Introducing

$$\bar Q = \bar Q_0\, \bar Q_0^T \tag{32}$$

and

$$\hat Q = I - \bar Q_0^T G_2^T \Phi^{-1} G_2 \bar Q_0 \tag{33}$$

it is easy to show that

$$\tilde Q_0 = \bar Q_0\, \hat Q \tag{34}$$

when taking into account that $C_2 G = G_2$. Given the above condition for a stable filter in terms of A_r and G_r, it is of interest to know the corresponding condition for the non-reduced system $(\bar A, \bar G, \bar C)$. The answer is contained in the following lemma.

Lemma 1. If the system

$$\dot z(t) = \bar A z(t) + \bar G \bar Q_0 w(t) \tag{35}$$

$$y_2(t) = [0 \quad I_\kappa]\, z(t) \tag{36}$$

has no zeros which are located on the imaginary axis, then the pair $(A_r, G_r \tilde Q_0)$ has no uncontrollable eigenvalues on the imaginary axis and *vice versa.*

Proof: If $s = s_i$ is a non-controllable eigenvalue of the pair $(A_r, G_r \tilde Q_0)$ then

$$\operatorname{rank} \begin{bmatrix} s_i I - A_r & \vdots & G_r \tilde Q_0 \end{bmatrix} < n - \kappa \tag{37}$$

(see, *e.g.*, Kailath (1980)).

Now define the system matrix

$$P(s) = \begin{bmatrix} sI_{n-\kappa} - A_{11} & -A_{12} & G_1 \bar Q_0 \\ -A_{21} & sI_\kappa - A_{22} & G_2 \bar Q_0 \\ 0 & -I_\kappa & 0 \end{bmatrix} \tag{38}$$

which characterizes the zeros of the system (35)–(36) (see Rosenbrock (1970)).

If the system (35)–(36) has a zero at $s = s_i$, then $\operatorname{rank} P(s = s_i) < n + \kappa$.

Using the unimodular matrix

$$U_L = \begin{bmatrix} I_{n-\kappa} & -G_1 \bar Q G_2^T \Phi^{-1} & 0 \\ 0 & I_\kappa & 0 \\ 0 & 0 & I_\kappa \end{bmatrix} \tag{39}$$

and the unimodular matrix

$$U_R = \begin{bmatrix} I_{n-\kappa} & 0 & 0 \\ 0 & I_\kappa & 0 \\ \bar Q_0^T G_2^T \Phi^{-1} A_{21} & 0 & I_q \end{bmatrix} \tag{40}$$

one obtains

$$U_L P(s = s_i) U_R = \begin{bmatrix} s_i I - A_r & * & G_r \tilde Q_0 \\ 0 & * & G_2 \bar Q_0 \\ 0 & -I_\kappa & 0 \end{bmatrix} \tag{41}$$

Since it has been assumed that $\operatorname{rank} G_2 \bar Q_0 = \kappa$ (see (6)) this shows that the system (35)–(36) has a zero at $s = s_i$ if and only if $s = s_i$ is an uncontrollable eigenvalue in the pair $(A_r, G_r \tilde Q_0)$ and *vice versa.* This is, of course, not only true for the system (35)–(36) but also for the system $(A, G\bar Q_0, C_2)$.

3. THE FILTER DESIGN IN THE FREQUENCY DOMAIN

In the frequency domain, the system (1)–(2) or (19)–(20) is described by

$$y(s) = F(s)w(s) + \begin{bmatrix} v_1(s) \\ 0 \end{bmatrix} \tag{42}$$

with

$$F(s) = \bar{C}(sI - \bar{A})^{-1}\bar{G} = C(sI - A)^{-1}G \tag{43}$$

Given the left coprime MFD

$$F(s) = \bar{D}^{-1}(s)\bar{N}_w(s) \tag{44}$$

the reduced-order Kalman filter is parameterized by the polynomial matrix $\tilde{\bar{D}}(s)$ resulting by spectral factorization of the right hand side of

$$\tilde{\bar{D}}(s)\tilde{R}\tilde{\bar{D}}^T(-s) = \tag{45}$$
$$\bar{D}(s)\begin{bmatrix} \bar{R}_1 & 0 \\ 0 & 0 \end{bmatrix}\bar{D}^T(-s) + \bar{N}_w(s)\bar{Q}\bar{N}_w^T(-s)$$

where

$$\Gamma_r\left[\tilde{\bar{D}}(s)\right] = \Gamma_r\left[\bar{D}_\kappa(s)\right] \tag{46}$$

with the row-reduced polynomial matrix

$$\bar{D}_\kappa(s) = \Pi\left\{\bar{D}(s)\begin{bmatrix} I_{m-\kappa} & 0 \\ 0 & s^{-1}I_\kappa \end{bmatrix}\right\} \tag{47}$$

(see Hippe and Deutscher (2009)). Here, $\Gamma_r[\cdot]$ denotes the highest row-degree-coefficient matrix and $\Pi[\cdot]$ taking the polynomial part.

The polynomial matrix $\tilde{\bar{D}}(s)$ is related with the time domain parameters by

$$\bar{D}^{-1}(s)\tilde{\bar{D}}(s) = \tag{48}$$
$$\bar{C}(sI - \bar{A})^{-1}[\bar{L}_1 \quad \bar{\Psi}_2] + \begin{bmatrix} I_{m-\kappa} & 0 \\ 0 & 0_\kappa \end{bmatrix}$$

where

$$\bar{L}_1 = \begin{bmatrix} \bar{P}_r\bar{C}_1^T \\ 0 \end{bmatrix}\bar{R}_1^{-1} \tag{49}$$

and

$$\bar{\Psi}_2 = \begin{bmatrix} \left(\bar{P}_r A_{21}^T + G_1\bar{Q}G_2^T\right)\Phi^{-1} \\ I_\kappa \end{bmatrix} \tag{50}$$

In Hippe and Deutscher (2009) the solution (45)–(47) is presented without rigorous proof, because the polynomial matrix (45) contains a singular measurement covariance at its right hand side and the known proofs of optimality of a $\tilde{\bar{D}}(s)$ obtained by spectral factorization are based on a full-order filter with a regular measurement covariance matrix.

The polynomial matrix (45) was derived on the basis of the ARE (12). As shown in Section 2, the reduced-order Kalman filter can also be designed on the basis of the regularized ARE (26), using a system description (A_r, C_r, G_r) of reduced order $n - \kappa$ and a regular measurement covariance matrix $\tilde{R}$.

Introducing the left coprime MFD of

$$F_r(s) = C_r(sI - A_r)^{-1}G_r \tag{51}$$

namely

$$F_r(s) = \bar{D}_r^{-1}(s)\bar{N}_{wr}(s) \tag{52}$$

and the polynomial matrix $\tilde{\bar{D}}_r(s)$ parameterizing the reduced-order Kalman filter related to the parameters $(A_r, G_r, C_r, \bar{P}_r)$ according to

$$\bar{D}_r^{-1}(s)\tilde{\bar{D}}_r(s) = C_r(sI - A_r)^{-1}L_r + I_m \tag{53}$$

the Riccati equation (26) can be transformed into the polynomial matrix equation

$$\tilde{\bar{D}}_r(s)\tilde{R}\tilde{\bar{D}}_r^T(-s) = \tag{54}$$
$$\bar{D}_r(s)\tilde{R}\bar{D}_r^T(-s) + \bar{N}_{wr}(s)\tilde{Q}\bar{N}_{wr}^T(-s)$$

by similar steps as in the derivation of (45) from (12) in Hippe and Deutscher (2009). This is a regular polynomial matrix equation with $\tilde{R} > 0$ and consequently the polynomial matrix $\tilde{\bar{D}}_r(s)$ obtained by spectral factorization of the right hand side of (54) with

$$\Gamma_r\left[\tilde{\bar{D}}_r(s)\right] = \Gamma_r\left[\bar{D}_r(s)\right] \tag{55}$$

parameterizes the optimal full-order Kalman filter for the reduced-order system (52) in the frequency domain.

If this $\tilde{\bar{D}}_r(s)$ is identical with $\tilde{\bar{D}}(s)$ obtained from the spectral factorization of (45), it follows that the solution procedure presented in Hippe and Deutscher (2009) yields indeed the optimal results.

Given the MFD (44), define the MFD

$$\bar{C}(sI - \bar{A})^{-1} = \bar{D}^{-1}(s)\bar{N}_z(s) \tag{56}$$

with $\bar{N}_z(s)$ partitioned according to

$$\bar{N}_z(s) = \begin{bmatrix} \bar{N}_{z1}(s) & \bar{N}_{z2}(s) \end{bmatrix} \tag{57}$$

where $\bar{N}_{z1}(s)$ has $n-\kappa$ columns and $\bar{N}_{z2}(s)$ has κ columns.

Theorem 1. The polynomial matrix $\tilde{\bar{D}}_r(s)$ resulting from (54) is identical with $\tilde{\bar{D}}(s)$ resulting from (45) if the polynomial matrices in the MFD (52) are chosen as

$$\bar{N}_{wr}(s) = \bar{N}_{z1}(s)G_1 \tag{58}$$

and

$$\bar{D}_r(s) = \tag{59}$$
$$\begin{bmatrix} \bar{N}_{z1}(s) & \bar{N}_{z2}(s) \end{bmatrix}\begin{bmatrix} 0_{n-\kappa,m-\kappa} & G_1\bar{Q}G_2^T\Phi^{-1} \\ 0_{\kappa,m-\kappa} & I_\kappa \end{bmatrix} +$$
$$\bar{D}(s)\begin{bmatrix} I_{m-\kappa} & 0 \\ 0 & 0_\kappa \end{bmatrix}$$

The polynomial matrix $\tilde{\bar{D}}(s) = \tilde{\bar{D}}_r(s)$ parameterizes a stable filter, if the pair

$$\left(\bar{D}(s)\begin{bmatrix} I_{m-\kappa} & 0 \\ 0 & 0_\kappa \end{bmatrix}, \ \bar{N}_w(s)\bar{Q}_0\right) \tag{60}$$

has no greatest common left devisor with zeros on the imaginary axis.

Proof: From (51), (52), (58) and (28) follows

$$C_r(sI - A_r)^{-1} = \bar{D}_r^{-1}(s)\bar{N}_{z1}(s) \tag{61}$$

As a consequence of the rearranged form of (56), namely $\bar{D}(s)\bar{C} = \bar{N}_z(s)(sI - \bar{A})$, together with (22) – (24) one obtains

$$\bar{D}(s)\begin{bmatrix}\bar{C}_1\\0\end{bmatrix} = \bar{N}_{z1}(s)(sI - A_{11}) - \bar{N}_{z2}(s)A_{21} \tag{62}$$

This allows to show that $\bar{N}_{z1}(s)(sI-A_r) = \bar{D}_r(s)C_r$ which then proves that the pair (58) and (59) constitutes a left MFD of (51).

Inserting (58) and (59) in (54) it is straightforward to show, that the right hand sides of the polynomial equations (45) and (54) coincide, so that $\tilde{\bar{D}}(s)\tilde{R}\tilde{\bar{D}}^T(-s) = \tilde{\bar{D}}_r(s)\tilde{R}\tilde{\bar{D}}_r^T(-s)$ and since $\tilde{R}$ is positive definite, this yields $\tilde{\bar{D}}(s) = \tilde{\bar{D}}_r(s)$.

The equality $\tilde{\bar{D}}(s) = \tilde{\bar{D}}_r(s)$ can also be installed by comparing

$$\tilde{\bar{D}}(s) = \bar{N}_z(s)\begin{bmatrix}\bar{L}_1 & \bar{\Psi}_2\end{bmatrix} + \bar{D}(s)\begin{bmatrix}I_{m-\kappa} & 0\\0 & 0_\kappa\end{bmatrix} \tag{63}$$

which results from (48) and

$$\tilde{\bar{D}}_r(s) = \bar{N}_{z1}(s)L_r + \bar{D}_r(s) \tag{64}$$

which results from (53) and then using (30), (49), (50) and (59). This proves the first part of the theorem.

Since on the right hand side of (54) the measurement covariance term $\tilde{R}$ is regular, the full-order filter for the reduced system (A_r, G_r, C_r) is stable if the pair

$$\left(\bar{D}_r(s)\ ,\ \bar{N}_{wr}(s)\tilde{Q}_0\right) \tag{65}$$

has no common greatest left devisor $U_L(s)$ with zeros on the imaginary axis (Goodwin et al. (2001)).

Two polynomial matrices are relatively left coprime if they meet the Bezout identity. If they contain a non-unimodular greatest common left devisor $U_L(s)$, the identity matrix is replaced by $U_L(s)$ (Kailath (1980)).

If the pair (65) contains a non-unimodular greatest common left devisor $U_L(s)$ there exist solutions $\bar{Y}_{0r}(s)$ and $\bar{X}_{0r}(s) = \begin{bmatrix}\bar{X}_{0r1}(s)\\\bar{X}_{0r2}(s)\end{bmatrix}$ of the Diophantine equation

$$\bar{N}_{z1}(s)G_1\tilde{Q}_0\bar{Y}_{0r}(s) + \bar{D}_r(s)\begin{bmatrix}\bar{X}_{0r1}(s)\\\bar{X}_{0r2}(s)\end{bmatrix} = U_L(s) \tag{66}$$

(see, *e.g.*, Hippe and Deutscher (2009)). If, on the other hand, the pair (60) contains a non-unimodular greatest common left devisor $U_L(s)$ there exist solutions $\bar{Y}_0(s)$ and $\bar{X}_0(s) = \begin{bmatrix}\bar{X}_{01}(s)\\\bar{X}_{02}(s)\end{bmatrix}$ of the Diophantine equation

$$\begin{aligned}&\begin{bmatrix}\bar{N}_{z1}(s)G_1 + \bar{N}_{z2}(s)G_2\end{bmatrix}\bar{Q}_0\bar{Y}_0(s) + \\ &\qquad \bar{D}(s)\begin{bmatrix}I & 0\\0 & 0_\kappa\end{bmatrix}\begin{bmatrix}\bar{X}_{01}(s)\\\bar{X}_{02}(s)\end{bmatrix} = U_L(s)\end{aligned} \tag{67}$$

Given the solutions $\bar{Y}_{0r}(s)$ and $\bar{X}_{0r}(s)$ of (66) the polynomial matrices

$$\bar{X}_{01}(s) = \bar{X}_{0r1}(s) \tag{68}$$

$$\bar{X}_{02}(s) = 0 \tag{69}$$

and

$$\bar{Y}_0(s) = \hat{Q}\bar{Y}_{0r}(s) + \bar{Q}_0^T G_2^T \Phi^{-1}\bar{X}_{0r2}(s) \tag{70}$$

solve the equation (67).

Given the solutions $\bar{Y}_0(s)$ and $\bar{X}_0(s)$ of (67) the polynomial matrices

$$\bar{X}_{01r}(s) = \bar{X}_{01}(s) \tag{71}$$

$$\bar{X}_{0r2}(s) = G_2\bar{Q}_0\bar{Y}_0(s) \tag{72}$$

and

$$\bar{Y}_{0r}(s) = \hat{Q}\bar{Y}_0(s) \tag{73}$$

solve the equation (66). This shows that, if the pair (65) does not contain a greatest common left devisor with zeros on the imaginary axis, then also the pair (60) does not contain such a greatest common left devisor and *vice versa*. This proves the second part of the theorem.

4. CONCLUSIONS

Some open problems in the design of reduced-order Kalman filters for linear continuous-time systems have been solved. Due to the noise-free measurements, the measurement covariance matrix becomes singular and therefore, standard software cannot be used to solve the ARE of the reduded-order filter. By defining an artificial output of the system, a form of the ARE can be obtained which exhibits a regular measurement covariance matrix. However, also this form is not solvable by the standard routines, as the corresponding Hamiltonian has eigenvalues at $s = 0$. By using an appropriate state transformation on the original system, this modified form of the ARE can be subdivided into a regular part and a vanishing part. The regular part is readily solvable for the matrix $\bar{P}$, parameterizing the filter in the time domain, and it also characterizes the conditions which guarantee a stable filter. These conditions for the parameters of the reduced-order system have been translated into conditions for the original full-order system.

The known polynomial matrix defining the parameterizing polynomial matrix of the reduced-order filter in the frequency domain contains a singular measurement covariance matrix. This does not cause problems when applying spectral factorization to obtain the parameterizing polynomial matrix of the reduced-order filter. However, neither a proof of optimality nor a set of conditions for the stability of the filter were known so far. Based on the reduced-order model in the time domain, a regular full-order filter design for a reduced-order system also becomes possible in the frequency domain. This allows to prove optimality of the results obtained so far and it also allows to formulate the conditions on the MFD of the original full-order system that are required to obtain a stable filter.

Along similar lines as presented in this paper, regularized design equations can be derived for the discrete-time case. However, the derivation of the DARE in standard form related to the artificial output is not as straightforward as in the continuous-time case, where the optimal matrices

L_1 and Ψ_2 can simply be substituted in the ARE (12) to obtain the form (13). Different from the continuous-time case, this DARE is solvable by standard routines for the rank deficient matrix $\bar{P}$, because the eigenvalues of the corresponding Hamiltonian at $z = 0$ are now inside the stability region. However, a reduced full-order filtering problem can also be formulated for a reduced-order system of the order $n - \kappa$, and the parameters (A_r, G_r, C_r) have the same form as in the continuous-time case. Starting from this reduced-order system, a regular polynomial matrix can be derived whose spectral factorization yields a parameterizing polynomial matrix for the optimal filter. This parameterizing matrix can also be shown to coincide with the known matrix obtained by spectral factorization of the non-regular equation, provided that the MFD of the reduced system is chosen in the same way as in the continuous-time case. An additional technicality arizes due to the *a posteriori* estimate, so that an additional presentation of these results is beyond the scope of this paper.

REFERENCES

Anderson, B.D.O. and Moore, J.B. (1979). *Optimal Filtering.* Prentice Hall, Englewood Cliffs, NJ.

Bryson, A. and Johansen, D. (1965). Linear filtering for time-varying systems using measurements containing colored noise. *IEEE Transactions on Automatic Control*, 10, 4–10.

Fairman, F.W. and Luk, L. (1985). On reducing the order of Kalman filters for discrete-time stochastic systems having singular measurement noise. *IEEE Transactions on Automatic Control*, 30, 1150–1152.

Gelb, A. (1996). *Applied optimal estimation.* The MIT Press, Cambridge, MA.

Goodwin, G.C., Graebe, S.F., and Salgado, M.E. (2001). *Control system design.* Prentice Hall, Upper Saddle River, NJ.

Haddad, W.M. and Bernstein, D.S. (1987). The optimal projection equations for reduced-order state estimation: The singular measurement case. *IEEE Transactions on Automatic Control*, 32, 1135–1139.

Hippe, P. (1989). Design of reduced-order optimal estimators directly in the frequency domain. *International Journal of Control*, 50, 2599–2614.

Hippe, P. and Deutscher, J. (2009). *Design of Observer-based Compensators - From the time to the frequency domain.* Springer, Berlin Heidelberg New York London.

Kailath, T. (1980). *Linear systems.* Prentice Hall, Englewood Cliffs, NJ.

Kwakernaak, H. and Sivan, R. (1972). *Linear Optimal Control Systems.* Wiley Interscience, New York London Sidney Toronto.

O'Reilly, J. (1982). Comments on two recent papers on reduced-order optimal state estimation for linear systems with partially noise corrupted measurements. *IEEE Transactions on Automatic Control*, 27, 280–282.

O'Reilly, J. (1983). *Observers for linear systems.* Academic Press, London.

Rosenbrock, H.H. (1970). *State space and multivariable theory.* Nelson, London.

Sage, A.P. and Melsa, J.L. (1971). *Estimation theory with applications to communications and control.* McGraw Hill, New York, NY.

Nonlinear Control of a Chemical Reactor

P. Dostál, V. Bobál, J. Vojtěšek

Tomas Bata University in Zlin, Faculty of Applied Informatics
nam. T.G. Masaryka 5555, 760 01 Zlín, Czech Republic
(Tel: +420 57 6035195; e-mail:dostalp@fai.utb.cz).

Abstract: The paper deals with continuous-time nonlinear adaptive control of a continuous stirred tank reactor. The control strategy is based on an application of the controller consisting of a linear and nonlinear part. The static nonlinear part is derived in the way of a inversion and consecutive polynomial approximation of a measured or simulated input-output data. The design of the dynamic linear part is based on approximation of nonlinear elements in the control loop by a continuous-time external linear model with parameters estimated using a corresponding delta model. In the control design procedure, the polynomial approach with the pole assignment method is used. The nonlinear adaptive control is tested by simulations on the nonlinear model of the CSTR with a consecutive exothermic reaction.

1. INTRODUCTION

Continuous stirred tank reactors (CSTRs) are units frequently used in chemical and biochemical industry. From the system theory point of view, CSTRs belong to a class of nonlinear systems. Their mathematical models are described by sets of nonlinear differential equations. Their models are derived and described in e.g. Ogunnaike and Ray (1994), Schmidt (2005) and Corriou (2004).

It is well known that the control of chemical reactors often represents very complex problem. The control problems are due to the process nonlinearity and high sensitivity of the state and output variables to input changes. In addition, the dynamic characteristics may exhibit a varying sign of the gain in various operating points as well as non-minimum phase behaviour. Evidently, the process with such properties is hardly controllable by conventional control methods, and, its effective control requires application some of advanced methods.

One possible method to cope with this problem exploits a linear adaptive controller with parameters computed and readjusted on the basis of recursively estimated parameters of an appropriate chosen continuous-time external linear model (CT ELM) of the process. Some results obtained by this method can be found in e.g. Dostál et al. (2007) and Dostál et al. (2009).

An effective approach to the control of CSTRs and similar processes utilizes various methods of the nonlinear control (NC). Several modifications of the NC theory are described in e.g. Astolfi et al. (2008), Vincent and Grantham (1997), Ioannou and Fidan (2006) or Zhang et al. (2000). Especially, a large class of the NC methods exploits linearization of nonlinear plants, e.g. Huba and Ondera (2009), an application of PID controllers, e.g. Tan et al. (2002), Bányász and Keviczky (2002) or factorization of nonlinear models of the plants on linear and nonlinear parts, e.g. Nakamura et al. (2002), Vallery et al. (2009) and Chyi-Tsong Chen1 et al. (2006).

In this paper, the CSTR control strategy is based on an application of the controller consisting of a static nonlinear part (SNP) and dynamic linear part (DLP). The static nonlinear part is obtained from simulated or measured steady-state characteristic of the CSTR, its inversion, polynomial approximation, and, subsequently, its differentiation. On behalf of development of the linear part, the SNP including the nonlinear model of the CSTR are approximated by a CT external linear model. For the CT ELM parameter estimation, an external delta model with the same structure as the CT model is used (see, e.g. Mukhopadhyay et al. (1992), Goodwin et al. (2001) and Stericker and Sinha (1993)). Then, the resulting CT controller is derived using the polynomial approach and pole assignment method, e.g. Kučera (1993). The simulations are performed on a nonlinear model of the CSTR with a consecutive exothermic reaction.

2. MODEL OF THE CSTR

Consider a CSTR with the first order consecutive exothermic reaction according to the scheme $A \xrightarrow{k_1} B \xrightarrow{k_2} C$ and with a perfectly mixed cooling jacket. Using the usual simplifications, the model of the CSTR is described by four nonlinear differential equations

$$\frac{dc_A}{dt} = -\left(\frac{q_r}{V_r} + k_1\right) c_A + \frac{q_r}{V_r} c_{Af} \tag{1}$$

$$\frac{dc_B}{dt} = -\left(\frac{q_r}{V_r} + k_2\right) c_B + k_1 c_A + \frac{q_r}{V_r} c_{Bf} \tag{2}$$

$$\frac{dT_r}{dt} = \frac{h_r}{(\rho c_p)_r} + \frac{q_r}{V_r}(T_{rf} - T_r) + \frac{A_h U}{V_r (\rho c_p)_r}(T_c - T_r) \tag{3}$$

$$\frac{dT_c}{dt}=\frac{q_c}{V_c}(T_{cf}-T_c)+\frac{A_hU}{V_c(\rho c_p)_c}(T_r-T_c) \qquad (4)$$

with initial conditions $c_A(0)=c_A^s$, $c_B(0)=c_B^s$, $T_r(0)=T_r^s$ and $T_c(0)=T_c^s$. Here, t is the time, c are concentrations, T are temperatures, V are volumes, ρ are densities, c_p are specific heat capacities, q are volumetric flow rates, A_h is the heat exchange surface area and U is the heat transfer coefficient. The subscripts are denoted $(\cdot)_r$ for the reactant mixture, $(\cdot)_c$ for the coolant, $(\cdot)_f$ for feed (inlet) values and the superscript $(\cdot)^s$ for steady-state values. The reaction rates and the reaction heat are expressed as

$$k_j=k_{0j}\exp\left(\frac{-E_j}{RT_r}\right),\ j=1,2 \qquad (5)$$

$$h_r=h_1k_1c_A+h_2k_2c_B \qquad (6)$$

where k_0 are pre-exponential factors, E are activation energies and h are reaction entalpies. The values of all parameters, feed values and steady-state values are given in Table 1.

Table 1. Parameters, inlet values and initial conditions.

V_r = 1.2 m^3 V_c = 0.64 m^3 ρ_r = 985 kg m^{-3} ρ_c = 998 kg m^{-3}	c_{pr} = 4.05 kJ kg^{-1}K^{-1} c_{pc} = 4.18 kJ kg^{-1}K^{-1} A_h = 5.5 m^2 U = 43.5 kJ m^{-2}min^{-1}K^{-1}
k_{10} = 5.616 · 10^{16} min^{-1} k_{20} = 1.128 · 10^{18} min^{-1} h_1 = 4.8 · 10^4 kJ kmol^{-1}	E_1/R = 13477 K E_2/R = 15290 K h_2 = 2.2 · 10^4 kJ kmol^{-1}
c_{Af}^s = 2.85 kmol m^{-3} T_{rf}^s = 323 K q_r^s = 0.08 m^{3}min^{-1}	c_{Bf}^s = 0 kmol m^{-3} T_{cf}^s = 293 K q_c^s = 0.08 m^{3}min^{-1}
c_A^s = 1.5796 kmol m^{-3} T_r^s = 324.80 K	c_B^s = 1.1975 kmol m^{-3} T_c^s = 306.28 K

In term of the practice, only the coolant flow rate can be taken into account as the control input. As the controlled output, the reactant temperature is considered. For the control purposes, the control input and the controlled output are defined as deviations from steady values

$$u(t)=q_c(t)-q_c^s\,,\quad y(t)=T_r(t)-T_r^s \qquad (7)$$

The dependence of the reactant temperature on the coolant flow rate in the steady-state is in Fig.1.

In subsequent control simulations, the operating interval for q_c has been determined as

$$q_{c\min}\le q_c(t)\le q_{c\max} \qquad (8)$$

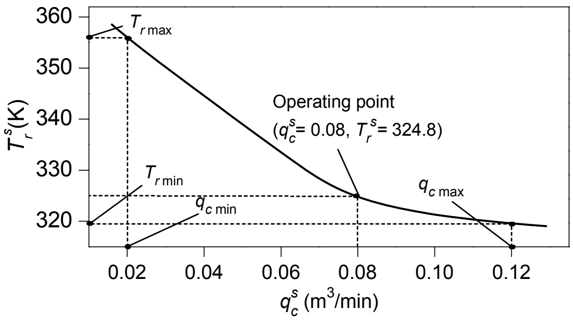


Fig. 1. Dependence of the reactant temperature on the coolant flow rate in the steady-state.

3. CONTROLLER DESIGN

As previously introduced, the controller consist of a static nonlinear part and a dynamic linear part as shown in Fig. 2.

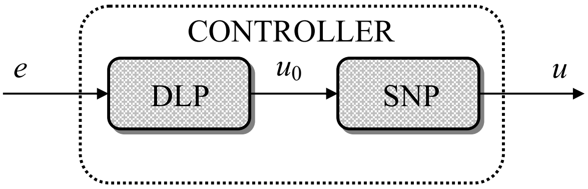


Fig. 2. The controller scheme.

The DLP creates a linear dynamic relation between the tracking error $e(t)$ and $u_0(t)=\Delta T_{rw}(t)$ which represents a difference of the reactant temperature adequate to its desired value. Evidently, for a well proposed SNP, the limit relation $\lim_{t\to\infty}u_0(t)=w$ holds.

Then, the SNP generates a static nonlinear relation betveen u_0 and a corresponding increment (decrement) of the coolant flow rate.

3.1 Nonlinear part of the controller

The SNP derivation appears from a simulated or measured steady-state charasteristics. From the purposes of a later polynomial approximation, the coordinates on the graph axis are defined as

$$\theta=\frac{q_c^s-q_{cL}}{q_{cL}}\,,\quad \xi=T_r^s-T_{rL}\,. \qquad (9)$$

where q_{cL} is the lower bound in the interval

$$q_{cL}\le q_c^s\le q_{cU} \qquad (10)$$

and, T_{rL} is the temperature corresponding to q_{cU}.

It can be recommended to select the interval (10) slightly longer than (8). In this paper, lower and upper values in (8) and (10) were chosen $q_{cL}=0.016, q_{c\min}=0.02, q_{c\max}=0.12$ and $q_{cU}=0.13$.

In term of the practice, it can be supposed that the measured data will be affected by measurement errors. The simulated steady-state characteristic that corresponds to reality is shown in Fig. 3.

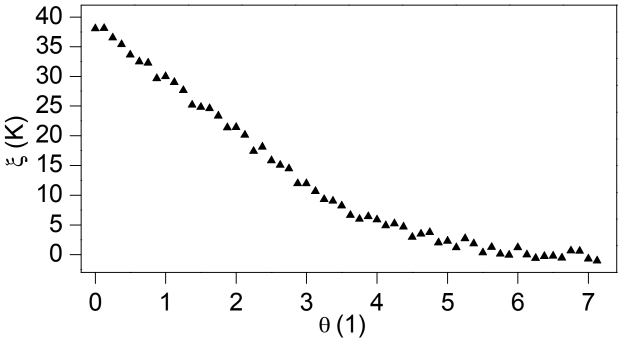


Fig. 3. Simulated characteristics $\xi = f(\theta)$.

Making the replacement of coordinates, the inverse of this characteristic can be approximated by a polynomial in the general form

$$\theta = a_0 + a_1\xi + ... + a_{n-1}\xi^{n-1} + a_n\xi^n \,. \tag{11}$$

The inverse characteristic accordant with Fig. 3 together with the fourth order approximate polynomial is in Fig. 4.

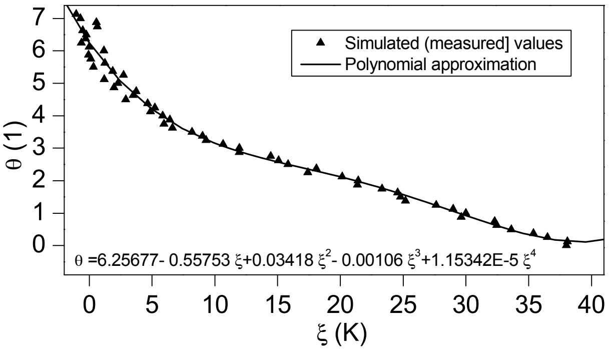


Fig. 4. Simulated and approximated inverse relation $\theta = \tilde{f}(\xi)$.

Now, a difference of the coolant flow rate $u(t) = \Delta q_c(t)$ in the output of the SNP can be computed for each T_r as

$$u(t) = \Delta q_c(t) = q_{cL} \left.\frac{d\theta}{d\xi}\right|_{\xi(T_r)} u_0(t) \tag{12}$$

The derivative of the approximate polynomial is in Fig. 5.

3.2 CT external linear model of nonlinear elements

A structure of the CT ELM of the SNP in conjuction with the CSTR nonlinear model was chosen on the basis of step responses simulated in a neighbourhood of the operating point. The step responses for some step changes of u_0 are shown in Fig. 6. For all responses, the gain of the SNP+CSTR system has been computed as $g_s = \lim\limits_{t\to\infty} \dfrac{y(t)}{u_0}$.

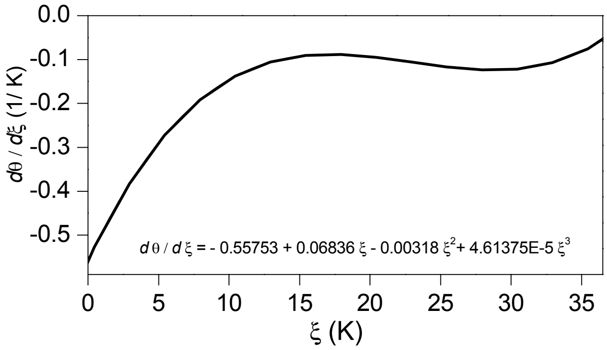


Fig. 5. Derivative of θ with respect to ξ.

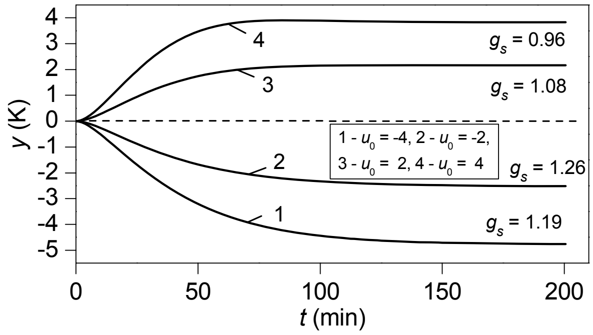


Fig. 6. Step responses of the SNP+CSTR.

Taking into account profiles of curves in Fig. 6 with zero derivatives for t = 0, the second order CT ELM has been chosen in the form of the second order linear differential equation

$$\ddot{y}(t) + a_1\dot{y}(t) + a_0 y(t) = b_0 u_0(t) \tag{13}$$

or, in the transfer function representation as

$$G(s) = \frac{Y(s)}{U_0(s)} = \frac{b_0}{s^2 + a_1 s + a_0} \,. \tag{14}$$

3.3 Delta external linear model

Establishing the δ operator

$$\delta = \frac{q-1}{T_0} \tag{15}$$

where q is the forward shift operator and T_0 is the sampling period, the delta ELM corresponding to (13) takes the form

$$\delta^2 y(t') + a_1'\delta y(t') + a_0' y(t') = b_0' u_0(t') \tag{16}$$

where t' is the discrete time.

When the sampling period is shortened, the delta operator approaches the derivative operator, and, the estimated parameters a', b' of (16) reach the parameters a, b of the CT model (13) as proved in e.g. Stericker and Sinha (1993).

Substituting $t' = k - 2$, equation (16) may be rewriten to the form

$$\delta^2 y(k-2) + a_1'\delta y(k-2) + a_0' y(k-2) = b_0' u_0(k-2) \,. \tag{17}$$

3.4 Delta model parameter estimation

Establishing the regression vector

$$\boldsymbol{\Phi}_\delta^T(k-1) = \left(\delta y(k-2)\; y(k-2)\; u_0(k-2)\right) \tag{18}$$

where

$$\delta y(k-2) = \frac{y(k-1) - y(k-2)}{T_0} \tag{19}$$

then, the vector of delta model parameters

$$\boldsymbol{\Theta}_\delta^T(k) = \left(a_1'\; a_0'\; b_0'\right) \tag{20}$$

is recursively estimated from the ARX model

$$\delta^2 y(k-2) = \boldsymbol{\Theta}_\delta^T(k)\boldsymbol{\Phi}_\delta(k-1) + \varepsilon(k) \tag{21}$$

where

$$\delta^2 y(k-2) = \frac{y(k) - 2y(k-1) + y(k-2)}{T_0^2}. \tag{22}$$

The recursive estimation of delta model parameters was performed with the sampling interval T_0 = 0.2 min. Here, the recursive identification method with exponential and directional forgetting according to Rao and Unbehauen (2005) and Bobál et al. (2005) was used.

3.5 Linear part of the controller

The DLP is inserted into the control loop according to Fig. 7.

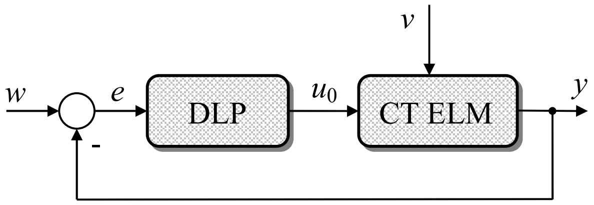


Fig. 7. Simplified scheme of the control loop.

In the scheme, w is the reference signal, v is the disturbance, y is the controlled output and u_0 is the input to the CT ELM. The transfer function G of the CT ELM is given by (14). Both the reference w and the disturbance v are considered to be step functions with transforms

$$W(s) = \frac{w_0}{s}, \quad V(s) = \frac{v_0}{s} \tag{23}$$

The transfer function of the DLP is in the form

$$Q(s) = \frac{U_0(s)}{E(s)} = \frac{q(s)}{p(s)} \tag{24}$$

where q and p are polynomials in s, and, $\deg q \le \deg p$.

The controller design described in this section stems from the polynomial approach. General conditions required to govern the control system properties are formulated as strong stability (in addition to the control system stability, also the stability of controllers is required), internal properness, asymptotic tracking of a step reference and step disturbance attenuation.

It is well known from the algebraic control theory that a controller which satisfies above requirements is in the polynomial ring given by a solution of the polynomial (Diophantine) equation

$$a(s)\,p(s) + b(s)\,q(s) = d(s) \tag{25}$$

with a stable polynomial $d(s)$ on the right side.

For step input signals w and v, the polynomial p is in the form

$$p(s) = s\,\tilde{p}(s)\,. \tag{26}$$

The degrees of unknown polynomials in (25) and (26) are

$$\deg q = \deg a\,, \quad \deg \tilde{p} = \deg a - 1, \quad \deg d = 2\deg a\,.$$

Then, for the ELM (14), the controller transfer function takes the form

$$Q(s) = \frac{q(s)}{s\,\tilde{p}(s)} = \frac{q_2 s^2 + q_1 s + q_0}{s\,(s + p_0)} \tag{27}$$

In this paper, the polynomial d with roots determining the closed-loop poles is chosen as

$$d(s) = n(s)\,(s+\alpha)^2 \tag{28}$$

where n is a stable polynomial obtained by spectral factorization

$$a^*(s)\,a(s) = n^*(s)\,n(s) \tag{29}$$

and α is the selectable parameter that can usually be chosen by way of simulation experiments.

Note that a choice of d in the form (28) provides the control of a good quality for aperiodic controlled processes.

The polynomial n has the form

$$n(s) = s^2 + n_1 s + n_0 \tag{30}$$

with coefficients

$$n_0 = \sqrt{a_0^2}\,, \quad n_1 = \sqrt{a_1^2 + 2n_0 - 2a_0}\,. \tag{31}$$

The controller parameters can be obtained from solution of the matrix equation

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ a_1 & b_0 & 0 & 0 \\ a_0 & 0 & b_0 & 0 \\ 0 & 0 & 0 & b_0 \end{pmatrix} \cdot \begin{pmatrix} p_0 \\ q_2 \\ q_1 \\ q_0 \end{pmatrix} = \begin{pmatrix} d_3 - a_1 \\ d_2 - a_0 \\ d_1 \\ d_0 \end{pmatrix} \tag{32}$$

where

$$\begin{aligned} d_3 &= n_1 + 2\alpha, d_2 = 2\alpha n_1 + n_0 + \alpha^2 \\ d_1 &= 2\alpha n_0 + \alpha^2 n_1, d_0 = \alpha^2 n_0 \end{aligned}\;. \tag{33}$$

Evidently, the controller parameters can be adjusted by the selectable parameter α. The complete adaptive control system is shown in Fig. 8.

4. CONTROL SIMULATIONS

The control simulations were performed in a neighbourhood of the operating point (q_c^s = 0.08 m^{3}min^{-1}, T_r^s = 324.8 K).

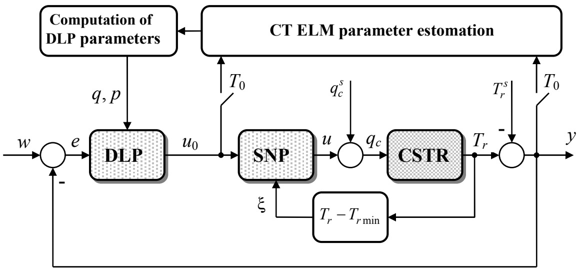


Fig. 8. Scheme of the nonlinear adaptive control.

For the start (the adaptation phase), a P controller with a small gain was used in all simulations.

The effect of the pole α on the control responses is transparent from Figs. 9, 10 and 11. Here, on the basis of precomputed simulations, two values of α were selected. The control results show sensitivity of the controlled output and the input signals to α. Obviously, careless selection of this parameter can lead to controlled output responses of a poor quality or even to unstability. Further, a increasing α leads to higher values and changes of the input signals.

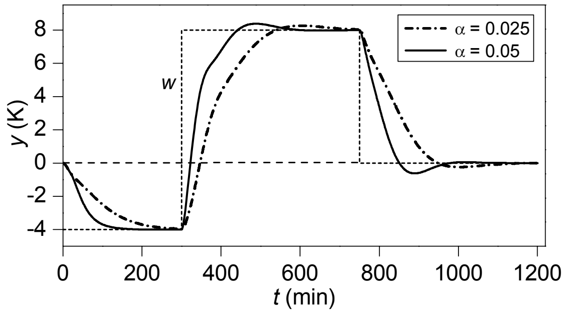


Fig. 9. Nonlinear adaptive control: Controlled output responses.

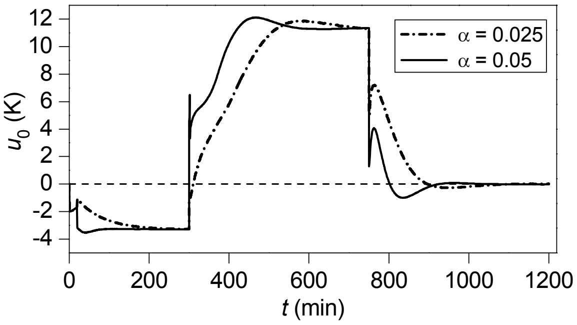


Fig. 10. Nonlinear adaptive control: Input signal to the SNP.

Evolution of the DLP parameters during control is shown in Fig. 12.

A presence of the integrating part in the DLP enables rejection of various step disturbances entering into the process. Here, step disturbances $\Delta c_{Af} = \pm 0.1\,\mathrm{kmol\,m^{-3}}$ at times $t_{v1} = 250\,\mathrm{min}$ and $t_{v2} = 550\,\mathrm{min}$ were injected into the CSTR. The DLP parameters were estimated only in the first (tracking) interval $t < 200$ min. The experiences of authors of this paper proved that an utilization of recursive identification using the delta model in the phase of a constant reference and in a presence of step disturbances decreases the control quality. From this reason, during interval $t \geq 200$ min, fixed DLP parameters were used. The controlled output responses are shown in Fig. 13.

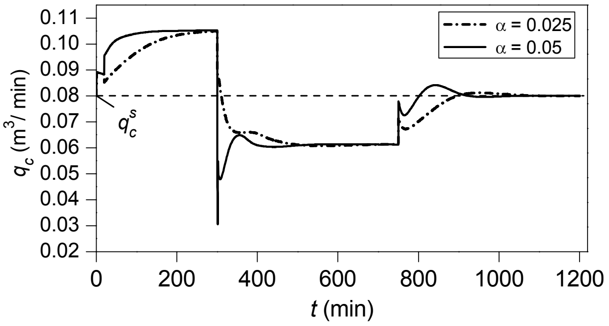


Fig. 11. Nonlinear adaptive control: Input to the CSTR.

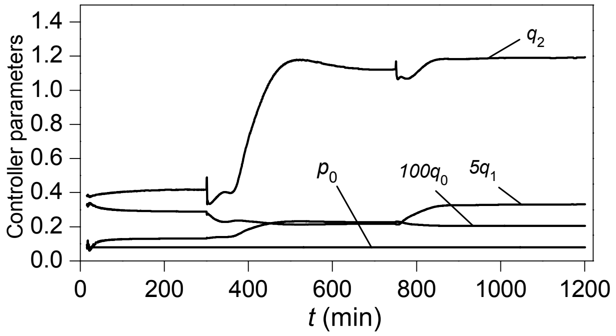


Fig. 12. Evolution of the DLP parameters during control.

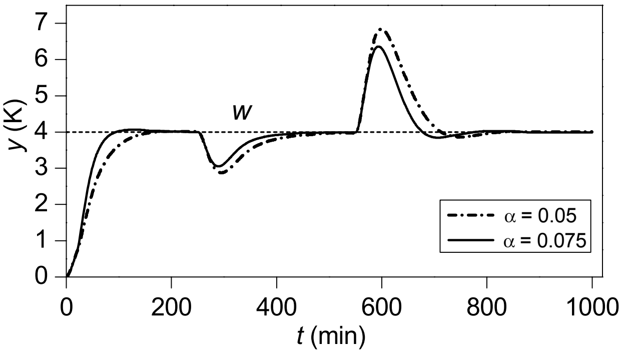


Fig. 13. Nonlinear adaptive control: Step disturbances rejection.

An influence of the SNP is evident from the control response shown in Fig. 14. Here, the standard adaptive control without the nonlinear part of the controller was used. The simulation has been performed under the same conditions as by above presented cases. A confrontation with responses in Fig. 9 shows that an application of the nonlinear control is suitable especially for greater changes of the reference signal.

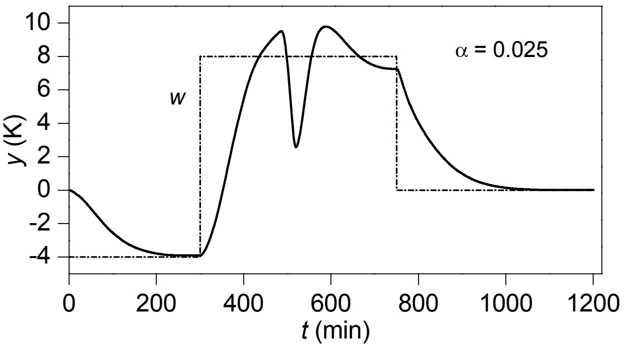


Fig. 14. Adaptive control without the SNP.

5. CONCLUSIONS

In this paper, one approach to the nonlinear continuous-time adaptive control of the reactant temperature in a continuous stirred tank reactor was proposed. The control strategy is based on a factorization of a controller into the linear and the nonlinear part. A design of the controller nonlinear part employs simulated or measured steady-state characteristics of the process and their additional modifications. Then, the system consisting of the controller nonlinear part and a nonlinear model of the CSTR is approximeted by a continuous time external linear model with parameters obtained through recursive parameter estimation of a corresponding delta model. The resulting continuous-time controller linear part is derived using the polynomial approach and given by a solution of a polynomial equation. Tuning of its parameters is possible via closed-loop pole assignment. The presented method has been tested by computer simulation on the nonlinear model of the CSTR with a consecutive exothermic reaction. Simulation results demonstrated an applicability of the presented control strategy and its usefulness especially for greater changes of input signals in strongly nonlinear regions. It can be expected that the described control strategy is also suitable for other similar technological processes such as tubular chemical reactors.

ACKNOWLEDGMENT

The authors wish to thank to the Ministry of Education of the Czech Republic (MSM7088352101) for financial support. This article was created with support of Operational Programme Research and Development for Innovations co-funded by the European Regional Development Fund (ERDF) and national budget of Czech Republic within the framework of the Centre of Polymer Systems project (reg.number: CZ.1.05/2.1.00/03.0111).

REFERENCES

Astolfi, A., Karagiannis, D., and Ortega, R. (2008). *Nonlinear and adaptive control with applications*, Springer-Verlag, London.

Bányász, Cs., and Keviczky, L. (2002). A Simple PID regulator applicable for a class of factorable nonlinear plants. *American Control Conference*, Anchorage, Alaska, 2354-2359.

Bobál, V., Böhm, J., Fessl, J., and Macháček, J. (2005). *Digital Self-tuning Controllers*, Springer Verlag, Berlin.

Chen1, Chyi-Tsong, Chuang1, Yao-Chen, and Hwang, Chyi (2006). A Simple Nonlinear Control Strategy for Chemical Processes. *6th Asian Control Conference*, Bali, Indonesia, 64-70.

Corriou, J.-P. (2004). *Process control. Theory and applications*, Springer – Verlag, London.

Dostál, P., Gazdoš, F., Bobál, V., and Vojtěšek, J. (2007). Adaptive control of a continuous stirred tank reactor by two feedback controllers. *9th IFAC Workshop Adaptation and Learning in Control and Signal Processing ALCOSP'2007*, Saint Petersburg, Russia, P5-1 – P5-6.

Dostál, P., Vojtěšek, J., and Bobál, V. (2009). Simulation of adaptive control of a continuous stirred tank reactor. *23rd European Conference on Modelling and Simulation, ECMS 2009*, Madrid, 625-630.

Goodwin, G.C., Graebe, S.F., and Salgado, M.E. (2001). *Control System Design*, Prentice Hall, New Jersey.

Huba, M., and Ondera, M. (2009). Simulation of nonlinear control systems represented as generalized transfer functions. *European Control Conference 2009*, Budapest, Hungary, 1444-1449.

Ioannou, P., and Fidan, B. (2006). *Adaptive control tutorial*, SIAM, Philadelphia.

Kučera, V. (1993). Diophantine equations in control – A survey. *Automatica*, 29, 1361-1375.

Mukhopadhyay, S., Patra, A.G., and Rao, G.P. (1992). New class of discrete-time models for continuos-time systems. *International Journal of Control*, 55**,** 1161-1187.

Nakamura, M., Sugi, T., and Goto, S. (2002). Nonlinear separation model and control for a complex process realized by conventional PID controller hardware. *4th Asian Control Conference*, Singapore, 274-279.

Ogunnaike, B.A., and Ray, W.H. (1994). *Process dynamics, modeling, and control*, Oxford University Press, New York.

Rao, G.P., and Unbehauen, H. (2006). Identification of continuous-time systems. *IEE Proc.-Control Theory Appl.*, 153, 185-220.

Schmidt, L.D. (2005). *The engineering of chemical reactions*, Oxford University Press, New York.

Stericker, D.L., and Sinha, N.K. (1993). Identification of continuous-time systems from samples of input-output data using the δ-operator. *Control-Theory and Advanced Technology*, 9, 113-125.

Tan, K.K., Ferdous, R., and Huang, S. (2002). Closed-loop automatic tuning of PID controller for nonlinear systems. *4th Asian Control Conference*, Singapore, 1874-1879.

Vallery, H., Neumaier, M., and Buss, M. (2009). Anti-causal identification of Hammerstein models. *European Control Conference 2009*, Budapest, Hungary, 1071-1076.

Vincent, T.L., and Grantham, W.J. (1997). *Nonlinear and optimal control systems*, John Wiley & Sons, New York.

Zhang, T., Ge, S.S., and Hang, C.C. (2000). Stable adaptive control for a class of nonlinear systems using a modified Lyapunov function. *IEEE Transactions on Automatic Control,* 45(1), 129-132.

H∞ control of time-delay systems with time-varying delays

A. Filasová, D. Krokavec

Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Letná 9, 042 00 Košice, Slovakia
(tel: ++ 421 55 602 4389; e-mail: anna.filasova@tuke.sk,
tel: ++ 421 55 602 2564; e-mail: dusan.krokavec@tuke.sk)

Abstract: The linear matrix inequality (LMI) based memory-less controller design approach for continuous time systems with time-varying delays is presented in the paper. If the time-delay variation is from the specified range the design conditions are formulated as feasibility problem and expressed over a set of LMIs with the matrix rank constraints implying from integral quadratic constraints. The proposed method is demonstrated using a system model example.

Keywords: Linear matrix inequality, time-delay systems, time-varying delays, linear systems.

1. INTRODUCTION

Continuous-time control systems are used in many industrial applications, where time delays can take a deleterious effect on both the stability and the dynamic performance in the open and closed-loop systems. Thus, the problems of asymptotic stability and stabilization for time-delay systems have received considerable attention and intensive activity are done to develop a sophisticated control for such systems.

Linear matrix inequality (LMI) approaches based on convex optimization algorithms have been extensively applied to solve the above mentioned problem, since it can be solved numerically efficiently by using interior-point algorithm which has recently been developed for solving optimization problem. Using the LMI approach, two categories of stability criteria for guaranteeing stability of the delayed system were developed. Delay independent criteria provide conditions which guarantee stability for any length of the time delay, whereas delay dependent criteria exploit a priori knowledge of upper-bounds on the amount of time-delay or its variation. Of course, delay dependent criteria are generally less conservative than delay-independent ones since more information about the time-delay is assumed to be known.

The use of Lyapunov method for the stability analysis of the time delay systems has been ever growing subject of interest starting with the pioneering works of Krasovskii (Krasovskii (1956), Krasovskii (1963)). Progress review in this research field is presented e.g. in Niculescu at al. (1998), Wu at al. (2004), Kao and Rantzer (2005), and the references therein, some special forms of Lyapunov-Krasovskii functions can be also found in Wu at al. (2010).

Considering the influence of time-varying delay as perturbation in the system, where delay parameter is an unknown time-varying function with given upper bounds on the magnitude and the variation, the paper address the problems of asymptotic stabilization for such time-delay systems if the time-delay variation is from the specified range. Translating into LMI framework the closed-loop system stability is characterized in the terms of convex LMIs, where the convex parameterizations are based on extended Lyapunov function with integral quadratic constraints in the bounded real lemma form.

2. PROBLEM DESCRIPTION

Through this paper the task is concerned with the computation of a state feedback $\boldsymbol{u}(t)$, which control the time-delay linear dynamic system given by the set of equations

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{A}_d\boldsymbol{q}(t-\tau(t)) + \boldsymbol{B}\boldsymbol{u}(t) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{D}\boldsymbol{u}(t) \tag{2}$$

with initial condition

$$\boldsymbol{q}(\vartheta) = \varphi(\vartheta),\ \forall\vartheta \in \langle -h, 0\rangle \tag{3}$$

where $\tau(t)$ is an unknown time-varying parameter satisfying conditions

$$0 \le \tau(t) \le h, \qquad |\,\dot{\tau}(t)\,| \le d, \quad \forall t \ge 0 \tag{4}$$

where $\boldsymbol{q}(t) \in I\!R^n$ stands up for the system state, $\boldsymbol{u}(t) \in I\!R^r$ denotes the control input, $\boldsymbol{y}(t) \in I\!R^m$ is the system measurable output, and nominal system matrices $\boldsymbol{A} \in I\!R^{n\times n}$, $\boldsymbol{A}_d \in I\!R^{n\times n}$, $\boldsymbol{B} \in I\!R^{n\times r}$, $\boldsymbol{C} \in I\!R^{m\times n}$ and $\boldsymbol{D} \in I\!R^{m\times r}$ are real matrices.

Problem of the interest is to design asymptotically stable closed-loop system with the linear memoryless state feedback controller of the form

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{q}(t) \tag{5}$$

for $t \ge 0$, where matrix $\boldsymbol{K} \in I\!R^{r\times n}$ is a gain matrix.

3. BASIC PRELIMINARIES

Proposition 1. (Bounded real lemma) System (1), (2), where $\boldsymbol{A}_d = \boldsymbol{0}$, is stable with quadratic performance $\|\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}\|_\infty \le \gamma$ if there exist a symmetric positive definite matrix $\boldsymbol{P} > 0$ and a positive scalar $\gamma > 0$ such that

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A} & \boldsymbol{P}\boldsymbol{B} & \boldsymbol{C}^T \\ * & -\gamma^2\boldsymbol{I}_r & \boldsymbol{D}^T \\ * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{6}$$

where $\boldsymbol{I}_r \in \mathbb{R}^{r\times r}$, $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$ are identity matrices, respectively,

Hereafter, $*$ denotes the symmetric item in a symmetric matrix.

Proof. Generally, there exists an enough small $\gamma > 0$ such that Lyapunov function can be defined as follows

$$\begin{aligned} v(\boldsymbol{q}(t)) &= \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+ \\ &+\int_0^t (\boldsymbol{y}^T(r)\boldsymbol{y}(r)-\gamma^2\boldsymbol{u}^T(r)\boldsymbol{u}(r))\mathrm{d}r > 0 \end{aligned} \tag{7}$$

where $\boldsymbol{P} = \boldsymbol{P}^T > 0$, $\boldsymbol{P} \in \mathbb{R}^{n\times n}$, $\gamma > 0 \in \mathbb{R}$, and evaluating the derivative of $v(\boldsymbol{q}(t))$ with respect to t along a system trajectory then it yields

$$\begin{aligned} \dot{v}(\boldsymbol{q}(t)) &= \dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+\boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t)+ \\ &+\boldsymbol{y}^T(t)\boldsymbol{y}(t)-\gamma^2\boldsymbol{u}^T(t)\boldsymbol{u}(t) < 0 \end{aligned} \tag{8}$$

Thus, substituting (1), (2) into (8) it can be written

$$\begin{aligned} \dot{v}(\boldsymbol{q}(t)) &= (\boldsymbol{A}\boldsymbol{q}(t)+\boldsymbol{B}\boldsymbol{u}(t))^T\boldsymbol{P}\boldsymbol{q}(t)+ \\ &+\boldsymbol{q}^T(t)\boldsymbol{P}(\boldsymbol{A}\boldsymbol{q}(t)+\boldsymbol{B}\boldsymbol{u}(t))-\gamma\boldsymbol{u}^T(t)\boldsymbol{u}(t)+ \\ &+(\boldsymbol{C}\boldsymbol{q}(t)+\boldsymbol{D}\boldsymbol{u}(t))^T(\boldsymbol{C}\boldsymbol{q}(t)+\boldsymbol{D}\boldsymbol{u}(t)) < 0 \end{aligned} \tag{9}$$

and with the next notation

$$\boldsymbol{q}_c^T(t) = \begin{bmatrix} \boldsymbol{q}^T(t) & \boldsymbol{u}^T(t) \end{bmatrix} \tag{10}$$

it is obtained

$$\dot{v}(\boldsymbol{q}(t)) = \boldsymbol{q}_c^T(t)\boldsymbol{P}_c\boldsymbol{q}_c(t) < 0 \tag{11}$$

where

$$\boldsymbol{P}_c = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A} & \boldsymbol{P}\boldsymbol{B} \\ * & -\gamma^2\boldsymbol{I}_r \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{C}^T\boldsymbol{D} \\ * & \boldsymbol{D}^T\boldsymbol{D} \end{bmatrix} < 0 \tag{12}$$

Since

$$\begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{C}^T\boldsymbol{D} \\ * & \boldsymbol{D}^T\boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}^T \\ \boldsymbol{D}^T \end{bmatrix} [\,\boldsymbol{C}\ \boldsymbol{D}\,] \ge 0 \tag{13}$$

Schur complement property implies

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}^T \\ * & \boldsymbol{0} & \boldsymbol{D}^T \\ * & * & -\boldsymbol{I}_m \end{bmatrix} \ge 0 \tag{14}$$

and using (14) the LMI condition (12) can be written compactly as (6). This concludes the proof. ■

Proposition 2. (Symmetric upper-bound inequality) Let $f(\boldsymbol{x}(\eta))$, $\boldsymbol{x}(\eta) \in \mathbb{R}^n$, $\boldsymbol{X} > 0$, $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ is a real positive definite and integrable vector function of the form

$$f(\boldsymbol{x}(\eta)) = \boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta) \tag{15}$$

such, that there exists a well defined integration as following

$$\int_{t-h}^{t} f(\boldsymbol{x}(\eta))\mathrm{d}\eta > 0 \tag{16}$$

with $h > 0$, $h \in \mathbb{R}$, then

$$\int_{t-h}^{t}\boldsymbol{x}^T(\eta)\mathrm{d}\eta\boldsymbol{X}\int_{t-h}^{t}\boldsymbol{x}(\eta)\mathrm{d}\eta \le h\int_{t-h}^{t}\boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta)\mathrm{d}\eta \tag{17}$$

Proof. Since for (15) it can be written

$$\boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta) - \boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta) = 0 \tag{18}$$

and according to Schur complement property it is true that

$$\begin{bmatrix} \boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta) & \boldsymbol{x}^T(\eta) \\ \boldsymbol{x}(\eta) & \boldsymbol{X}^{-1} \end{bmatrix} = 0 \tag{19}$$

then the integration of (19) with respect to η gives

$$\begin{bmatrix} \int_{t-h}^{t}\boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta)\mathrm{d}\eta & \int_{t-h}^{t}\boldsymbol{x}^T(\eta)\mathrm{d}\eta \\ * & \int_{t-h}^{t}\boldsymbol{X}^{-1}\mathrm{d}\eta \end{bmatrix} \ge 0 \tag{20}$$

$$\begin{bmatrix} \int_{t-h}^{t}\boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta)\mathrm{d}\eta & \int_{t-h}^{t}\boldsymbol{x}^T(\eta)\mathrm{d}\eta \\ * & h\boldsymbol{X}^{-1} \end{bmatrix} \ge 0 \tag{21}$$

respectively. Thus,

$$h^{-1}\int_{t-h}^{t}\boldsymbol{x}^T(\eta)\mathrm{d}\eta\boldsymbol{X}\int_{t-h}^{t}\boldsymbol{x}(\eta)\mathrm{d}\eta \le \int_{t-h}^{t}\boldsymbol{x}^T(\eta)\boldsymbol{X}\boldsymbol{x}(\eta)\mathrm{d}\eta \tag{22}$$

and it is evident that with $h > 0$ (22) implies (17). This concludes the proof. ■

4. DESCRIPTOR SYSTEM PROPERTIES

Adding and subtracting vector element $\boldsymbol{A}_d\boldsymbol{q}(t)$ to (1) results in

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{B}\boldsymbol{u}(t)+(\boldsymbol{A}+\boldsymbol{A}_d)\boldsymbol{q}(t)-\boldsymbol{A}_d(\boldsymbol{q}(t)-\boldsymbol{q}(t-\tau(t))) \tag{23}$$

It is well-known fact that the descriptor model (23) is not equivalent to system (1), since this transformation introduces additional dynamics. However, stability of system (23) does imply stability of system (1), i.e. the delay-derivative-independent stability criterion it is necessary to be stated.

Considering $\boldsymbol{u}(t)=\mathbf{0}$ then the autonomous system to (23) can be written as

$$\dot{\boldsymbol{q}}(t)=(\boldsymbol{A}+\boldsymbol{A}_d)\boldsymbol{q}(t)+\boldsymbol{A}_d\boldsymbol{u}^{\circ}(t) \tag{24}$$

where

$$\boldsymbol{u}^{\circ}(t)=-(\boldsymbol{q}(t)-\boldsymbol{q}(t-\tau(t))) \tag{25}$$

$$\boldsymbol{u}^{\circ}(t)=-\boldsymbol{I}\int\limits_{t-\tau(\eta)}^{t}\dot{\boldsymbol{q}}(\eta)\mathrm{d}\eta=-\boldsymbol{I}\int\limits_{t-\tau(\eta)}^{t}\boldsymbol{y}^{\circ}(\eta)\mathrm{d}\eta \tag{26}$$

respectively, with

$$\boldsymbol{y}^{\circ}(t)=\dot{\boldsymbol{q}}(t)=(\boldsymbol{A}+\boldsymbol{A}_d)\boldsymbol{q}(t)+\boldsymbol{A}_d\boldsymbol{u}^{\circ}(t) \tag{27}$$

Therefore, (24), (25) can be interpreted as a dynamic system with uncertain internal integral closed-loop feedback.

Denoting

$$\dot{\boldsymbol{q}}(t)=\boldsymbol{A}^{\circ}\boldsymbol{q}(t)+\boldsymbol{B}^{\circ}\boldsymbol{u}^{\circ}(t) \tag{28}$$

$$\boldsymbol{y}^{\circ}(t)=\boldsymbol{C}^{\circ}\boldsymbol{q}(t)+\boldsymbol{D}^{\circ}\boldsymbol{u}^{\circ}(t) \tag{29}$$

where

$$\boldsymbol{A}^{\circ}=\boldsymbol{C}^{\circ}=\boldsymbol{A}+\boldsymbol{A}_d,\quad \boldsymbol{B}^{\circ}=\boldsymbol{D}^{\circ}=\boldsymbol{A}_d \tag{30}$$

then an equivalent Lyapunov function to the (7) can be introduced. Unlike a delay-free linear system there exist state boundaries in the descriptor system, so the weighting matrices of Lyapunov function have to be introduced in special forms.

Considering the quadratic integral form

$$\begin{aligned}J_1&=\int\limits_{0}^{\infty}\boldsymbol{u}^{\circ T}(t)\boldsymbol{X}\boldsymbol{u}^{\circ}(t)\mathrm{d}t=\\&=\int\limits_{0}^{\infty}\int\limits_{t-\tau(\eta)}^{t}\dot{\boldsymbol{q}}^T(\eta)\mathrm{d}\eta\,\boldsymbol{X}\int\limits_{t-\tau(\eta)}^{t}\dot{\boldsymbol{q}}(\eta)\mathrm{d}\eta\mathrm{d}t\end{aligned} \tag{31}$$

then using (17) it is obvious that

$$\begin{aligned}J_1&\le\int\limits_{0}^{\infty}\int\limits_{t-h}^{t}\dot{\boldsymbol{q}}^T(\eta)\mathrm{d}\eta\,\boldsymbol{X}\int\limits_{t-h}^{t}\dot{\boldsymbol{q}}(\eta)\mathrm{d}\eta\mathrm{d}t\le\\&\le\int\limits_{0}^{\infty}h\int\limits_{t-h}^{t}\dot{\boldsymbol{q}}^T(\eta)\boldsymbol{X}\dot{\boldsymbol{q}}(\eta)\mathrm{d}\eta\mathrm{d}t=h^2\int\limits_{0}^{\infty}\dot{\boldsymbol{q}}^T(\eta)\boldsymbol{X}\dot{\boldsymbol{q}}(\eta)\mathrm{d}\eta\end{aligned} \tag{32}$$

It is evident that the integral norm-weighting matrix in (32) is independent of d. Analogously, respecting

$$J_2=\int\limits_{0}^{\infty}\boldsymbol{q}^T(t-\tau(t))\boldsymbol{X}\boldsymbol{q}(t-\tau(t))\mathrm{d}t \tag{33}$$

then setting

$$t-\tau(t)=\eta,\quad (1-\dot{\tau}(t))\mathrm{d}t=\mathrm{d}\eta \tag{34}$$

(33) can be rewritten as follows

$$\begin{aligned}J_2&=\int\limits_{-\tau(0)}^{\infty}\frac{1}{1-\dot{\tau}(\eta(t))}\boldsymbol{q}^T(\eta)\boldsymbol{X}\boldsymbol{q}(\eta)\mathrm{d}\eta\le\\&\le\frac{1}{1-d}\int\limits_{0}^{\infty}\boldsymbol{q}^T(\eta)\boldsymbol{X}\boldsymbol{q}(\eta)\mathrm{d}\eta\end{aligned} \tag{35}$$

Conversely, the integral norm-weighting matrix in (35) is independent of h as long as h is strictly greater than 0. Using (35) property then

$$\begin{aligned}&\int\limits_{0}^{\infty}\left[\,\boldsymbol{q}^T(t-\tau(t))\ \boldsymbol{q}^T(t)\,\right]\begin{bmatrix}\boldsymbol{X}&\\&\boldsymbol{X}\end{bmatrix}\begin{bmatrix}\boldsymbol{q}(t-\tau(t))\\\boldsymbol{q}(t)\end{bmatrix}\mathrm{d}t\le\\&\le\int\limits_{0}^{\infty}\boldsymbol{q}^T(\eta)\boldsymbol{X}\boldsymbol{q}(\eta)\mathrm{d}\eta+\frac{1}{1-d}\int\limits_{0}^{\infty}\boldsymbol{q}^T(\eta)\boldsymbol{X}\boldsymbol{q}(\eta)\mathrm{d}\eta=\\&=\frac{2-d}{1-d}\int\limits_{0}^{\infty}\boldsymbol{q}^T(\eta)\boldsymbol{X}\boldsymbol{q}(\eta)\mathrm{d}\eta\end{aligned} \tag{36}$$

Considering $|\dot{\tau}(t)|\le d$, $1<d\le 2$ it is evident that (36) is negative.

Summarizing, such forms as (36) cannot be generally included into Lyapunov-Krasovskii functional if $1<d\le 2$ since may cause its negative definiteness, and only the standard form of Lyapunov function is proposed to use.

Theorem 1. Autonomous linear time-delay system (1) is stable for $|\dot{\tau}(t)|\le d$, $1<d\le 2$ if there exist symmetric positive definite matrices $\boldsymbol{P}>0$, $\boldsymbol{Q}>0$, $\boldsymbol{P},\boldsymbol{Q}\in I\!R^{n\times n}$, such that

$$\boldsymbol{P}=\boldsymbol{P}^T>0\qquad \boldsymbol{Q}=\boldsymbol{Q}^T>0 \tag{37}$$

$$\begin{bmatrix}\boldsymbol{\Pi}_{11}&h^2(\boldsymbol{A}+\boldsymbol{A}_d)^T\boldsymbol{Q}\boldsymbol{A}_d+\boldsymbol{P}\boldsymbol{A}_d\\ *&h^2\boldsymbol{A}_d^T\boldsymbol{Q}\boldsymbol{A}_d-\boldsymbol{Q}\end{bmatrix}<0 \tag{38}$$

$$\begin{aligned}\boldsymbol{\Pi}_{11}&=(\boldsymbol{A}+\boldsymbol{A}_d)^T\boldsymbol{P}+\boldsymbol{P}(\boldsymbol{A}+\boldsymbol{A}_d)+\\&+h^2(\boldsymbol{A}+\boldsymbol{A}_d)^T\boldsymbol{Q}(\boldsymbol{A}+\boldsymbol{A}_d)\end{aligned} \tag{39}$$

Proof. Lyapunov function candidate can be chosen as

$$\begin{aligned}0<v(\boldsymbol{q}(t))&=\boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t)+\\&+\int\limits_{0}^{t}(h^2\boldsymbol{y}^{\circ T}(r)\boldsymbol{Q}\boldsymbol{y}^{\circ}(r)-\boldsymbol{u}^{\circ T}(r)\boldsymbol{Q}\boldsymbol{u}^{\circ}(r))\mathrm{d}r\end{aligned} \tag{40}$$

where $\boldsymbol{P}=\boldsymbol{P}^T>0$, $\boldsymbol{Q}=\boldsymbol{Q}^T>0$. Evaluating derivative of $v(\boldsymbol{q}(t))$ with respect to t results in

$$\begin{aligned}\dot{v}(\boldsymbol{q}(t))&=-\boldsymbol{u}^{\circ T}(t)\boldsymbol{Q}\boldsymbol{u}^{\circ}(t)+\\&+(\boldsymbol{q}^T(t)\boldsymbol{A}^{\circ T}+\boldsymbol{u}^{\circ T}(t)\boldsymbol{B}^{\circ T})\boldsymbol{P}\boldsymbol{q}(t)+\\&+\boldsymbol{q}^T(t)\boldsymbol{P}(\boldsymbol{A}^{\circ}\boldsymbol{q}(t)+\boldsymbol{B}^{\circ}\boldsymbol{u}^{\circ}(t))+\\&+h^2(\boldsymbol{C}^{\circ}\boldsymbol{q}(t)+\boldsymbol{D}^{\circ}\boldsymbol{u}^{\circ}(t))^T\boldsymbol{Q}(\boldsymbol{C}^{\circ}\boldsymbol{q}(t)+\boldsymbol{D}^{\circ}\boldsymbol{u}^{\circ}(t))<0\end{aligned} \tag{41}$$

Thus, introducing the composite vector $\boldsymbol{q}^{\circ}(t)$ as follows

$$\boldsymbol{q}^{\circ T}(t)=\left[\,\boldsymbol{q}^T(t)\ \boldsymbol{u}^{\circ T}(t)\,\right] \tag{42}$$

it is possible to write the Lyapunov function derivative (42) as follows

$$\dot{v}(\boldsymbol{q}^{\circ}(t))=\boldsymbol{q}^{\circ T}(t)\boldsymbol{P}^{\circ}\boldsymbol{q}^{\circ}(t)<0 \tag{43}$$

where

$$\boldsymbol{P}^{\circ}=\begin{bmatrix}\boldsymbol{P}_{11}^{\circ}&h^2\boldsymbol{C}^{\circ T}\boldsymbol{Q}\boldsymbol{D}^{\circ}+\boldsymbol{P}\boldsymbol{B}^{\circ}\\ *&h^2\boldsymbol{D}^{\circ T}\boldsymbol{Q}\boldsymbol{D}^{\circ}-\boldsymbol{Q}\end{bmatrix}<0 \tag{44}$$

$$\boldsymbol{P}_{11}^{\circ}=\boldsymbol{A}^{\circ T}\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}^{\circ}+h^2\boldsymbol{C}^{\circ T}\boldsymbol{Q}\boldsymbol{C}^{\circ} \tag{45}$$

Subsequently, inserting (30) then (44), (45) implies (38), (39). This concludes the proof. ■

5. CONTROL LAW PARAMETER DESIGN

Theorem 2. Linear time-delay system (1) is stable for $|\dot{\tau}(t)| \leq d$, $1 < d \leq 2$ with mentioned quadratic performance $\|\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}\|_\infty \leq \gamma$ if there exist symmetric positive definite matrices $\boldsymbol{X} > 0$, $\boldsymbol{Z} > 0$, $\boldsymbol{X}, \boldsymbol{Z} \in I\!R^{n\times n}$, a matrix $\boldsymbol{Y} \in I\!R^{r\times n}$, and a scalar $\gamma > 0$, $\gamma \in I\!R$ such that

$$\boldsymbol{X} = \boldsymbol{X}^T > 0 \qquad \boldsymbol{Z} = \boldsymbol{Z}^T > 0 \tag{46}$$

$$\begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{A}_d\boldsymbol{Z} & \boldsymbol{B} & h\boldsymbol{X}(\boldsymbol{A}+\boldsymbol{A}_d)^T & \boldsymbol{X}\boldsymbol{C}^T \\ * & -\boldsymbol{Z} & \boldsymbol{0} & h\boldsymbol{Z}\boldsymbol{A}_d^T & \boldsymbol{0} \\ * & * & -\gamma\boldsymbol{I}_r & \boldsymbol{0} & \boldsymbol{0} \\ * & * & * & -\boldsymbol{Z} & \boldsymbol{0} \\ * & * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{47}$$

$$\boldsymbol{\Gamma}_{11} = (\boldsymbol{A} + \boldsymbol{A}_d)\boldsymbol{X} + \boldsymbol{X}(\boldsymbol{A} + \boldsymbol{A}_d)^T - \boldsymbol{B}\boldsymbol{Y} - \boldsymbol{Y}^T\boldsymbol{B}^T \tag{48}$$

Then the control law gain matrix be computed as

$$\boldsymbol{K} = \boldsymbol{Y}\boldsymbol{X}^{-1} \tag{49}$$

Proof. Choosing Lyapunov function candidate as

$$\begin{aligned} 0 < v(\boldsymbol{q}(t)) &= \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t) + \\ &+ \int_0^t (\boldsymbol{y}^T(r)\boldsymbol{y}(r) - \gamma\boldsymbol{u}^T(r)\boldsymbol{u}(r))\mathrm{d}r + \\ &+ \int_0^t (h^2\boldsymbol{y}^{\circ T}(r)\boldsymbol{Q}\boldsymbol{y}^\circ(r) - \boldsymbol{u}^{\circ T}(r)\boldsymbol{Q}\boldsymbol{u}^\circ(r))\mathrm{d}r \end{aligned} \tag{50}$$

where $\boldsymbol{P} = \boldsymbol{P}^T > 0$, $\boldsymbol{Q} = \boldsymbol{Q}^T > 0$, then derivative evaluating of $v(\boldsymbol{q}(t))$ with respect to t gives

$$\begin{aligned} \dot{v}(\boldsymbol{q}(t)) &= \boldsymbol{y}^T(t)\boldsymbol{y}(t) - \gamma\boldsymbol{u}^T(t)\boldsymbol{u}(t) + \\ &+ (\boldsymbol{q}^T(t)\boldsymbol{A}^{\circ T} + \boldsymbol{u}^{\circ T}(t)\boldsymbol{B}^{\circ T})\boldsymbol{P}\boldsymbol{q}(t) + \\ &+ \boldsymbol{q}^T(t)\boldsymbol{P}(\boldsymbol{A}^\circ\boldsymbol{q}(t) + \boldsymbol{B}^\circ\boldsymbol{u}^\circ(t)) + \\ &+ h^2(\boldsymbol{C}^\circ\boldsymbol{q}(t) + \boldsymbol{D}^\circ\boldsymbol{u}^\circ(t))^T\boldsymbol{Q}(\boldsymbol{C}^\circ\boldsymbol{q}(t) + \boldsymbol{D}^\circ\boldsymbol{u}^\circ(t)) - \\ &- \boldsymbol{u}^{\circ T}(t)\boldsymbol{Q}\boldsymbol{u}^\circ(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{u}^T(t)\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{q}(t) < 0 \end{aligned} \tag{51}$$

Introducing the composite vector $\boldsymbol{q}^\bullet(t)$ as follows

$$\boldsymbol{q}^{\bullet T}(t) = \begin{bmatrix} \boldsymbol{q}^T(t) & \boldsymbol{u}^{\circ T}(t) & \boldsymbol{u}^T(t) \end{bmatrix} \tag{52}$$

the Lyapunov function derivative (52) takes form

$$\dot{v}(\boldsymbol{q}^\bullet(t)) = \boldsymbol{q}^{\bullet T}(t)\boldsymbol{P}^\bullet\boldsymbol{q}^\bullet(t) < 0 \tag{53}$$

where

$$\boldsymbol{P}^\bullet = \begin{bmatrix} \boldsymbol{P}_{11}^\bullet & h^2\boldsymbol{C}^{\circ T}\boldsymbol{Q}\boldsymbol{D}^\circ + \boldsymbol{P}\boldsymbol{B}^\circ & \boldsymbol{B}\boldsymbol{P} \\ * & h^2\boldsymbol{D}^{\circ T}\boldsymbol{Q}\boldsymbol{D}^\circ - \boldsymbol{Q} & \boldsymbol{0} \\ * & * & -\gamma\boldsymbol{I}_r \end{bmatrix} < 0 \tag{54}$$

$$\boldsymbol{P}_{11}^\bullet = \boldsymbol{A}^{\circ T}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^\circ + h^2\boldsymbol{C}^{\circ T}\boldsymbol{Q}\boldsymbol{C}^\circ + \boldsymbol{C}^T\boldsymbol{C} \tag{55}$$

Thus, inequality (55) can be written as

$$\boldsymbol{P}^\bullet = \boldsymbol{P}_1^\bullet + \boldsymbol{P}_2^\bullet + \boldsymbol{P}_3^\bullet \tag{56}$$

with

$$\boldsymbol{P}_1^\bullet = \begin{bmatrix} \boldsymbol{A}^{\circ T}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^\circ & \boldsymbol{P}\boldsymbol{B}^\circ & \boldsymbol{P}\boldsymbol{B} \\ * & -\boldsymbol{Q} & \boldsymbol{0} \\ * & * & -\gamma\boldsymbol{I}_r \end{bmatrix} \tag{57}$$

$$\begin{aligned} \boldsymbol{P}_2^\bullet &= \begin{bmatrix} h^2\boldsymbol{C}^{\circ T}\boldsymbol{Q}\boldsymbol{C}^\circ & h^2\boldsymbol{C}^{\circ T}\boldsymbol{Q}\boldsymbol{D}^\circ & \boldsymbol{0} \\ h^2\boldsymbol{D}^{\circ T}\boldsymbol{Q}\boldsymbol{C}^\circ & h^2\boldsymbol{D}^{\circ T}\boldsymbol{Q}\boldsymbol{D}^\circ & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} = \\ &= \begin{bmatrix} \begin{bmatrix} h\boldsymbol{C}^{\circ T} \\ h\boldsymbol{D}^{\circ T} \end{bmatrix} \boldsymbol{Q}\,[h\boldsymbol{C}^\circ \; h\boldsymbol{D}^\circ] & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \geq 0 \end{aligned} \tag{58}$$

$$\boldsymbol{P}_3^\bullet = \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}^T \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} [\boldsymbol{C} \; \boldsymbol{0} \; \boldsymbol{0}] \geq 0 \tag{59}$$

Now, using Schur complement property it yields

$$\boldsymbol{P}_2^\bullet = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & h\boldsymbol{C}^{\circ T} \\ * & \boldsymbol{0} & \boldsymbol{0} & h\boldsymbol{D}^{\circ T} \\ * & * & \boldsymbol{0} & \boldsymbol{0} \\ * & * & * & -\boldsymbol{Q}^{-1} \end{bmatrix} \geq 0 \tag{60}$$

$$\boldsymbol{P}_3^\bullet = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}^T \\ * & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ * & * & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ * & * & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ * & * & * & * & -\boldsymbol{I}_m \end{bmatrix} \geq 0 \tag{61}$$

and subsequently

$$\boldsymbol{P}^\bullet = \begin{bmatrix} \boldsymbol{A}^{\circ T}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^\circ & \boldsymbol{P}\boldsymbol{B}^\circ & \boldsymbol{P}\boldsymbol{B} & h\boldsymbol{C}^{\circ T} & \boldsymbol{C}^T \\ * & -\boldsymbol{Q} & \boldsymbol{0} & h\boldsymbol{D}^{\circ T} & \boldsymbol{0} \\ * & * & -\gamma\boldsymbol{I}_r & \boldsymbol{0} & \boldsymbol{0} \\ * & * & * & -\boldsymbol{Q}^{-1} & \boldsymbol{0} \\ * & * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{62}$$

Defining the congruence transform matrix

$$\boldsymbol{T} = \mathrm{diag}\left[\boldsymbol{P}^{-1} \; \boldsymbol{Q}^{-1} \; \boldsymbol{I}_r \; \boldsymbol{I} \; \boldsymbol{I}_m\right] \tag{63}$$

and pre-multiplying left-hand side as well as right-hand side of (62) by $\boldsymbol{T}$ gives

$$\begin{bmatrix} \boldsymbol{P}_{11}^\diamond & \boldsymbol{B}^\circ\boldsymbol{Q}^{-1} & \boldsymbol{B} & h\boldsymbol{P}^{-1}\boldsymbol{C}^{\circ T} & \boldsymbol{P}^{-1}\boldsymbol{C}^T \\ * & -\boldsymbol{Q}^{-1} & \boldsymbol{0} & h\boldsymbol{Q}^{-1}\boldsymbol{D}^{\circ T} & \boldsymbol{0} \\ * & * & -\gamma\boldsymbol{I}_r & \boldsymbol{0} & \boldsymbol{0} \\ * & * & * & -\boldsymbol{Q}^{-1} & \boldsymbol{0} \\ * & * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{64}$$

$$\boldsymbol{P}_{11}^\diamond = \boldsymbol{P}^{-1}\boldsymbol{A}^{\circ T} + \boldsymbol{A}^\circ\boldsymbol{P}^{-1} \tag{65}$$

Denoting

$$\boldsymbol{P}^{-1} = \boldsymbol{X}, \; \boldsymbol{Q}^{-1} = \boldsymbol{Z}, \; \boldsymbol{Y} = \boldsymbol{K}\boldsymbol{P}^{-1} \tag{66}$$

and inserting

$$\begin{aligned} \boldsymbol{A}^\circ\boldsymbol{P}^{-1} &= (\boldsymbol{A} + \boldsymbol{A}_d)\boldsymbol{P}^{-1} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{P}^{-1} = \\ &= (\boldsymbol{A} + \boldsymbol{A}_d)\boldsymbol{X} - \boldsymbol{B}\boldsymbol{Y} \end{aligned} \tag{67}$$

$$\boldsymbol{C}^\circ = \boldsymbol{A} + \boldsymbol{A}_d, \quad \boldsymbol{B}^\circ = \boldsymbol{D}^\circ = \boldsymbol{A}_d \tag{68}$$

then (64), (65) implies (47), (48). This concludes the proof.

■

6. ILLUSTRATIVE EXAMPLE

The system is given by (1), (2), where $h = 2.5$,

$$\boldsymbol{A} = \begin{bmatrix} -2.6 & 0.0 & 0.8 \\ -1.2 & 0.2 & 0.0 \\ 0.0 & 0.5 & -3.0 \end{bmatrix}, \; \boldsymbol{B} = \begin{bmatrix} 0 & 2 \\ 3 & 1 \\ 1 & 0 \end{bmatrix}$$

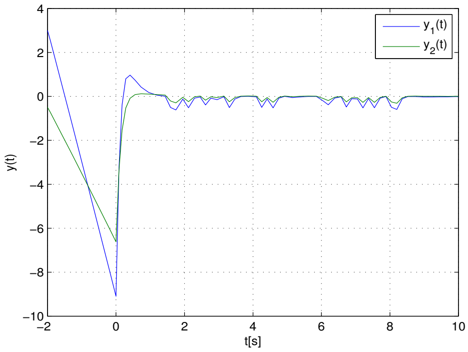


Fig. 1. Output of the system

$$\boldsymbol{A}_d = \begin{bmatrix} 0.00 & 0.02 & 0.00 \\ 0.00 & 0.00 & -1.00 \\ -0.02 & 0.00 & 0.00 \end{bmatrix}, \quad \boldsymbol{C}^T = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix}$$

Solving (46), (47) with respect to LMI matrix variables $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, γ using SeDuMi (Self-Dual-Minimization) package for Matlab (Peaucelle et al. (1994)) given task was feasible with

$$\boldsymbol{X} = \begin{bmatrix} 1.7362 & -0.7762 & -0.0051 \\ -0.7762 & 1.0524 & -0.1625 \\ -0.0051 & -0.1625 & 0.8490 \end{bmatrix}$$

$$\boldsymbol{Z} = \begin{bmatrix} 3.4802 & 0.0113 & 0.0097 \\ 0.0113 & 6.4817 & 0.0071 \\ 0.0097 & 0.0071 & 0.5562 \end{bmatrix}$$

$$\boldsymbol{Y} = \begin{bmatrix} 0.0611 & 1.6975 & -0.3312 \\ -0.7696 & 0.6231 & -0.0458 \end{bmatrix}$$

$$\gamma = 5.0659$$

and giving the control system parameters as follows

$$\boldsymbol{K} = \begin{bmatrix} 1.1377 & 2.4656 & 0.0886 \\ -0.2642 & 0.4004 & 0.0210 \end{bmatrix}$$

$$\boldsymbol{A}_c = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K} = \begin{bmatrix} -2.0716 & -0.8008 & 0.7579 \\ -4.3488 & -7.5972 & -0.2867 \\ -1.1377 & -1.9656 & -3.0886 \end{bmatrix}$$

$$\boldsymbol{A}_{cs} = \boldsymbol{A} + \boldsymbol{A}_d - \boldsymbol{B}\boldsymbol{K} = \begin{bmatrix} -2.0716 & -0.7808 & 0.7579 \\ -4.3488 & -7.5972 & -1.2867 \\ -1.1577 & -1.9656 & -3.0886 \end{bmatrix}$$

$$\rho(\boldsymbol{A}_c) = \{-1.3917, \ -3.2964, \ -8.0692\}$$

$$\rho(\boldsymbol{A}_{cs}) = \{-1.3742, \ -2.9561, \ -8.4271\}$$

It is evident, that the both sets of eigenvalues spectra $\rho(\boldsymbol{A}_c)$, $\rho(\boldsymbol{A}_{cs})$ of the closed loop system matrices are stable.

In the presented Fig. 1 the example is shown of the unforced closed-loop system output response, where the initial state was $\boldsymbol{q}^T(-2) = [-1 \ 0.5 \ 3]$, $h = 2.5$, $1 < d \leq 2$. It is possible to verify that closed-loop dynamic properties for this unstable autonomous time-delay system are better than any obtained using results implying from Lyapunov-Krasovskii inequality (Lyapunov-Krasovskii functional can stay negative).

7. CONCLUDING REMARKS

Stability conditions for autonomous linear time delay systems as well as the feedback control gain matrix parameter design method are derived in the paper. Considering the delay parameter as an unknown time-varying function with given upper bounds on the magnitude and the variation, the influence of time-varying delay is considered as perturbation in the system, and the presented algorithm gives necessary and sufficient conditions for design in the sense of H_∞ control if the time-delay variation is from the specified range. The advantage of this approach is that the results can be easily generalized for systems with multiple delays, and extended to deal with systems with parametric uncertainties.

ACKNOWLEDGMENTS

The work presented in this paper was supported by VEGA, Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/0256/11, as well by Research & Development Operational Programme Grant No. 26220120030 realized in Development of Center of Information and Communication Technologies for Knowledge Systems. These supports are very gratefully acknowledged.

REFERENCES

D. Boyd, L. El Ghaoui, E. Peron, and V. Bala-krishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1994.

R.S. Burns. *Advanced Control Engineering*, Butterworth-Heinemann, Oxford, 2001.

A. Filasová and D. Krokavec. Global asymptotically stable control design for time-delay systems. *AT&P Journal Plus*, 2, 89-92, 2009.

A. Filasová and D. Krokavec. Uniform stability guaranty control of the discrete time-delay systems, *Journal of Cybernetics and Informatics*, 10, 21-28, 2010.

P. Gahinet, A. Nemirovski, A.J. Laub, and M. Chilali. *LMI Control Toolbox User's Guide*, The MathWorks, Inc., Natick, 1995.

G. Herrmann, M.C. Turner, and I. Postlethwaite. Linear matrix inequalities in control, In *Mathematical Methods for Robust and Nonlinear Control*, Springer–Verlag, Berlin, 123-142, 2007.

C.Y. Kao and A. Rantzer. Robust stability analysis of linear systems with time-varying delays. In *Proceedings of 2005 IFAC World Congress, Prag, Czech Republic*, 2005.

N.N. Krasovskii. On the application of Lyapunov's second method for equations with time delays, *Prikladnaja matematika i mechanika*, 20, 315-327, 1956. (in Russian)

N.N. Krasovskii. *Stability of Motion: Application of Lyapunov's Second Method to Differential Systems and Equations with Delay*, Standford University Press, Standford, 1963.

Y. Nesterov and A. Nemirovsky. *Interior Point Polynomial Methods in Convex Programming. Theory and Applications*, SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1994.

S.I. Niculescu, E.I. Veriest, L. Dugard, and J.M. Dion. Stability and robust stability of time-delay systems: A guided tour. In *Stability and Control of Time-delay Systems*, Springer–Verlag, Berlin, 1-71, 1998.

D. Peaucelle, D. Henrion, Y. Labit, and K. Taitz. *User's Guide for SeDuMi Interface 1.04*, LAAS-CNRS, Toulouse, 2002.

U. Shaked, I. Yaesh, and C.E. De Souza. Bounded real criteria for linear time systems with state-delay. *IEEE Transactions on Automatic Control*, 43, 1116–1121, 1998.

R.E. Skelton, T. Iwasaki, and K. Grigoriadis. *A Unified Algebraic Approach to Linear Control Design*, Taylor & Francis, London, 1998.

M. Sun and Y.Gu. Delay-dependent robust H2 control for discrete systems with time-delay and polytopic uncertainty. In *Proceedings of the American Control Conference 2010, Baltimore, MD, USA*, pp. 5789-5793.

V. Veselý and A. Rosinová. Robust output model predictive control design. BMI approach, *International Journal of Innovative Computing, Information and Control*, 5:4, 1115-1123, 2009.

M. Wu, Y. He, J.J. She, and G.P. Liu. Delay-dependent criteria for robust stability of time-varying delay systems, *Automatica*, 40, 1435-1439, 2004.

M. Wu, Y. He, and J.H. She. *Stability Analysis and Robust Control of Time-Delay Systems*, Springer-Verlag, Berlin, 2010.

Fault Accommodation in Nonlinear Time Delay Systems

A. N. Zhirabok*, A.Ye. Shumsky*, Ye.Yu. Bobko**

** Institute of Applied Mathematics, Vladivostok, Russia*
e-mail : zhirabok@mail.ru shumsky@mail.primorye.ru
***Far Eastern Federal University, Vladivostok, Russia*
e-mail : spray-only@inbox.ru

Abstract: Solution to the problem of fault accommodation in nonlinear time delay dynamic systems is related to constructing the control law which provides full decoupling with respect to fault effects. Existing conditions are formulated and calculating relations are given for the control law.

1. INTRODUCTION

Fault tolerant control (FTC) is a tool intended for increasing a reliability and safety for critical purpose control systems. The goal of FTC is to determine such control law that preserves the main performances of the faulty system while the minor performances may degrade. There are two principal approaches to FTC. The first of them involves adaptive control techniques and assumes on-line fault detection and estimation followed by control law accommodation, see, e.g. (Blanke *et al* 2003, Jang *et al* 2006, Staroswiecki *et al* 2006). The second approach is focused on such control law determination which provides full decoupling with respect to fault effects in the output space of the system. In contrast to the first approach, the second one does not need in fault estimation. Therefore, such approach looks reasonable if on-line fault estimation is impossible.

Full decoupling problem solution under appropriate its statement has been obtained in (Isidori 1995) for affine systems. But the possibility of this solution applying in the framework of FTC problem is strictly limited by the demand on the system state vector availability (this vector is immediately included into the control law description). As a rule, not all components of the state vector are immediately measurable in practice, and estimation of full state vector for the system with unknown (affected by the faults) dynamics is impossible.

In (Shumsky *et al* 2009, Shumsky & Zhirabok 2010) a solution to the accommodation problem in nonlinear systems has been obtained on the basis of algebra of functions and differential geometry (Shumsky & Zhirabok 2006). In present paper, this problem is solved for time delay systems. These systems form an important class of nonlinear systems. They are used to represent a wide variety of processes and systems including hydraulic/pneumatic systems, communication systems, biological systems, *etc*. To solve the problem of fault accommodation for this class of systems, we use so-called logic-dynamic approach which allows obtaining a solution for time delay nonlinear systems with no differentiable nonlinearities using linear methods. Besides, we use more sophisticated treatment in contrast to the paper (Shumsky *et al* 2009) which allows obtaining in some cases more simple solution with the point of view of designed systems dimensions.

Consider nonlinear systems described by equation

$$\begin{aligned} \dot{x}(t) &= Fx(t) + F_d x(t-\tau) + C\varphi(Ax(t), u(t)) + \\ &\quad Gu(t) + L\vartheta(t), \\ y(t) &= Hx(t). \end{aligned} \tag{1}$$

In (1), x, y, and u are vectors of state, output, and control; F, F_d, C, A, G, L, and H are known matrices of appropriate dimensions; φ is an arbitrary scalar nonlinear function, $\vartheta(t) \in R^v$ is the vector describing the fault. Assume that for healthy system it holds $\vartheta(t) = 0$. For simplicity, the system with a single nonlinearity is considered. Denote system (1) as Σ.

It is assumed that fault detection procedure is performed by known methods (Blanke *et al* 2003). If a fault occurs, $\vartheta(t)$ becomes an unknown function, and a solution to the control problem based on model (1) becomes impossible. To overcome this difficulty, it is suggested to obtain the vector $u(t)$ according to

$$u(t) = g(y(t), x_0(t), u_*(t)) \tag{2}$$

for some function g where $u_*(t) \in R^m$ is a new control vector, $x_0(t) \in R^q$, $q \le n$, is a state vector of the system has to be determined and described by equation

$$\begin{aligned} \dot{x}_0(t) &= F_0 x_0(t) + F_{d0} x_0(t-\tau) + \\ &C_0\varphi(A_0\begin{pmatrix} x_0(t) \\ y(t) \end{pmatrix}, u(t)) + G_0 u(t) + J_0 y(t). \end{aligned} \tag{3}$$

Note that model (3) does not depend on the unknown vector $\vartheta(t)$.

Assume that the model obtained by substitution (2) into (1) can be transformed to the form

$$\dot{x}_*(t) = F_* x_*(t) + F_{d*} x_*(t-\tau) + C_* \varphi(A_* x_*(t), u_*(t)) + G_* u_*(t) \tag{4}$$

with $x_*(t) \in R^p$, $p \le q$. If the control (2) exists and the fault occurred and detected, then a solution to the control problem is performed on the basis of model (4) which does not contain the unknown vector $\vartheta(t)$. As a result, fault accommodation effect is achieved. Scheme for system Σ control is shown in Figure 1.

Note that the use of the control (2) assumes moving system (1) only in some subspace of its state space which corresponds to the state space of system (4). Under this, the goal of control should be achieved by appropriate choosing the trajectory belonging to this subspace. The need of appropriate trajectory existence (or a possibility to correct the goal of control for finding appropriate trajectory) restricts the sphere of the considered approach application.

The problem is to determine the existing condition for the control (2) and to obtain all matrices describing systems (3) and (4). To solve this problem, is it necessary initially to design the auxiliary system Σ' described by equation

$$\dot{x}'(t) = f'(x'(t), y(t), u(t)) = F'x'(t) + F'_d x'(t-\tau) + C'\varphi(A'\begin{pmatrix} x'(t) \\ y(t) \end{pmatrix}, u(t)) + G'u(t) + J'y(t). \tag{5}$$

2. LOGIC-DYNAMIC APPROACH

So-called logic-dynamic approach developed in (Zhirabok & Usoltsev 2002) will be used for designing the system Σ'. The feature of this approach is the use of conventional linear algebraic tools in contrast to nonlinear algebraic and differential geometric tools of the work (Shumsky & Zhirabok 2006).

The logic-dynamic approach for systems in the form (1) includes the following three steps.

Step 1. Replacing the initial nonlinear system (1) by certain linear system.

Step 2. Solving the problem under consideration for this linear system with some additional restrictions.

Step 3. Transforming the obtained linear system into the nonlinear one by adding a nonlinear term.

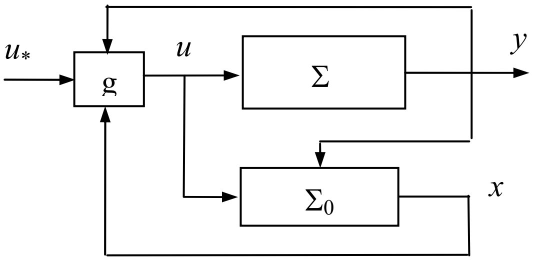


Fig. 1. Scheme for system Σ control

At the first step of this approach, the nonlinear term $C\varphi(Ax(t), u(t))$ is removed from system (1). The corresponding linear system is of the form

$$\dot{x}(t) = Fx(t) + F_d x(t-\tau) + Gu(t) + L\vartheta(t),$$
$$y(t) = Hx(t). \tag{6}$$

It will be named the linear part of system (1).

At the second step, according to the logic-dynamic approach, a linear part of system (5) is designed. It is well-known from the fault detection and isolation theory of linear systems (Frank 1990) that for this linear part design, the state x' is a linear combination of system (6) state according to

$$\Phi x(t) = x'(t)$$

in the unfaulty case after the response to unlike conditions has died out. We will say, with this equality in mind, that system (5) estimates the initial system state vector with accuracy to a function realized by the matrix Φ. In the absence of faults, the following set of equations can be obtained by analogy with (Zhirabok & Usoltsev 2002):

$$\Phi F = F'\Phi + J'H, \quad \Phi F_d = F'_d \Phi,$$
$$G' = \Phi G. \tag{7}$$

It follows immediately from definition of the matrix Φ and (7) that

$$\Phi C\varphi(Ax, u) = C'\varphi(A'\begin{pmatrix} x'(t) \\ y(t) \end{pmatrix}, u(t)) = C'\varphi(A'\begin{pmatrix} \Phi x(t) \\ Hx(t) \end{pmatrix}, u(t)).$$

This equality is true if the following relations hold:

$$C' = \Phi C, \qquad A = A'\begin{pmatrix} \Phi \\ H \end{pmatrix}.$$

One can show that the relation $A = A'\begin{pmatrix} \Phi \\ H \end{pmatrix}$ is equivalent to the equality

$$rank\begin{pmatrix} \Phi \\ H \end{pmatrix} = rank\begin{pmatrix} \Phi \\ H \\ A \end{pmatrix}. \tag{8}$$

By analogy, it can be shown that the second equation in (7) is equivalent to equality

$$rank(\Phi) = rank\begin{pmatrix} \Phi \\ \Phi F_d \end{pmatrix}. \tag{9}$$

These conditions are those mentioned at Step 2. If the matrix Φ satisfies the first equation in (7) and these conditions, the problem under consideration can be solved.

To ensure if system (3) is independent of the unknown vector $\vartheta(t)$, or if the full decoupling demand is fulfilled, the equality $\Phi L = 0$ has to hold.

3. SYSTEM Σ' DESIGN

The matrix Φ can be obtained as follows. Introduce the matrix L^0 of maximal row rank such that $L^0 L = 0$. Condition $\Phi L = 0$ implies the equality

$$\Phi = NL^0$$

for some matrix N. Replace the matrix Φ in the first equation in (7) with NL^0 that gives $NL^0 F = F'NL^0 + J'H$ and transform it as follows:

$$(N \quad -F'N \quad -J') \cdot \begin{pmatrix} L^0 F \\ L^0 \\ H \end{pmatrix} = 0 . \tag{10}$$

Expression (10) can be considered as an algebraic equation for the matrices N, F', and J'.

Let the matrix (A B C) presents all linearly independent solutions to equation (10), i.e.

$$(A \quad B \quad C) \cdot \begin{pmatrix} L^0 F \\ L^0 \\ H \end{pmatrix} = 0 . \tag{11}$$

To use the matrices A and B for the system Σ' design, the relation $B = -F'A$ for the matrix F' must hold according to (10) and (11). To obtain these matrices, find rows of the matrix B which are independent of the matrix A rows and remove them from (A B C). Denote the obtained matrix as (A^0 B^0 C^0). Set $N = A^0$ and $\Phi = NL^0$. If the matrix Φ satisfies conditions (8) and (9), the system Σ' can be built otherwise the problem under consideration is not solvable because in this case full decoupling can not be achieved.

Suppose that conditions (8) and (9) hold. Take $G' = \Phi G$ and $J' = -C^0$; the matrices F' and F'_d are solutions to the algebraic equations $F'N = -B^0$ and $\Phi F_d = F'_d \Phi$ respectively.

As a result, a linear part of the system Σ' is described by the following equation:

$$\dot{x}'(t) = F'x'(t) + F'_d x'(t-\tau) + G'u(t) + J'y(t) . \tag{12}$$

At the third step of design, it is necessary to transform the obtained linear system into the nonlinear one. According to (Zhirabok & Usoltsev 2002), the nonlinear term

$$C'\varphi(A'\begin{pmatrix} x'(t) \\ y(t) \end{pmatrix}, u(t))$$

with the matrices $C' = \Phi C$ and A' obtained from the algebraic equation

$$A = A'\begin{pmatrix} \Phi \\ H \end{pmatrix}$$

must be added to the right-hand side of equation (12) that gives

$$\begin{aligned} \dot{x}'(t) = f'(x'(t), y(t), u(t)) = F'x'(t) + F'_d x'(t-\tau) + \\ C'\varphi(A'\begin{pmatrix} x'(t) \\ y(t) \end{pmatrix}, u(t)) + G'u(t) + J'y(t). \end{aligned} \tag{13}$$

Note that the algebraic equations $A = A'\begin{pmatrix} \Phi \\ H \end{pmatrix}$ and $\Phi F_d = F'_d \Phi$ are solvable because conditions (8) and (9) hold respectively.

4. CONTROL LAW DETERMINATION

To carry out an analysis of the system Σ', introduce the matrices H' and R whose rows present all linearly independent solutions to the algebraic equation

$$(H' \quad -R)\begin{pmatrix} \Phi \\ H \end{pmatrix} = 0 .$$

Note that the vector Ry presents those components of the vector y and their linear combinations which can be computed as a function of the state vector x', i.e. $Ry = H'x'$. Consider two cases.

(1) Every component of the function f' contains only those components of the vector y which depend on the vector Ry components; in this case set

$$u_j = u_{*j} , \qquad j = 1,...,m .$$

This means that a block "g" in Figure1 is absent.

(2) Suppose that Case 1 does not hold and find in the function f' all terms with minimal numbers of variables in the form

$$\alpha_i(x', y, u) , \qquad i = 1,...,r ,$$

which contain the control u and components of the vector y functionally independent of the vector Ry (some terms do not contain the variable x'). Denote

$$\begin{aligned} u_{*1} &= \alpha_1(x', y, u), \\ &\vdots \\ u_{*r} &= \alpha_r(x', y, u). \end{aligned} \tag{14}$$

To check solvability of these nonlinear algebraic equations, assume that

$$rank\left(\frac{\partial\alpha}{\partial u}\right) = s$$

for all x', y, and u except perhaps on a set of measure zero, where $\alpha = (\alpha_1 \ \dots \ \alpha_r)^{\mathrm{T}}$, and the function α contains m', $m' \le m$, components of the vector u as its arguments. It is obvious that the inequalities $m' \ge s$ and $r \ge s$ hold by definition of m', r, and s. Assume for simplicity that if some u_j is contained in the function α, then it is not in other part of the function f'. Consider three cases.

(1) $m' = r = s$; in this case the system of equations (14) is solvable for some m' components of the control vector u (without loss of generality suppose that they are $u_1, \dots, u_{m'}$):

$$u_j = \gamma_j(x', y, u_*), \qquad j = 1,...,m'. \tag{15}$$

Take

$$u_j = u_{*j}, \qquad j = m'+1,...,m.$$

(2) $m' > r = s$; in this case the function α contains $m' - r$ redundant components of the vector u. Without loss of generality assume that these components are the last $m' - r$ ones, i.e. $u_{r+1}, \dots, u_{m'}$. Using additional equations for these components

$$u_j = u_{*j}, \qquad j = r+1,...,m, \tag{16}$$

one can solve the system of equations (14) in the form (15) for $j = 1,...,r$.

(3) $m' \ge r > s$ or $r > m' \ge s$; in these cases find the matrix P with s rows such that

$$rank\left(P\frac{\partial\alpha}{\partial u}\right) = s$$

for all x', y, and u except perhaps on a set of measure zero. The matrix P collects s functionally independent components from all ones of the function α. The redundant components $u_{s+1}, \dots, u_{m'}$ (when $m' > s$) are now in the function $P\alpha$. Using (16) for $j = s+1,...,m$, one can solve the equation

$$u_* = P\alpha$$

in the form (15) for $j = 1,...,s$.

5. SYSTEM Σ_0 DESIGN

Note that in some cases all relations in (15) do not depend on the components of the vector x'; in this case the system Σ_0 is absent. In some cases these relations depend on all components of the vector x'; in this case the system Σ_0 coincides with Σ'.

Generally, (15) depends on some components of the vector x', in this case it is necessary to design the nontrivial system Σ_0. To do this, define the matrix Q (by analogy with Φ) such that

$$Qx'(t) = x_0(t) \qquad \forall t. \tag{17}$$

It can be shown that the following set of equations

$$\begin{pmatrix} F_0 \\ F_{d0} \end{pmatrix} Q = \begin{pmatrix} QF' \\ QF'_d \end{pmatrix}, \tag{18}$$

$$A_{01}Q = A'_1 \tag{19}$$

holds where $A' = (A'_1 \ \ A'_2)$, $A_0 = (A_{01} \ \ A_{02})$. Equations (18) and (19) are equivalent to equations

$$rank(Q) = rank\begin{pmatrix} Q \\ QF' \\ QF'_d \end{pmatrix}, \tag{20}$$

$$rank(Q) = rank\begin{pmatrix} Q \\ A'_1 \end{pmatrix} \tag{21}$$

respectively. The matrices F_0, F_{d0}, and A_{01} are obtained from (18) and (19) respectively, other matrices described the system Σ_0 can be found as follows:

$$J_0 = QJ', \qquad G_0 = QG',$$
$$C_0 = QC', \qquad A_{02} = A'_2. \tag{22}$$

Equality (17) is used for replacing the vector x' in (15) by x_0. As a result, (15) is transformed into

$$u_j = g_j(x_0, y, u_*), \qquad j = 1,...,m',$$

corresponding to the general law (2).

The matrix Q can be constructed according to the following procedure. Let $x'^{(1)}$ be a subvector of x' whose components are in the function α and $x'^{(1)} = Q^{(1)}x'$. Consider three cases.

(1) If conditions (20) and (21) hold with $Q = Q^{(1)}$, then set $Q = Q^{(1)}$ and define the matrices F_0, F_{d0}, J_0, G_0, C_0, A_0 from (18), (19), and (22) respectively.

(2) If some rows of the matrix A'_1 (denote them A'') do not satisfy condition (21), set $Q^{(2)} = \begin{pmatrix} Q^{(1)} \\ A'' \end{pmatrix}$, otherwise $Q^{(2)} = Q^{(1)}$. If condition (20) holds with $Q = Q^{(2)}$, then set $Q = Q^{(2)}$ and define the matrices F_0, F_{d0}, J_0, G_0, C_0, A_0 from (18), (19), and (22) respectively.

(3) If some rows of the matrix $\begin{pmatrix} Q^{(2)}F' \\ Q^{(2)}F'_d \end{pmatrix}$ (denote them $Q'^{(2)}$) do not depend on the rows of the matrix $Q^{(2)}$, then set $\mathrm{Q}^{(3)} = \begin{pmatrix} Q^{(2)} \\ Q'^{(2)} \end{pmatrix}$ and check condition (20). If it holds, set $Q = Q^{(3)}$ and define the matrices F_0, A_{01}, J_0, G_0, C_0, A_{02} from (18), (19), and (22) respectively. Otherwise repeat above operations until condition (20) satisfies. Let Q be equal to the final matrix $Q^{(*)}$; define the matrices F_0, F_{d0}, J_0, G_0, C_0, A_0 from (18), (19), and (22) respectively.

As a result, the system Σ_0 is described as follows:

$$\dot{x}_0(t) = F_0 x_0(t) + F_{d0} x_0(t-\tau) + C_0 \varphi(A_0 \begin{pmatrix} x_0(t) \\ y(t) \end{pmatrix}, u(t)) + G_0 u(t) + J_0 y(t).$$

6. SYSTEM Σ_* DESIGN

Consider two kinds of components of the function f': the first kind contains components of the vector y functionally independent of the vector Ry and do not depend on the control u; the second kind contains those components of the function α which are not in the function $P\alpha$ (if $P\alpha \neq \alpha$). Denote a set of these components numbers by $N = (n_1, n_2, ..., n_k)$.

If $N \neq \varnothing$, then the system Σ' contains components of the vector y which can not be decoupled from the unknown function $\vartheta(t)$. In this case Σ' must be redesigned as follows. Remove rows with numbers $n_1, n_2, ..., n_k$ from the matrix $(A^0 \ \ B^0 \ \ C^0)$ and analyze the obtained matrix by analogy with the matrix $(A \ B \ C)$. Denote the obtained matrix as $(A^{00} \ \ B^{00} \ \ C^{00})$ and set

$$N = A^{00}, \quad \Phi = NL^0, \quad G' = \Phi G, \quad J' = -C^{00};$$

the matrix F' is a solution to the equation $F'N = -B^{00}$ (we denote the redesigned system and its elements as the initial ones for simplicity). Other matrices of the redesigned system can be obtained as it is described above.

If $N = \varnothing$, then the system Σ' is not need to be redesigned. Assume that the general description of the initial or redesigned system is given by (13).

Consider the redesigned system Σ' and find all terms in the form

$$\alpha_i(x', y, u), \qquad i = 1, 2, ..., r,$$

which are investigated in Section 4. Replace all these terms by components of the new control vector u_* according to (14). The system Σ' may contain components of the vector y which depend on the vector Ry only. These components must be replaced by components of the vector x' as follows. Suppose that some y_j is in Σ' and $y_j = \delta(Ry)$ for some function δ. Then

$$y_j = \delta(Ry) = \delta(RHx) = \delta(H'\Phi x) = \delta(H'x').$$

Take

$$x_{*j} = x'_j, \ j = 1, ..., p = \dim x'.$$

These replacements transform the system Σ' into the system Σ_*.

7. ILLUSTRATIVE EXAMPLE

Consider the system described by the model

$$\dot{x}(t) = \begin{pmatrix} -x_1(t) - x_1(t)x_5(t) + u_2(t) \\ x_1(t-\tau) + u_1(t) - x_1(t) - x_2(t) - \vartheta(t) \\ x_4(t) - x_1(t) - x_2(t) + u_1(t) - \vartheta(t) \\ x_3(t) - x_2(t) - x_4(t) \\ x_1(t) + x_2(t) + \vartheta(t) \end{pmatrix},$$

$$y(t) = \begin{pmatrix} x_1(t) \\ x_4(t) \\ x_5(t) \end{pmatrix}.$$

The following matrices can be chosen for logic-dynamic description of the initial system:

$$F = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & -1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}, \quad F_d = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad L = \begin{pmatrix} 0 \\ -1 \\ -1 \\ 0 \\ 1 \end{pmatrix},$$

$$A^{(1)} = (1 \ \ 0 \ \ 0 \ \ 0 \ \ 0), \quad A^{(2)} = (0 \ \ 0 \ \ 0 \ \ 0 \ \ 1),$$

$$\varphi(A^{(1)}x, A^{(2)}x, u) = x_1 x_5.$$

The matrix L^0 is computed as follows:

$$L^0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Equation (11) $(A \quad B \quad C)\begin{pmatrix} L^0F \\ L^0 \\ H \end{pmatrix} = 0$ has several independent solutions for matrices A^0, B^0, and C^0, one of them is

$$(A^0 \quad B^0 \quad C^0) =$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & -1 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Therefore

$$N = A^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \Phi = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$F' = \begin{pmatrix} -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 \end{pmatrix}, \quad J' = -C^0 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$G' = \Phi G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad C' = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

It can be shown that

$$F'_d = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$A'^{(1)} = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0),$$

$$A'^{(2)} = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1).$$

It is easily to check that conditions (8) and (9) hold. As a result, system (3) is described as follows

$$\dot{x}'(t) = F'x'(t) + F'_d x'(t-\tau) + J'y(t) + C'\varphi(A'x(t), u(t)) =$$

$$\begin{pmatrix} -x'_1(t) + x'_1(t)y_3(t) + u_2(t) \\ x'_1(t-\tau) + u_1(t) \\ y_2(t) + u_1(t) \\ -x'_2(t) + x'_3(t) - y_2(t) \end{pmatrix}.$$

An analysis shows that

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad Ry = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad N = \varnothing .$$

Because the first equation contains the variable y_3 which is functionally independent of Ry, then take $\alpha(x', y, u) = x'_1 y_3 + u_2$, $r = m' = s = 1$. Set $u_{*2} = x'_1 y_3 + u_2$, then $u_2 = u_{*2} - x'_1 y_3$. It is easy to show that $Q^{(1)} = (1 \quad 0 \quad 0 \quad 0)$. One can check that conditions (20) and (21) hold, then $Q = Q^{(1)}$, $x_{01} = x'_1$, and the system Σ_0 is described by equation

$$\dot{x}_{01}(t) = -x_{01}(t) - x_{01}(t)y_3(t) + u_2(t).$$

Finally, law (2) takes a form

$$u_1 = u_{*1}, \quad u_2 = u_{*2} - x_{01}y_3. \tag{23}$$

Since $N = \varnothing$, the system Σ' is not need to be redesigned. Description of the system Σ_* is as follows:

$$\dot{x}_* = \begin{pmatrix} -x_{*1}(t) + u_{*2}(t) \\ x_{*1}(t-\tau) + u_{*1}(t) \\ y_2(t) + u_{*1}(t) \\ -x_{*2}(t) + x_{*3}(t) - y_2(t) \end{pmatrix}. \tag{24}$$

It should be noted that methods suggested in the paper (Shumsky *et al* 2009) can be modified for time delay systems. Applying them to the considered example, one obtains the following results: the system Σ_0 coincides with Σ' and is 4-dimensional in contrast to the above 1-dimensional system Σ_0; an equation for the control u_1 is not trivial in contrast to our case $u_1 = u_{*1}$; the system Σ_* is 2-dimensional in contrast to our 4-dimensional system (24).

For simulation, set $\tau = 1$, the control $u_1(t) = \sin t$, $u_2(t) = 5\sin t$. The function $\vartheta(t)$ is modelled by variate with the mean equal to zero and the variance equal to 20. Figure 2 shows the output y_1 behaviour under $\vartheta(t) = 0$; Figure 3 shows the output y_1 behaviour under $\vartheta(t) \neq 0$ without use of the law (2); Figure 4 shows the output y_1 behaviour under $\vartheta(t) = 0$ with use of the law (23).

Clearly, this law provides full decoupling the output y_1 with respect to the fault, and the fault accommodation effect has been achieved.

8. CONCLUSION

The problem of fault accommodation in nonlinear time delay systems has been studied. More general case with several nonlinearities can be considered based on the logic-dynamic approach by analogy with (Zhirabok & Usoltsev 2002). Since this approach uses linear operations only, it is easy to show that the theory described in the paper can be applied to discrete-time dynamic systems.

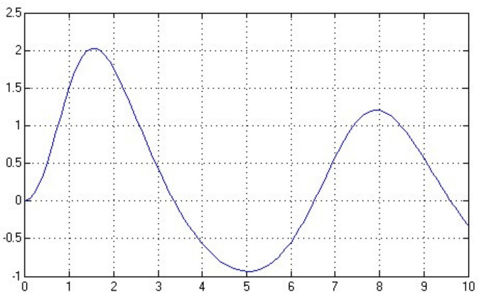


Fig. 2. Output y_1 behaviour under $\vartheta(t) = 0$

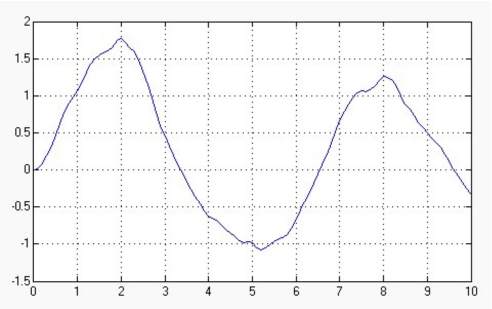


Fig. 3. Output y_1 behaviour under $\vartheta(t) \neq 0$ without correction of the input u_2

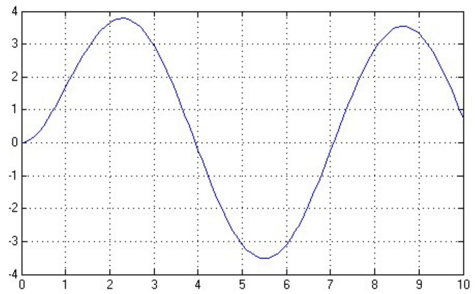


Fig. 4. Output y_1 behaviour under $\vartheta(t) \neq 0$ with the input u_2 corrected according (23)

ACKNOWLEDGMENTS

The work has been supported by the Russian Foundation of Basic Researches, grants 10-08-00133_a, 10-08-91220_CT, and 11-08-91151_GFEN.

REFERENCES

Blanke, M., M. Kinnaert, J. Lunze and M. Staroswiecki (2003). *Diagnosis and Fault Tolerant Control*, Springer-Verlag.

Frank, P. (1990). Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – A survey and some new results, *Automatica*, **26**(3), pp. 459-474.

Isidori, A. (1995). *Nonlinear control systems*, Springer-Verlag, 3d Ed.

Jang, B., M. Staroswiecki and V. Cocquempot (2006). Fault accommodation for nonlinear dynamic systems, *IEEE Transactions on Automatic Control,* **51**, pp. 1578-1583.

Staroswiecki, M., H. Yang and B. Jiang (2006). Progressive accommodation of aircraft actuator faults, *CD-ROM Proc. of the IFAC Symposium Safeprocess'2006*, Beijing, pp. 877-882.

Shumsky, A. and A. Zhirabok (2006). Nonlinear diagnostic filter design: Algebraic and geometric points of view, *Int. J. Applied Mathematics and Computer Science*, **16**, pp. 115-127.

Shumsky, A., A. Zhirabok, B. Jiang and K. Zhang (2009). Fault accommodation in dynamic systems: fault decoupling based approach, *CD ROM Proc. IEEE Conference CDC'2009*, Shanghai, China, Dec. 2009, pp. 8464-8469.

Shumsky, A., A. Zhirabok (2010). Unified approach to the problem of full decoupling via output feedback, *European Journal of Control*, **16**, No 4, pp. 313-325.

Zhirabok, A. and S. Usoltsev (2002). Fault diagnosis for nonlinear dynamic systems via linear methods, *CD ROM Proc. of the 15th World Congress IFAC*, Spain, Barcelona, 2002.

An Application of Model Predictive Control to a Gasoline Engine

Stefan Behrendt * **Peter Dünow** * **Bernhard P. Lampe** **

* *Hochschule Wismar, University of Applied Sciences,*
Faculty of Engineering, Germany
(Tel: +49 (0)3841 753 154; e-mail: stefan.behrendt@hs-wismar.de)
** *University of Rostock, Faculty of Computer Science and Electrical*
Engineering, Institute of Automation, Germany
(e-mail: bernhard.lampe@uni-rostock.de)

Abstract: The coordination of available actuators in modern gasoline engines is a challenging task. An available scheme for efficient coordination that respects the actuator constraints is model predictive control, but a specialised implementation of the incorporated optimisation algorithm is necessary to cope with the timing requirements. The numerical efficiency of the developed algorithm and the performance of the realised torque and speed control are presented in simulation and real-time in a Volkswagen T5 transporter respectively.

Keywords: Engine control, Model Predictive Control, Torque Coordination

1. INTRODUCTION

The number of the actuators in modern gasoline engines increases due to rising requirements regarding emissions and fuel consumption. The coordination of the available actuators (e.g. throttle, ignition plug, exhaust gas recirculation and turbo charging) is a challenging task. The electronic control unit (ECU) needs to calculate appropriate signals, which are hardly optimal because of the disregarded coupling within the actual structure, the non-linear characteristics and limitations on the actuator signals. These computed signals are mostly based on heuristics and look-up tables that are to a large extent chosen manually. This leads to high expenses and a later time-to-market.

A possibility to arrange the available torque sources in a systematic way is model predictive control (MPC). By superordinating this controller an efficient coordination of the subsidiary torque control structures is possible, due to the predictive nature of the controller and inherent consideration of constrained actuators.

Over the decades the MPC concept has proved successfully in controlling plants with complex dynamics. Due to its high computational complexity its usage is being limited to plants with slow dynamics like in the process industry. Numerous companies developed reliable software for process automation systems (PAS) (e.g. Aspen Technology, Inc. (2010)) and programmable logic controllers (PLC) (e.g. Siemens AG (2008)).

In contrast, controlling plants with fast dynamics still poses a problem. The fast sampling times necessary are perceived to prevent the MPC of these plants on standard embedded systems, beside the interest in this control approach is growing in the automotive industry and the benefits are already proven by e.g. Saerens et al. (2008) and del Re et al. (2010). To address this limitation several techniques have been developed to enlarge the field of MPC to embedded systems with small sampling times.

An implementation with sufficient worst case timing for the MPC of a single-input single-output (SISO) system is presented by Wills et al. (2008). It suggests the enhancement of embedded systems by a digital signal co-processor (DSP) for the fast evaluation of the underlying algorithms.

The implementation strategies and the actual implementation on field programmable gate array (FPGA) chips are presented by Knagge et al. (2009) and Ling et al. (2006) respectively. While the former address the specific architecture, like parallelism, explicitly and therefore the greater improvement is expected, the effort of implementing the necessary algorithms in a hardware description language should not be underestimated. To compete against high-potential micro-controllers in terms of computational time is a challenging task.

Another approach is the combination of on-line optimisation and a partial enumeration method presented by Pannocchia et al. (2006). The solutions of the optimisation problem with active constraint sets that appear with highest frequency are computed off-line and stored in a table. This table is searched on-line for the best control. In case, that meanwhile the on-line computation an active constraint set does not exist in the table, it is adapted. With this method a significant speed-up is possible. The drawbacks are performance degradation and the memory requirements.

The explicit MPC has gained much attention in the recent years. Therefore the state space is partitioned into polyhedral regions. The control law is formulated as a function of the plant state and the piecewise linear solutions to the control problem with respect to the constraints are calculated as described by Bemporad et al. (2002). The on-line computational complexity reduces to the selection of the appropriate control law depending on the actual state. Numerous successful applications followed by e.g. Ortner et al. (2006), Naus et al. (2008) and Arce et al. (2009). On the other hand, explicit MPC is limited to plants with a small number of inputs and short control horizons (for an explanation of this term see section 2) as the number of regions grows exponentially with these parameters. A prohibitively large amount of memory would be necessary, which is addressed e.g. by Rossiter and Grieder (2004). Additionally the control parameters and the constraints of the actuating variables are commonly fixed, which is undesired in some control problems. This limitation is addressed by Baric et al. (2005), but results in additional optimization variables and therefore a larger demand on memory. Hence, it quickly exceeds available resources in practical situations.

With the advent of multi-core controllers for embedded systems (e.g. XMOS Ltd. (2010)) and multi-core DSPs (e.g. Texas Instruments Inc. (2010)) it seems reasonable that parallel algorithms for MPC could lower the computational burden. To the knowledge of the author the main developments occur in the field of large-scale and sparse problems (e.g. Gondzio and Grothey (2006) and Gondzio and Grothey (2007)) and focus on the parallelisation of incorporated operations (e.g. matrix multiplication and inversion) as by Ruano and Daniel (1997). In a recent publication by Behrendt et al. (2010) a parallelisation approach on the optimisation level is presented that allows for super-linear speed-up of the computation at time-critical sampling instances.

The purpose of this paper is to show that the MPC of gasoline engines on a commonly used ECU is already feasible by utilising a specifically tailored algorithm and a limited number of actuators. This conclusion is evident by simulative results and by measurements in a Volkswagen T5 transporter with a 2l TFSI (Turbo Fuel Stratified Injection) engine.

The paper is organised as follows. The basics of MPC and the features of the applied algorithm for the solution of the incorporated optimisation are presented in section 2. The section 3 explains the utilised model structure. In section 4 the numerical properties are obtained by a Hardware-In-The-Loop simulation and the practical relevance is proven by controlling the engine of the Volkswagen T5.

2. MODEL PREDICTIVE CONTROL BASICS

In this section a short introduction to the MPC fundamentals is presented. For a more comprehensive survey on the theory of MPC the reader is referred to Maciejowski (2002) and Camancho and Bordons (2004).

The MPC method combines the advantages of predicting the behaviour of the plant, namely the output, and respects constraints on the actuators. Therefore the cost function

$$J(k) = \sum_{i=1}^{H_p} \|\hat{y}(k+i|k) - w(k+i|k)\|^2_{Q(i)} + \sum_{i=0}^{H_u-1} \|\Delta u(k+i|k)\|^2_{R(i)} \tag{1}$$

with the prediction horizon H_p, the control horizon H_u, the weights on the control deviation Q, the weights on the rate of change of the difference control action R, the predicted output $\hat{y}$, the reference w and the difference control action Δu needs to get minimized with respect to Δu.

The prediction follows from the state equations of the discrete-time linear plant

$$x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k) \tag{2}$$

$$y(k) = \mathbf{C}x(k) \tag{3}$$

with the states $x \in \Re^{n_\mathrm{x}}$, the input $u \in \Re^{n_\mathrm{u}}$ and the output $y \in \Re^{n_\mathrm{y}}$ by

$$\hat{y}(k) = \Psi\hat{x}(k) + \Upsilon u(k-1) + \Theta\Delta U(k)\,. \tag{4}$$

Thereby Ψ, Υ, Θ and $\Delta U(k)$ are in the notation as presented by Maciejowski (2002). With the reference signal $w(k+i|k)$, $i = 1,\ldots,n_\mathrm{y}H_\mathrm{p}$, the control difference over the prediction horizon

$$\varepsilon(k) = \begin{bmatrix} w(k+1|k) \\ w(k+2|k) \\ \vdots \\ w(k+H_p|k) \end{bmatrix} - \Psi\hat{x}(k) - \Upsilon u(k-1) \tag{5}$$

leads to the cost functional

$$J(k) = \Delta U(k)^\mathrm{T} H \Delta U(k) - g(k)^\mathrm{T}\Delta U(k) \tag{6}$$

with

$$H = \Theta^\mathrm{T}\mathcal{Q}\Theta + \mathcal{R} \;\text{ and }\; g(k) = 2\Theta^\mathrm{T}\mathcal{Q}\varepsilon(k)\,. \tag{7}$$

By means of the diagonal matrices $\mathcal{Q} \geq 0$ und $\mathcal{R} > 0$, which elements consist of $Q(i)$ with $i = 1,\ldots,H_\mathrm{p}$ and $R(i)$ with $i = 0,\ldots,H_\mathrm{u}-1$ respectively, the resulting control is parametrised.

Additionally constraints on the actuating variables are defined by the linear matrix inequality

$$A\Delta U(k) \leq b(k)\,. \tag{8}$$

The minimization of the cost function (6) subject to the constraints (8)

$$\min_{\Delta U(k)} \left\{\Delta U(k)^\mathrm{T} H\Delta U(k) - g(k)^\mathrm{T}\Delta U(k) \;:\; A\Delta U(k) \leq b(k)\right\} \tag{9}$$

defines a mathematical standard problem that can be solved by quadratic programming (QP). A variety of methods for solving the QP are commonly used. For an overview we refer to Nocedal and Wright (2006). The basis of the herein used algorithm is the active-set method described by Fletcher (1981), but is optimised for the solution of the QP within the MPC algorithm. Therefore several enhancements are applied like

tuning for the specific structure of the constraint The applied algorithm only accounts for constraints on the actuating variables. Therefore the matrix product $A\Delta U(k)$ equals an accumulated sum of the elements of $\Delta U(k)$ that belongs to the same actuating signal. Therefore a decreases number of operations in necessary.

restriction to a certain process class The actual implementation requires a certain process class that occurs in engine control as shown by Fritzsche et al. (2009). This specialisation allows for the on-line adjustment of the process model with a reasonable demand on processing time.

- **enhanced warm-starting feature** The usage of the solution of the QP at the previous time instant for initialisation often decreases the processing time significantly. In case of reference changes or large disturbances this approach can have the opposite effect. An enhanced initialisation routine based on the procedure by Hildreth (1957) is capable of avoiding such conditions (see Behrendt et al. (2010)).
- **pre-computing the system of equations** In each iteration of the active-set method a system of equations needs to get solved that varies in size depending on the number of active constraints. The application of a LDL^{T}-decomposition allows to pre-compute this system of equations to some extent. Behrendt (2009) has shown that this leads to a significantly lowered numerical complexity for the online iterations.
- **multiple activation and deactivation of constraints** The active constraints in the solution are identified sequentially in the active-set method. Because of the independence of $\Delta U(k)$ terms in (8) that are associated to different actuating signals a concurrent activation of constraints is viable. This often leads to a decreased number of iterations and improves the result in case of an early interruption of the algorithm.
- **exploit the influence of active constraints** Active constraints correspond to optimisation variables that are fixed. Hence, the computed step towards the minimum of (6) is equal to zero. Due to this a priori knowledge the calculation of the step is avoided and a decreased number of operations has to be performed.

These enhancements limit the region of attraction, but allow for real-time control of a certain process class.

3. A SPECIAL CLASS OF SYSTEMS FOR ENGINE CONTROL

This paper deals with a special class of systems that consists of a number of main control variables Y_i, $i = 1, \dots, m$ (e.g. torque and engine speed), a main actuating variable U_1 (e.g. torque by air path) and a number of auxiliary actuating variables U_i, $i = 2, \dots, n$ (e.g. torque by ignition path, exhaust gas recirculation and electric engine). The general system can be described by the discrete-time transfer function

$$\begin{bmatrix} Y_1(z) \\ \vdots \\ Y_m(z) \\ A_1(z) \\ \vdots \\ A_n(z) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) & \cdots & G_{1n}(z) \\ \vdots & \ddots & \ddots & \vdots \\ G_{m1}(z) & G_{m2}(z) & \cdots & G_{mn}(z) \\ 0 & G_{\mathrm{aux}}(z) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{\mathrm{aux}}(z) \end{bmatrix} \begin{bmatrix} U_1(z) \\ U_2(z) \\ \vdots \\ U_n(z) \end{bmatrix} . \tag{10}$$

The purpose of the auxiliary variables is to dynamically support the main actuating variable, but return to their references in the steady state. The transfer function G_{aux} in (10) represents the dynamic of the auxiliary variable realisation. In case, that the auxiliary variable is freely alterable within the constraints in every engine cycle the transfer function

$$G_{\mathrm{aux}}(z) = z^{-1} . \tag{11}$$

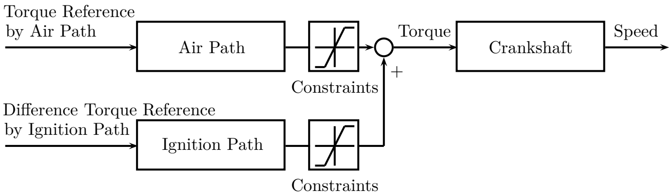


Fig. 1. Considered process structure

Therefore by controlling the variables $A_i, i = 1, \dots, n$ the auxiliary variables are controlled to a prescribed reference (e.g. optimal ignition angle) provided by the combustion process supervision. The relation between the main control variables and the main actuating variable described by $G_{i1}(z)$, $i = 1, \dots, m$ is of slower dynamic in comparison to the remaining sub-systems. This encourages the evident active aid by the auxiliary variables for controlling the plant.

The considered process structure for the specific purpose of this paper is shown in Fig. 1. Two actuating variables have been chosen for engine control. Namely the torque by the air path (AP), which determines the amount of air in the cylinder by altering the throttle opening angle. The amount of air directly affects the resulting torque at the crankshaft. The difference torque by the ignition path (IP), which determines the ignition angle, serves as auxiliary variable and allows for an efficiency deterioration. Thus, it is possible to establish a torque reserve by means of the ignition angle. Because the time constant of the ignition path is significantly smaller than that of the air path a faster response to reference changes or disturbances is available. Additionally both actuating signals are subject to constraints which are inherently respected by MPC. These constraints result from safety reasons in case of the air path, because an exceeding torque could damage the engine. The ignition path is constrained to prevent knocking in the cylinder. For an explanation of this term we refer to Gupta (2006).

The control variables are chosen to be the generated torque at the crankshaft T and the resulting engine speed n. The resulting model can be written as

$$\begin{bmatrix} T(z) \\ n(z) \\ \Delta T\_IP(z) \end{bmatrix} = \begin{bmatrix} G_{T\_AP}(z) & G_{T\_IP}(z) \\ G_{n\_AP}(z) & G_{n\_IP}(z) \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} T\_AP(z) \\ \Delta T\_IP(z) \end{bmatrix} . \tag{12}$$

Naturally the torque and speed cannot be controlled independently. The controller may take only one control variable at a time into account. This can be achieved by not considering the control offset of the particular process output by setting the associated element in the weight of the control deviation q to zero. Otherwise the MPC would attempt to track both references that would lead to an offset on both control signals.

Hence, a single controller structure grants to track different references in dependency on the choice of one parameter. In the field of automotive control this would allow to join the torque controller that tracks the commands of the driver by the gas pedal and the idle speed controller. This approach would significantly simplify the control structure within nowadays ECUs.

Table 1. Control parameters and implementation details

Parameter	Value
prediction horizon H_p	20
control horizon H_u	3
control deviation weight q	[0.1 0.004 0.05]
control action weight r	[10 1]
min. control variable	[10 -10]
max. control variable	[55 0]
processor clock rate	150 MHz
code memory (without FP libs)	6.3 KB
code memory (with FP libs)	13.2 KB
data memory	2.0 KB

It should be mentioned again that we are not going to influence the throttle or the advance angle directly. Instead the MPC is used as superordinate controller that supplies optimal references for the air and ignition path. These references are tracked by the subsidiary control structures that are already included in the ECU. This approach leads to some extent to a linearisation of the underlying structures and therefore a linear MPC can be applied.

4. RESULTS

The results presented in this section are based on the model (12) derived in the previous section. The parameters are obtained by experimental identification at a Volkswagen T5 transporter with 2l TFSI engine.

4.1 Hardware-In-The-Loop Simulation

By means of a Hardware-In-The-Loop (HiL) simulation on the target platform Tricore TC1796 by Infineon Technologies AG (2010) we will show the performance of the control and the algorithm. The TC1796 is a high-potential microcontroller that meets the demands of nowadays engine control. It incorporates an effective floating point unit (FPU), a digital signal processor (DSP) with fixed point arithmetic and a peripheral controller (PCP).

The experimentally chosen control parameters for MPC are summarised in Table 1 and the resulting control is shown in Fig. 2. Until $t = 5\,\mathrm{s}$ the torque is controlled to follow the reference. Therefore the weight on the speed control deviation q_2 is set to zero. At $t = 5\,\mathrm{s}$ the control objective changes by setting the weight on the speed control deviation according to Table 1, but the weight on the torque control deviation q_1 is set to zero. This modifies the MPC to track the speed reference until the end of the simulation.

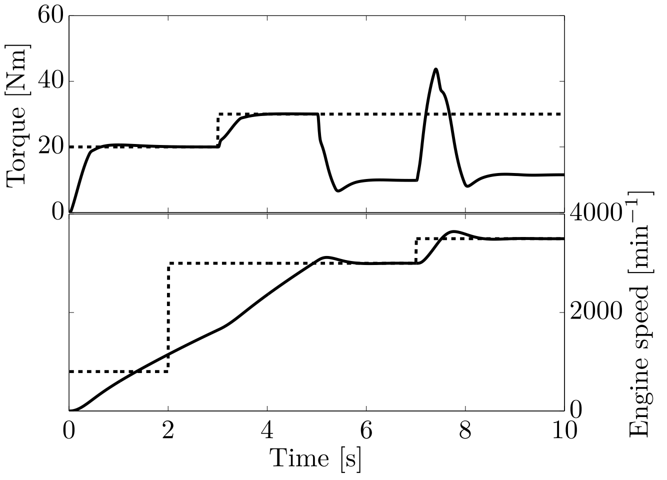


Fig. 2. Control signals (solid) and the references (dashed)

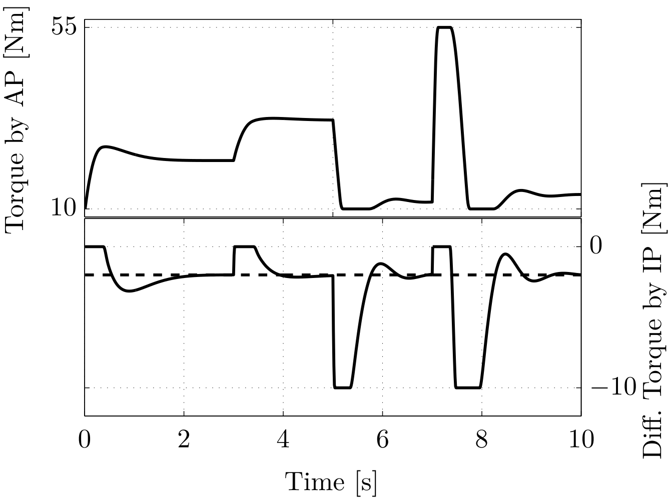


Fig. 3. Constrained actuating signals (solid) and the difference torque reference by IP (dashed)

The difference torque by the ignition path (IP) actively supports the torque by the air path (AP) to reduce the control deviation in the transient phases as shown in Fig. 3. As the control deviation tends to zero the difference torque by IP returns to its reference of $-2\,\mathrm{Nm}$. This realises a torque reserve to allow for quickly reaction on reference changes and disturbances. Nevertheless the constraints on the actuating signals according to Table 1 are respected at any time.

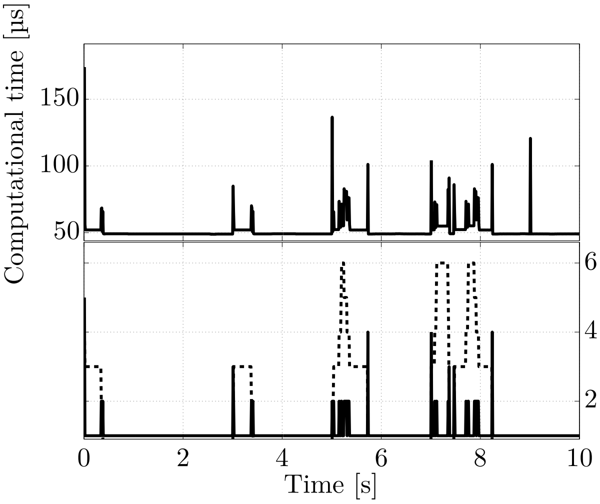


Fig. 4. The computational time for calculating the appropriate actuating signals is shown in the upper graph and the lower graph shows the number of iterations of the active-set algorithm (solid) and the number of active constraints (dashed)

According to the intended operation in an ECU the sample time is set to 10 ms, because that is the fundamental sample time for the air path functionalities. Fig. 4 reveals that the necessary computational time is sufficient for real-time computation of the MPC. The maximum execution time is approximately 180 µs at the beginning of the simulation that leaves enough resources for the remaining functions in the ECU. At the time $t = 5$ s the control objective is adapted by changing the weights and at time $t = 9$ s the plant parameters are slightly changed to demonstrate the effect on the computational time. Both changes cause the re-computation of parameters of the QP, but show moderately increased computational time. The lower graph of Fig. 4 shows the benefit of warm-starting the optimisation algorithm with the solution of the previous time instant. The number of iterations equals the number of necessary changes to the identified active constraints, but not to the number of active constraints itself.

4.2 Evaluation in a Volkswagen T5 transporter

The internal bypass concept by Accurate Technologies (2011) facilitates the easy insertion of additional functionality into existing software states without changing software source code. This allows for expanding an ECU for engine control by the MPC algorithm. The algorithm bypasses the ECU functions that would calculate the torque by AP and the torque by IP in a production ECU. By means of this technique the evaluation of the MPC on the intended hardware platform provides an insight into the applicability of the entire approach.

The first measurements occurred in the idle running of the engine. The Fig. 5 shows the controlled torque (green) and the torque reference (red) that is subject to changes. In case of a reference change the difference torque by IP (magenta) supports the torque by AP (blue) to reduce the control deviation e.g. at the time $t = 452$ s, but does not exceed the minimal difference torque of -10 Nm. In the steady state it returns to its reference (cyan) again. The changes to the difference torque reference by IP are stationary compensated by means of the torque by AP and show minor influence on the controlled torque itself e.g. at the time $t = 477$ s.

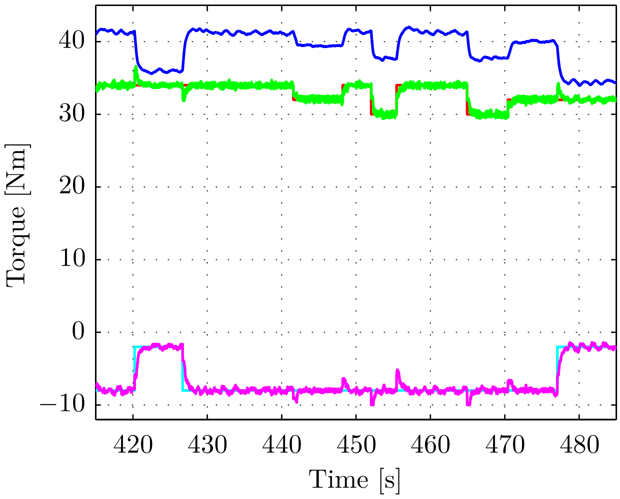


Fig. 5. Torque control at the vehicle in idle running subject to changes of the difference torque reference by IP

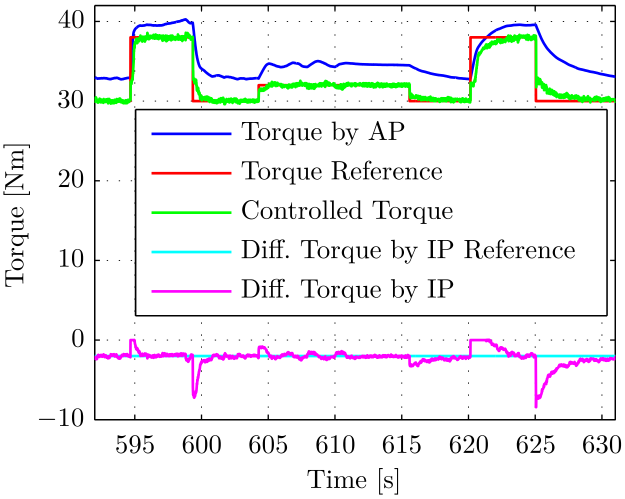


Fig. 6. Torque control at the vehicle in idle running subject to changes of control action weight of torque by AP

The applied MPC algorithm provides the opportunity to change the control parameters during operation. In Fig. 6 an adjustment of the control action weight for the torque by AP takes place. At the time $t = 594$ s the AP is manipulated dynamically to quickly reduce the control deviation due to the reference change. The difference torque by IP does not exceed the maximal difference torque of 0 Nm and immediately returns to its reference. In contrast at the time $t = 620$ s the weight is adapted to moderately utilise the AP. Consequently the torque control deviation and the deviation of the difference torque by IP persists for a longer period of time. This can be interpreted as a sport and economy mode respectively as it is available in nowadays vehicles.

Fig. 7 reveals the possibility of changing the control objective. Until the time $t = 128$ s the torque (green) is

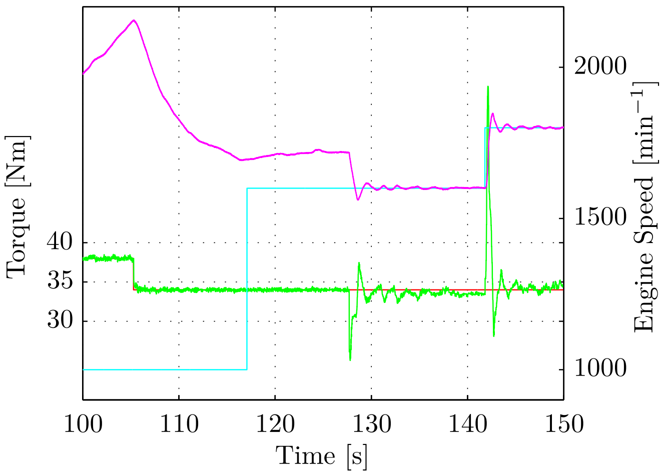


Fig. 7. Online change of the control objective from torque control to engine speed control at the vehicle in idle running

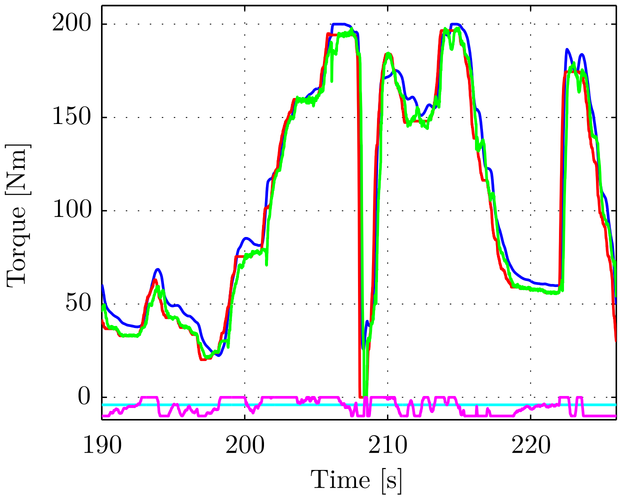


Fig. 8. Torque control at the vehicle on a test track

controlled to its reference (red). Afterwards the control objective is adapted to force the engine speed (magenta) to its reference (cyan).

The evaluation of the MPC on a test track proves the ability to dynamically control a driving scenario. The torque reference in Fig. 8 is shaped by the driver by means of the gas pedal. The actuating variables are manipulated accordingly. Because of seldom steady state phases the difference torque by IP almost always differs from its reference, but never exceeds the constraints. The figure shows, that the torque by AP also remains below the safety constraint of 200 Nm. The vehicle accelerated from 20 km h^{-1} to 66 km h^{-1} and at the time $t = 209$ s the gear is changed. Nevertheless, an adaptation of the process model is not necessary to cope with the changes in the process behaviour.

5. CONCLUSION

While the attractive features of MPC are especially interesting in the automotive industry for e.g. engine control, the numerical complexity of the incorporated optimisation is perceived to prevent its application in embedded systems like production ECUs. This concern has motivated this paper and it could be shown that model predictive engine control is feasible by utilising a specifically tailored algorithm and a limitation to two actuators. A worst case timing of approximately 180 µs during a sample control scenario by using a common micro-controller leaves the necessary resources for the remaining functions in the ECU. Further studies have unveiled that a worst case timing of 800 µs results if four actuators are utilised. That could be recognised of being acceptable as well.

The evaluation in the vehicle has pointed the practicability out. The actuating signals are manipulated in an efficient manner and the ability to control the torque and the engine speed by a single MPC could significantly simplify the existing control structure in nowadays ECUs. Nevertheless, the approach is not yet usable in production units. More work is necessary to account for e.g. overrun fuel cut-off, cylinder shut-off and the integration in the existing structure.

ACKNOWLEDGEMENT

The authors would like to thank Malte Köller and Christian Steinbrecher at IAV GmbH for the technical and organisational support. Their insights in the ECU and valuable help were crucial to the successful evaluation in the vehicle.

REFERENCES

Accurate Technologies (2011). No-Hooks Software. URL `http://www.accuratetechnologies.com/en/products/no-hooks-software-246-243.html`.

Arce, A., del Real, A.J., Bordons, C., and Ramirez, D.R. (2009). Real-Time Implementation of a Constrained MPC for Efficient Airflow Control in a PEM Fuel Cell. IEEE Transactions on Industrial Electronics.

Aspen Technology, Inc. (2010). aspenONE - Advanced Process Control. http://www.aspentech.com.

Baric, M., Baotic, M., and Morari, M. (2005). On-line tuning of the explicit solution to model predictive control. Technical Report AUT05-10, ETH Zurich, Automatic Control Laboratory. URL `http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=2218`.

Behrendt, S. (2009). *Echtzeitfähige Modellprädiktive Regelung für Verbrennungskraftmaschinen.* Master's thesis, Hochschule Wismar.

Behrendt, S., Dünow, P., and Lampe, B. (2010). Simulation study of parallel model predictive control. In *Proceeding of 7th EUROSIM Congress on Modelling and Simulation*. Prague, Czech Republic.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. In *Automatica*, volume 38, 3 – 20.

Camancho, E. and Bordons, C. (2004). *Model Predictive Control.* Springer.

del Re, L., Allgöwer, F., Glielmo, L., and Guardiola, C. (2010). *Automotive Model Predictive Control.* Springer.

Fletcher, R. (1981). *Practical Methods of Optimization*, volume 2: Constrained Optimization. John Wiley & Sons.

Fritzsche, C., Dünow, P., Behrendt, S., Seemann, P., Schnaubelt, M., and Schultalbers, M. (2009). Predictive speed and torque control. In *Proceedings of 7. Symposium "Steuerungssysteme für den Antriebsstrang"*. Berlin, Germany.

Gondzio, J. and Grothey, A. (2006). *Computational Finance and its Applications II*, chapter Solving Nonlinear Financial Planning Problems with 10^9 Decision Variables on Massively Parallel Architectures. WIT Press.

Gondzio, J. and Grothey, A. (2007). Parallel interior point solver for structured quadratic programs: Application to financial planning problems. In *Annals of Operations Research*, volume 152, 319 – 339.

Gupta, H. (2006). *Fundamentals of Internal Combustion Engines.* Prentice Hall.

Hildreth, C. (1957). A quadratic programming procedure. In *Naval Research Logistics Quarterly*, volume 4, 79 – 85.

Infineon Technologies AG (2010). TC1796 (AudoNextGeneration). http://www.infineon.com.

Knagge, G., Wills, A., Mills, A., and Ninness, B. (2009). ASIC and FPGA Implementation Strategies for Model

Predictive Control. In *European Control Conference (ECC)*.

Ling, K., Yue, S., and Maciejowski, J. (2006). A FPGA implementation of model predictive control. In *Proceedings of the American Control Conference*.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Pearson Education Limited.

Naus, G., van den Bleek, R., Ploeg, J., Scheepers, B., van de Molengraft, R., and Steinbuch, M. (2008). Explicit MPC design and performance evaluation of an ACC Step-&-Go. In *Proceedings of the 2008 American Control Conference*.

Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer, second edition.

Ortner, P., Langthaler, P., Ortiz, J.V.G., and del Re, L. (2006). MPC for a Diesel Engine Air Path using an Explicit Approach for Constraint Systems. In *Proceedings of the 2006 IEEE International Conference on Control Applications*.

Pannocchia, G., Rawlings, J.B., and Wright, S.J. (2006). The partial enumeration method for model predictive control: Algorithm and examples. Technical Report 1, TWMCC – Texas-Wisconsin Modeling and Control Consortium.

Rossiter, J. and Grieder, P. (2004). Using interpolation to simplify explicit model predictive control. In *Proceeding of the 2004 American Control Conference*, 885 – 890.

Ruano, A. and Daniel, H. (1997). Parallel implementation of an adaptive generalized predictive control algorithm. In *European Control Conference*. Brussels.

Saerens, B., Diehl, M., Swevers, J., and den Bulck, E.V. (2008). Model Predictive Control of Automotive Powertrains - First Experimental Results. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 5692 –5697.

Siemens AG (2008). SIMATIC PCS 7 APC-Portfolio. http://www.automation.siemens.com.

Texas Instruments Inc. (2010). TMS320C647x Multicore DSPs. http://www.ti.com.

Wills, A., Bates, D., Fleming, A., Ninness, B., and Moheimani, S.R. (2008). Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1), 3–12.

XMOS Ltd. (2010). XS1-G4: 4-core processor. http://www.xmos.com.

Hybrid methods for traffic lights control

Miroslav Makys*. Stefan Kozak**

** Slovak University of Technology, Faculty of Informatics and Information Technologies,
Ilkovicova 3, 84216 Bratislava, Slovakia (e-mail: makys@fiit.stuba.sk)
** Slovak University of Technology, Faculty of Electrical Engineering and Information Technology,
Ilkovicova 3, 84216 Bratislava, Slovakia (e-mail: stefan.kozak@ stuba.sk)*

Abstract: The paper deals with modeling and design of hybrid control using HYSDEL modeling language and MPT toolbox for MATLAB, respectively. We created optimal structure model using hybrid systems theory and designed predictive control of traffic lights of three interconnected intersections. Considered intersections are in detail described in proposed article. A model has been built describing the evolution of the queue based on number of incoming and outgoing cars and traffic lights. A realistic intersections model has been built and achieving optimal traffic lights control was successfully verified on many model variations.

1. INTRODUCTION

During last tenths of year traffic in towns has become serious problem and traffic jam everyday occurrence. As traffic lights are tool to control traffic we need to control the lights optimally. Many authors dealt this problem and various approaches were examined. For example in (Zhang, 2008) traffic light control system was considered to be hybrid system and hybrid Petri nets were used to examine this problem. Timed coloured Petri nets were used in (Huang, 2010) and in (Schutter, 1998) the problem was solved as minimization of special performance function.

In this paper we created model of intersection as a hybrid system and use hybrid systems control theory to control traffic lights. The main aim is to design effective and fast control of traffic lights of three interconnected intersections in optimal way.

The paper has the following structure. In the second section examined system structure is described and simplified yet precise model is derived. The section three deals with control problem definition and obtained results and finally in section four is conclusion of this paper.

2. DESCRIPTION AND MODEL OF INTERSECTIONS

2.1 Description of intersections

In this study, we work with three interconnected intersections. It means that cars outgoing from one of arms of intersection 1 are coming to arms of intersections 2 and 3 and vice versa. Scheme of this system is depicted in Figure 1. First intersection depicted in Figure. 2. consists of 4 two-way arms labelled *A, B, C, D* second intersection depicted in Figure 3. consists of 3 two-way arms labelled *E, F, G* and third intersection depicted in Figure 4. consists of 3 two-way arm labelled *H, I, J.*

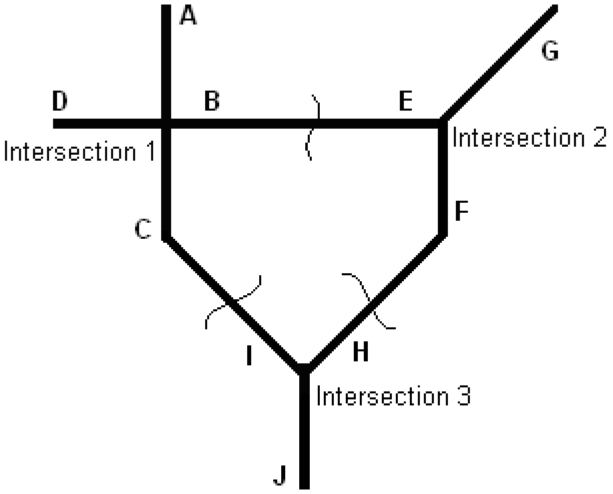


Fig. 1. System of interconnected intersections

Now we are going to describe first intersection. Streams of cars entering the intersection are labelled *A1, B1, B2, C1, C2, D1, D2* and controlled by traffic lights *SA1, SB1, SB2, SC1, SC2, SD1, SD2*, respectively. Each of traffic lights has 3 phases: green, amber and red.

Intersection is depicted in Figure 2. from which it is possible to identify which directions are allowed for cars entering the intersection. Cars in *A1*, *C1* and *D1* streams can only drive straight, cars in *B2*, *C2* and *D2* streams can drive to the right and from B1 stream straight and left.

The amount of cars coming into the intersection for each of the directions is denoted as λ_i, where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$. Function λ_i is formed by a series of Dirac pulses, and for each of the stream is different. One Dirac pulse represents coming of one car. When the traffic light is green or amber, amount of outgoing cars for each of the streams is denoted as μ_i or κ_i, respectively. Functions μi and κi are also formed by a series of Dirac pulses.

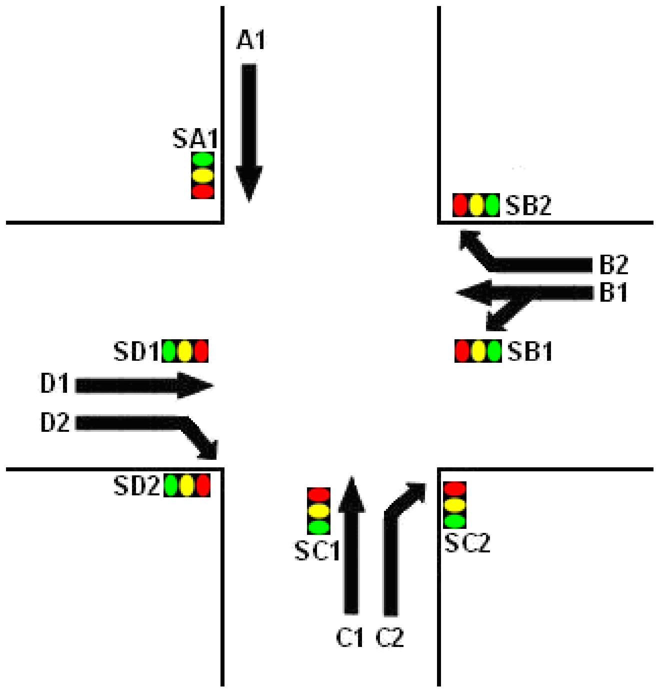


Fig. 2. Scheme of intersection 1

Intersection 2 which is depicted in Figure 3. can be described accordingly to intersection 1. Streams of intersection 2 are labelled *E1, E2, F1, F2, G1* and *G2* and are controlled by traffic lights *SE1, SE2, SF1, SF2, SG1* and *SG2* respectively.

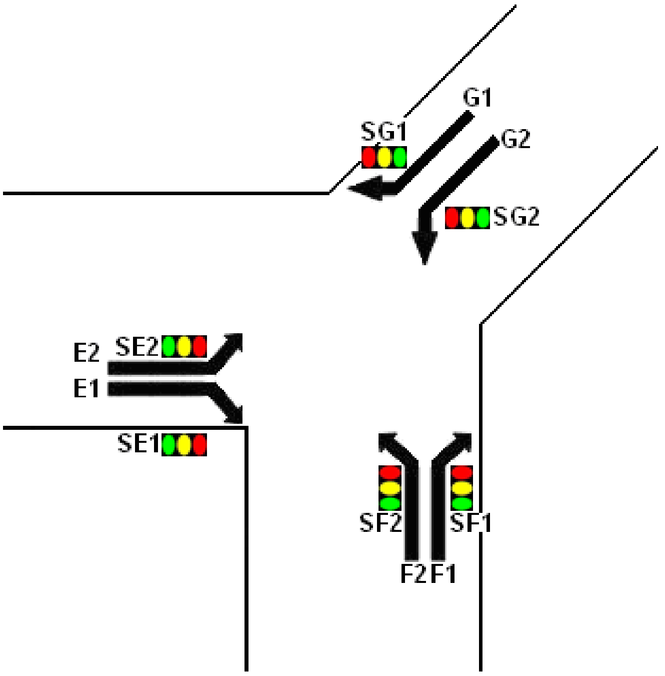


Fig. 3. Scheme of intersection 2

Intersection 3 is similar to intersection 2. Streams of intersection 3 are labelled *H1, H2, I1, I2, J1* ans *J2* and are controlled by traffic lights *SH1, SH2, SI1, SI2, SJ1* ans *SJ2* respectively.

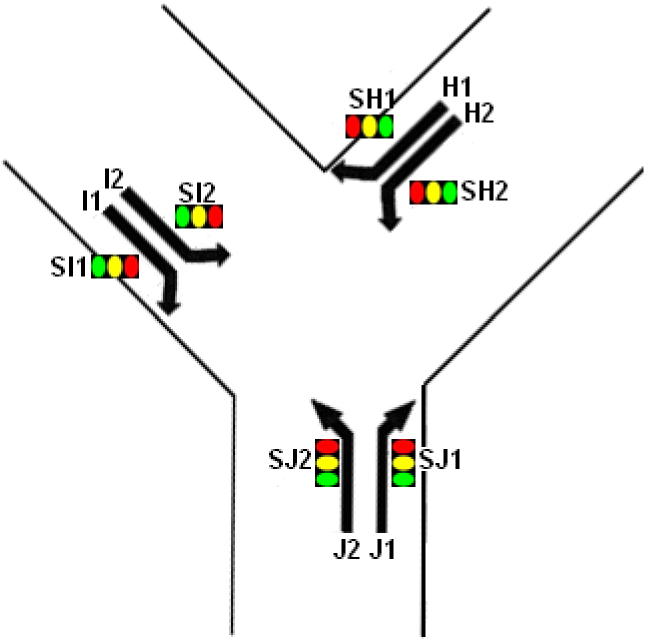


Fig. 4. Scheme of intersection 3

Cars outgoing from arm *B* of intersection 1 are coming to arm *E* of intersection 2, cars outgoing from arm *C* or intersection 1 are coming to arm *I* of intersection 3, cars outgoing from arm *F* of intersection 1 are coming to arm *H* of intersection 3 and wice versa. Time needed car to come from one intersection to another is 15 seconds.

2.2 *Model of intersections*

Proposal of intersection model was based on ideas in (Schutter, 1998). We again describe just model of intersection 1, models is of intersection 2 and intersection 3 were done accordingly. Let us denote length of waiting car queues as L_i, where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$. When traffic light *Si* is red cars are just coming to intersection when it is green or amber cars are coming and outgoing. Difference of queue length is determined by equation (1):

$$\frac{dL_i}{dt} = \begin{cases} \lambda_i(t) & \text{if Si ="red"} \\ \lambda_i(t) - \mu_i(t) & \text{if Si ="}green\text{"} \\ \lambda_i(t) - \kappa_i(t) & \text{if Si ="amber"} \end{cases} \qquad (1)$$

where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$.

Stream 1	Stream 2
A1	D1
A1	D2
A1	B1
D1	C1
D1	C2
D1	B1
D2	B1
C1	B1

Table 1. List of stream pairs which can not enter intersection 1 at the same time

For a realistic model of intersection it is important to define a constraint to the queue length: $Li \geq 0$. Due to avoid collisions in the intersection it is necessary to impose restrictions on the concurrent green color for determined pairs of traffic lights. These were determined on the basis of intersection specification. The list is in Table 1. Since the B2 stream is not in conflict with any other stream, it is not necessary to control it and we miss it.

Similar restrictions related to intersection 2 are listed in Table 2, restrictions for intersection 3 are listed in Table 3.

Stream 1	Stream 2
E1	G2
E2	F1
E2	F2
E2	G2
F2	G1
F2	G2

Table 2. List of stream pairs which can not enter intersection 2 at the same time

Stream 1	Stream 2
H1	J2
H2	I1
H2	I2
H2	J2
I2	J1
I2	J2

Table 3. List of stream pairs which can not enter intersection 3 at the same time

2.3 *Simplified intersection model*

Previous intersection model is too complicated for mathematical analysis, therefore, in this section we simplify the model so that it is easier to work with it while still precise enough. It includes following changes:

- lengths of queues are continuous variables,
-comings and outgoings of cars from intersection are represented by constant function,
-amber phase is missing.

Working with intersection 1 let us denote amount of incoming cars as π_i and amount of outgoing cars as τ_i for each stream where $i \in \{A1, B1, B2, C1, C2, D1, D2\}$.

Difference of queue length is determined by equation (2):

$$\frac{dL_i}{dt} = \begin{cases} \pi_i(t) & \text{if Si} = \text{"}red\text{"} \\ \pi_i(t) - \tau_i(t) & \text{if Si} = \text{"}green\text{"} \end{cases} \qquad (2)$$

Intersection model is thus simplified, so that when traffic light is green, cars are coming and outgoing from intersection in constant rate when traffic lights is red cars are just outgoing in constant rate. Amber phase is omitted. The model was created by HYSDEL modelling language described in (Bemporad, 2004), (Bemporad, 2007) and (Bemporad, 2009) which was designed for modelling of hybrid systems. HYSDEL model is than translated into MLD model using HYSDEL compiler which is one of most used hybrid models. Disadvantage of hybrid models is rapid complexity increase in dependance on number of binary variables which from control point of view results to very difficult and time consuming optimisation problem. Next factor which increases model complexity are constraints on state, input and output variables. That is why we try to create intersection model as simple as possible – with minimum number of binary variables and constraints.

Since the generated model is designed to simulate changes in the length of streams of cars on each of intersection arms, state of system is defined as the number of cars waiting in individual streams thus it is vector of length 6. The basic idea is simple: a queue of cars waiting before the intersection is increasing when the light is red and decreasing when the light is green.

Since the intersection is controlled by 6 traffic lights, intersection model will have 6 input binary control signals one for each of traffic lights. Traffic light can be green if it does not violate the restrictions in Table 1. For example, if *SC1* light is green also *SC2* and *SD2* lights or *SC2* and *SA1* lights can be green. *SB1* and *SD1* lights have to be red. We need not therefore to subject to each of the traffic lights in particular. With this feature it is possible to reduce the number of control signals, thus simplifying the intersection model and hence the problem of control. Our aim is to determine the minimum number of control signals. Task is therefore to determine minimum normal disjunctive form (MNDF) on the basis of Karnaugh map. Table 4 shows Karnaugh map for the intersection 1.

MNDF for given map is:

(!A1 & !B1 & !D1) + (!B1 & !D1 & !D2) + (!A1 & !C1 & !D1 & !D2) + (!A1 & !B1 & !C1 & !C2)

where ! denotes operator of negation.

				D2	D2			D2	D2			D2	D2			D2	D2	
					D1	D1	D1	D1					D1	D1	D1	D1	
							C2	C2	C2	C2	C2	C2	C2	C2				
			C1	C1	C1	C1	C1	C1	C1	C1								
		A1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
B2		A1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1
B2	B1	A1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B1	A1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	B1		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
B2	B1		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
B2			1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1
			1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	1

Table 4. Karnaugh map for intersection 1

It flows from MNDF that to control the intersection we need 4 signals which control the traffic lights following way:

S1 = !A1 & !B1 & !D1 – green light for *B2, C1, C2, D2* streams

S2 = !B1 & !D1 & !D2 – green light for *A1, B2, C1, C2* streams

S3 = !A1 & !C1 & !D1 & !D2 – green light for *B1, B2, C2* streams

S4 = !A1 & !B1 & !C1 & !C2 – green light for *B2, D1, D2* streams

Instead of restrictions listed in the Table 1. we get a new restriction: at most one of the signals can be set to TRUE. Because of maximum intersection throughput we can modify this restriction so that it is: just one of the signals must be set to TRUE.

Let us define vector *X* as system state and vector *U* as system input:

$$\boldsymbol{x}=\begin{bmatrix}L_{A1}\\L_{B1}\\L_{C1}\\L_{C2}\\L_{D1}\\L_{D2}\end{bmatrix},\ \boldsymbol{u}=\begin{bmatrix}S_1\\S_2\\S_3\\S_4\end{bmatrix} \qquad (3)$$

Model of system is then defined by (4):

$$x(k+1)=Ax(k)+Bu(k)+f \qquad (4)$$

where:

$$\boldsymbol{A}=\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix},\ \boldsymbol{f}=\begin{bmatrix}\pi_{A1}\\\pi_{B1}\\\pi_{C1}\\\pi_{C2}\\\pi_{D1}\\\pi_{D2}\end{bmatrix}$$

$$\boldsymbol{B}=\begin{bmatrix}-\tau_{A1}&0&0&0\\0&0&-\tau_{B1}&0\\-\tau_{C1}&-\tau_{C1}&0&0\\-\tau_{C2}&-\tau_{C2}&-\tau_{C2}&0\\0&0&0&-\tau_{D1}\\-\tau_{D2}&0&0&-\tau_{D2}\end{bmatrix}$$

Table 5 shows parameters used to create intersection 1 model. Row *Inc.* contains number of cars incoming to assigned stream per 1 second. Row *Out.* contains number of cars outgoing from assigned stream if traffic lights is green.

	A1	B1	B2	C1	C2	D1	D2
Inc.	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Out.	1.8	2	1.2	2.2	0.8	1.8	1.4

Table 5. List of intersection 1 parameters

Using similar methodology we get following results for intersection 2. Based of restrictions in Table 2, we create Karanugh map shown in Table 6.

MNDF of this map is:

(!E2 & !G1 & !G2) | (!E1 & !E2 & !F2) | (!F1 & !F2 & !G2) | (!E2 & !F2 & !G2)

Based on MNDF we have 4 control signals:

S5 = !E2 & !G1 & !G2 – green light for *E1, F1, F2*

S6 = !E1 & !E2 & !F2 – green light for *F1, G1, G2*

S7 = !F1 & !F2 & !G2 – green light for *E1, E2, G1*

S8 = !E2 & !F2 & !G2 – green light for *E1, F1, G1*

And we get similar intersection model.

				F1						
					E2					
			E1							
		F2	1	1	0	0	0	0	1	1
G2			0	0	0	0	0	0	0	0
	G1		0	0	0	0	0	0	0	0
			0	0	0	0	0	0	0	0
			1	1	0	1	1	0	1	1
			0	0	0	0	0	0	1	1
			0	0	0	0	0	0	1	1
			1	1	0	1	1	0	1	1

Table 6. Karnaugh map for intersection 2

Table 7 shows parameters used to create intersection 2 model. Row *Inc.* contains number of cars incoming to assigned stream per 1 second. Row *Out.* contains number of cars outgoing from assigned stream if traffic lights is green.

	E1	E2	F1	F2	G1	G2
Inc.	0.4	0.6	0.4	0.2	0.4	0.4
Out.	1	2	1.2	0.6	1.2	1.8

Table 7. List of intersection 2 parameters

				I1						
					H2					
			H1							
		I2	1	1	0	0	0	0	1	1
J2			0	0	0	0	0	0	0	0
	J1		0	0	0	0	0	0	0	0
			0	0	0	0	0	0	0	0
			1	1	0	1	1	0	1	1
			0	0	0	0	0	0	1	1
			0	0	0	0	0	0	1	1
			1	1	0	1	1	0	1	1

Table 8. Karnaugh map for intersection 3

Using similar approach we get Karnaugh map for intersection 3 shown in Table 8.

MNDF of this map is:

(!H2 & !J1 & !J2) | (!H1 & !H2 & !I2) | (!I1 & !I2 & !J2) | (!H2 & !I2 & !J2)

Based on MNDF we have 4 control signals:

S9 = !H2 & !J1 & !J2 – green light for *H1, I1, I2*

S10 = !H1 & !H2 & !I2 – green light for *I1, J1, J2*

S11 = !I1 & !I2 & !J2 – green light for *H1, H2, J1*

S12 = !H2 & !I2 & !J2 – green light for *H1, I1, J1*

And we get similar intersection model.

Table 9 shows parameters used to create intersection 3 model. Row *Inc.* contains number of cars incoming to assigned stream per 1 second. Row *Out.* contains number of cars outgoing from assigned stream if traffic lights is green.

	H1	H2	I1	I2	J1	J2
Inc.	0.3	0.4	0.6	0.6	0.4	0.2
Out.	1	2	1.2	1	1.2	1.4

Table 9. List of intersection 3 parameters

3. CONTROL DESIGN

Design of control was made using MPT toolbox for Matlab described in (Kvasnica, 2004) and (Kvasnica, 2009). The aim of control is to set traffic lights so that throughput of intersection is maximum, while the cars on less busy streams do not wait too long to get to turn. Theoretical maximum throughput is achieved when all the lights are green, which of course due to collisions is not possible. Restrictions are summarized in Table 1., Table 2. and Table 3.

Aim of approach used in this paper is to minimize the number of cars facing red lights. To avoid permanent switching of traffic lights performance function penalizes also changes in traffic lights. The aim is to minimize performance function (5):

$$J = \sum_{k=1}^{N} \|Q_x x(k)\|_1 - \sum_{k=1}^{N} \|Q_L u\|_1 + \\ + \sum_{k+1}^{N} \|Q_U (u(k-1) - u(k))\|_1 \tag{5}$$

subject to:

$$x(k+1) = Ax(k) + Bu(k) + f$$

$$\|\boldsymbol{u}\|_1 = 1$$

$$\boldsymbol{x}(t+k) \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ for } k \in \{1,2,\ldots,N\}$$

where: N is length of prediction horizon,

QX, QL, QU are penalty matrices,

$$Q_L = \begin{bmatrix} L_{A1} & 0 & 0 & 0 \\ 0 & 0 & L_{B1} & 0 \\ L_{C1} & L_{C1} & 0 & 0 \\ L_{C2} & L_{C2} & L_{C2} & 0 \\ 0 & 0 & 0 & L_{D1} \\ L_{D2} & 0 & 0 & L_{D2} \end{bmatrix}.$$

The sence of $||u||_1=1$ constraint is to ensure that during all control time just one of input singnals is set to TRUE.

Using this approach we get three similar control laws – one for each intersection. Relation between these three systems is presented by “road” which connects two arms of intersections.

During testing of this method we created temporary bigger amount of cars coming to stream *B1* of intersection 1. In resulting graphs we can see that designed control algorithm is able to adapt to changed conditions.

Results of this method are depicted in Figures 5-7 for intersection 1, Figures 8-10 for intersection 2 and Figures 11-13 for intersection 3.

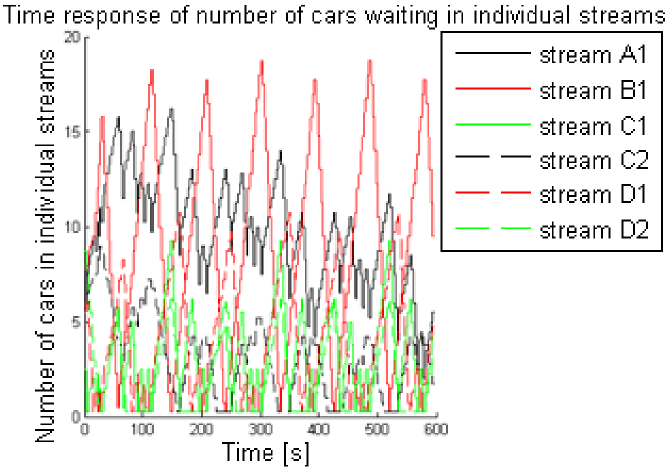


Fig. 5. Time response of number of cars waiting in individual streams in intersection 1

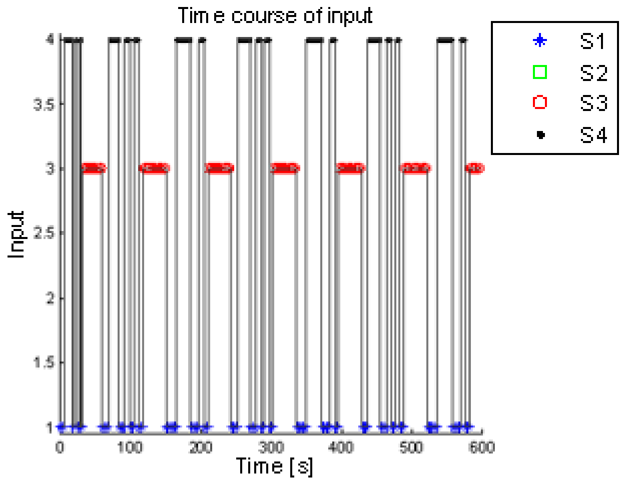


Fig. 6. Input signal S in intersection 1 is set to TRUE when value of function in graph is *i*

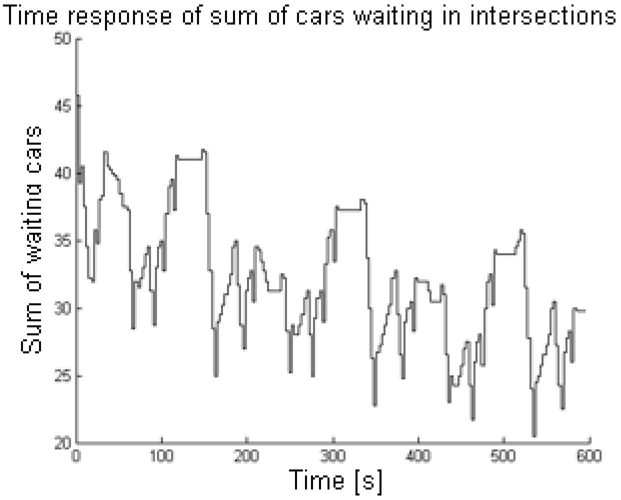


Fig. 7. Time response of total number of cars waiting in streams of intersection 2

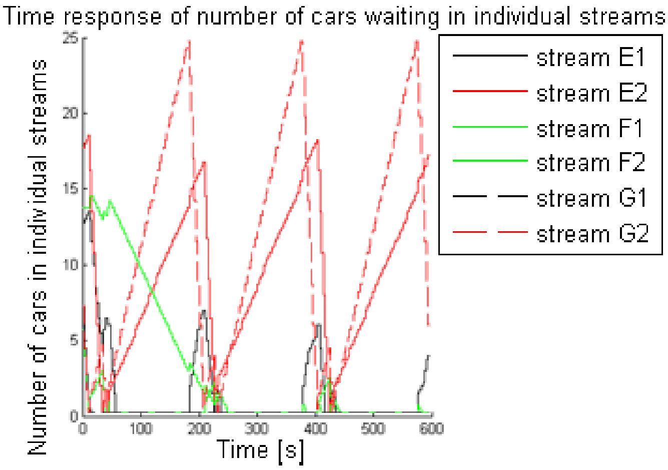


Fig. 8. Time response of number of cars waiting in individual streams in intersection 2

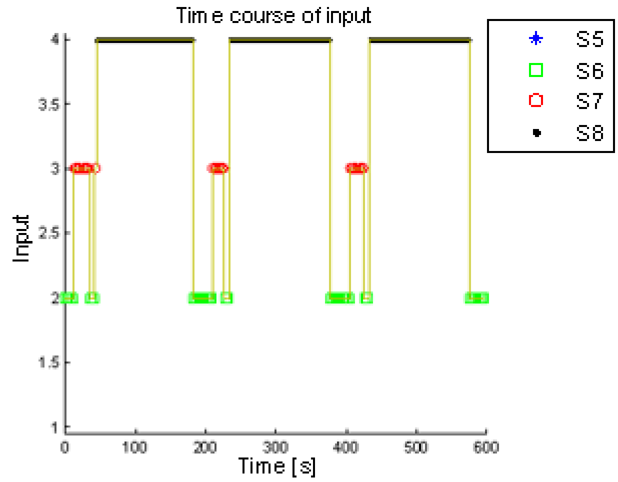


Fig. 9. Input signal S in intersection 2 is set to TRUE when value of function in graph is *i*

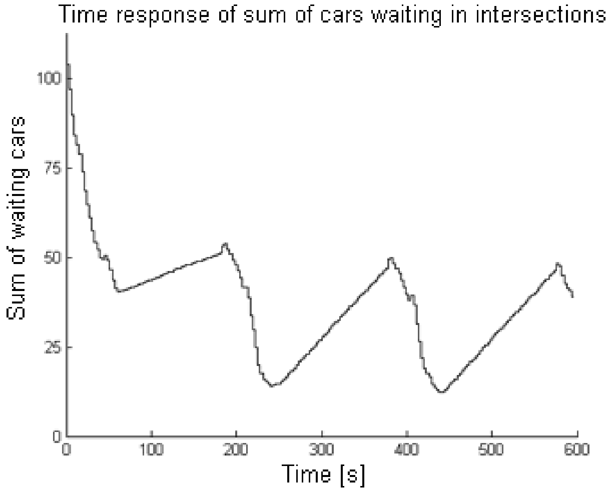


Fig. 10. Time response of total number of cars waiting in streams of intersection 2

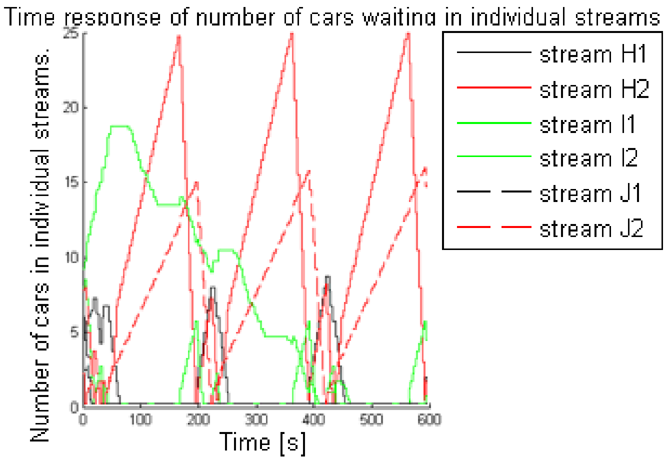


Fig. 11. Time response of number of cars waiting in individual streams in intersection 3

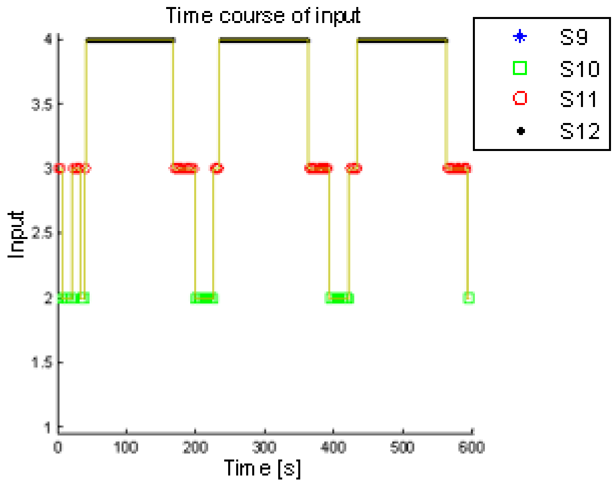


Fig. 12. Input signal S in intersection 3 is set to TRUE when value of function in graph is *i*

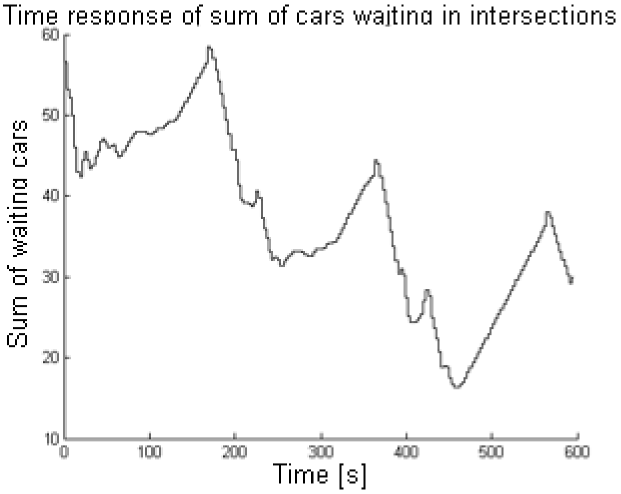


Fig. 13. Time response of total number of cars waiting in streams of intersection 3

4. CONCLUSION

The paper deals with the design of optimal model structure and innovative hybrid predictive control of typical hybrid dynamic real traffic systems. The main result of the paper is proposal of two interconnected intersections control based on hybrid predictive control. The main objective for traffic lights setting is the number of cars faced to red light. This is a “fair” approach because we let pass through cars which are in longest queue.

To get information about other approaches to modelling and control of transport systems we refer reader to (Kvasnica, 2009), (Saez, 2007), (Cortes, 2009), (Zhang, 2008) and (Huang, 2010).

Advantage of this approach is that it computes with actual lengths of car queues so it is able to adapt to changes in amount of cars coming to intersection as it was shown.

Modelling and simulation obtained results proved that the proposed approach is suitable for real intersections.

ACKNOWLEDGEMENTS

This project was supported by grant of VG 1/1105/11 (VEGA) project and APVV project No VMSP-P-0059-09.

REFERENCES

Bemporad. A. (2004). *Hybrid toolbox – user's guide*. http://www.ing.unitn.it/~bemporad/hybrid/toolbox

Bemporad. A. (2007). *Model Predictive Control of Hybrid Systems*. http://www.dii.unisi.it/hybrid/school07/pdf/08.Bemporad.pdf

Bemporad. A. (2009). Modeling and control of hybrid dynamical systems: The Hybrid toolbox for MATLAB. in Proc. MATHMOD Conference, I. Troch and F. Breitenecker, Eds, *ARGESIM Reports*, 82–100. Vienna, Austria

Cortés. C.E. (2009). Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization. *Transportation Research Part C: Emerging Technologies*, 757 – 769.

Huang. Y., Chung. T. (2010). Modelling and analysis of traffic light control systems using timed coloured Petri nets. In Pawel Pawlewski, *Petri Nets Applications*. INTECH

Kvasnica. M., Grieder. P., and Baotic. M. (2004). *Multi-parametric toolbox (MPT).* http://control.ee.ethz.ch/~mpt/

Kvasnica. M. (2009). *Real-time model predictive control via multi-parametric programming: theory and tools*. VDM Verlag, Saarbruecken

Sáez. D. et. Al. (2007). Hybrid predictive control strategy for a public transport system with uncertain demand. In *Proceedings of Sixth Triennial Symposium on Transportation Analysis* , Phuket Island, Thailand

Robust Controller Design for a Laboratory Process with Uncertainties

J. Závacká, M. Bakošová

Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics,
Radlinského 9, 812 37 Bratislava, Slovakia
fax : +421 259 325 340, e-mail: jana.zavacka@stuba.sk

Abstract: The paper presents a method for design of robust PI controllers for systems with interval uncertainty. The method is based on plotting the stability boundary locus in the (k_p, k_i)-plane and sixteen plant theorem. The stability boundaries obtained for interval plants split the (k_p, k_i)-plane in stable and unstable regions. The parameters of robust PI controllers are chosen from the stable region. The designed robust PI controller is used for control of a laboratory chemical continuous stirred tank reactor (CSTR). The reactor is used for preparing of NaCl solution with desired concentration. The conductivity of the solution is the controlled variable and the volumetric flow rate of water is the control variable.

Keywords: robust control, PI controller, interval uncertainty, process control

1. INTRODUCTION

Chemical reactors are ones of the most important plants in chemical industry, see e.g. Mikleš and Fikar (2007). Their operation, however, is corrupted with various uncertainties. Some of them arise from varying or not exactly known parameters, as e.g. reaction rate constants, reaction enthalpies or heat transfer coefficients. In other cases, operating points of reactors vary or reactor dynamics is affected by various changes of parameters of inlet streams. All these uncertainties can cause poor performance or even instability of closed-loop control systems. Application of robust control approach can be one of ways for overcoming all these problems, which may seriously influence control design for chemical reactors and other chemical processes, see e.g. Alvarez-Ramirez and Femat (1999), Gerhard et al. (2004).

In this paper, a simple method for design of robust PI controllers is presented (Tan and Kaya (2003)). The method is based on plotting the stability boundary locus in the (k_p, k_i)-plane and then parameters of a stabilizing PI controller are determined from the stability region. The PI controller stabilizes a controlled system with interval parametric uncertainties, when the stability region is found for sufficient number of Kharitonov plants (Barmish (1994)).

The approach is used for design of a robust PI controller for a laboratory continuous stirred tank reactor, which can be modelled in the form of a transfer function with parametric interval uncertainty. The reactor serves for preparing of the NaCl solution with required concentration. Composition of the solution is determined by measurement of the solution conductivity and the conductivity is the controlled variable. The volumetric flow rate of water which is used for adulterating of NaCl solution, is the control variable.

2. DESCRIPTION OF THE LABORATORY CSTR

Multifunctional process control teaching system - The Armfield PCT40 (Armfield (2005), Vojtešek et al. (2007)) is the system which enables to test a wide class of technological processes, as a tank, a heat exchanger, a continuous stirred tank reactor and their combinations (Armfield (2006a), Armfield (2006b)).

PCT40 unit consists of two process vessels, several pumps, sensors and connection to the computer. Additional equipments PCT41 and PCT42 represent a chemical reactor with a stirrer and a cooling/heating coil.

Inlet streams of reactants can be injected into the reactor via a normally closed solenoid valve or by a proportional solenoid valve (PSV). The third possibility for feeding water into the reactor is using one of two peristaltic pumps. The technological parameters of the reactor are shown in Table 1.

Table 1. Technological parameters of the reactor

Parameter	Value
Vessel diameter	0.153 m
Maximum vessel depth	0.108 m
Maximum operation volume	2 l
Minimum vessel depth	0.054 m
Minimum operation volume	1 l

The connection to the computer is realized via an I/O connector, which is connected to the PCL card. The card used is the MF624 multifunction I/O card from Humusoft. This card has 8 inputs and 8 outputs. The whole system provides 9 inputs and 17 outputs, hence two MF624 cards were used. This connection enables use of Matlab Realtime Toolbox and Simulink or data entry from the Matlab command window.

NaCl solution with the concentration 0.8555 mol/dm^3 is fed into the reactor by a peristatic pump. The performance of the pump may be theoretically set in the range $0-100\%$. But for the pump performance less than 30%, revolutions of the rotor are very small and the produced force is not high enough to transport the fluid from the barrel. The volumetric flow rate of the NaCl solution for all measurements was 0.00175 dm^3/s, which represents the pumpe performance 40%.

The water was dosed into the reactor by the PSV. Application of the PSV allowed flow measurements by the adjoint flowmeter. The PSV opening could be again done in the range $0-100\%$, but the volumetric flow rate of water for the PSV opening in the range $0-30\%$ was negligible.

For control purposes, the laboratory continuous stirred tank reactor is a SISO system. The control variable is the volumetric flow rate of water (F) and the controlled variable is the conductivity of the NaCl solution (G) inside the reactor. Used water was cold water from the standard water distribution. The volume of the solution in the reactor was kept constant with the value 1 dm^3 during all experiments.

3. PROCESS IDENTIFICATION

Identification of the controlled laboratory reactor was done from measured step responses. The constant flow rate 0.00175 dm^3/s of NaCl solution dosed into the reactor was assured by the peristaltic pump with performance 40% in all experiments. Fourteen various step changes of water flow rate were realized between 0.0032 dm^3/s - 0.01145 dm^3/s which represented the PSV opening $50-100\%$. The step responses were measured repeatedly. The resultant transfer function of the laboratory reactor was identified in the form of a transfer function (1) with the parametric interval uncertainty. The software LDDIF (Čirka and Fikar (2007)) was used for identification, which is based on the least squares algorithm. The values of the uncertain parameters are shown in Table 2.

$$G(s) = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} \tag{1}$$

Table 2. Uncertain parameters

Parameter	Minimal value	Maximal value
b$_1$	0.0028	0.0428
b$_0$	-0.2776	-0.0156
a$_2$	1	1
a$_1$	0.6349	5.5024
a$_0$	0.2084	3.1351

4. DESIGN OF A ROBUST PI CONTROLLER

A simple method based on plotting the stability boundary locus in the (k_p, k_i)-plane and the sixteen plant theorem is used for robust PI controller design, Tan and Kaya (2003), Závacká et al. (2008), Barmish (1994). Parameters of a stabilizing PI controller are determined from the stability region of the (k_p, k_i)-plane. The PI controller stabilizes a controlled system with interval parametric uncertainties, when the stability region is found for sufficient number of Kharitonov plants.

For the controlled system in the form of the transfer function (1) with interval uncertainty (Table 2), the Kharitonov polynomials $N_i(s)$, $i = 1,2,3,4$ for the numerator and $D_j(s)$, $j = 1,2,3,4$ for the denominator can be created, as it is seen in (2), (3).

$$\begin{aligned} N_1(s) &= b_1^- s + b_0^- \\ N_2(s) &= b_1^+ s + b_0^+ \\ N_3(s) &= b_1^+ s + b_0^- \\ N_4(s) &= b_1^- s + b_0^+ \end{aligned} \tag{2}$$

$$\begin{aligned} D_1(s) &= a_2^- s^2 + a_1^- s + a_0^+ \\ D_2(s) &= a_2^+ s^2 + a_1^+ s + a_0^- \\ D_3(s) &= a_2^+ s^2 + a_1^- s + a_0^- \\ D_4(s) &= a_2^- s^2 + a_1^+ s + a_0^+ \end{aligned} \tag{3}$$

where b_k^- and b_k^+, $k = 0,1$ are lower and upper bounds of the intervals of the numerator and a_l^- and a_l^+, $l = 0,1,2$, are lower and upper bounds of intervals of the denominator parameters. 16 Kharitonov systems (4) can be obtained using polynomials (2), (3)

$$G_{ij}(s) = \frac{N_i(s)}{D_j(s)} \tag{4}$$

Substituting $s = j\omega$ into (4) and decomposing the numerator and the denominator polynomials of (4) into their even and odd parts one obtains

$$G_{ij}(j\omega) = \frac{N_{ie}(-\omega^2) + j\omega N_{io}(-\omega^2)}{D_{je}(-\omega^2) + j\omega D_{jo}(-\omega^2)} \tag{5}$$

The closed loop characteristic polynomial is as follows

$$\begin{aligned} \Delta(j\omega) &= [k_i N_{ie}(-\omega^2) - k_p \omega^2 N_{io}(-\omega^2) - \\ &\quad -\omega^2 D_{jo}(-\omega^2)] + j[k_p \omega N_{ie}(-\omega^2) + \\ &\quad +k_i \omega N_{io}(-\omega^2) + \omega D_{je}(-\omega^2)] \end{aligned} \tag{6}$$

Then, equating the real and imaginary parts of $\Delta(j\omega)$ to zero, one obtains

$$\begin{aligned} & k_p(-\omega^2 N_{io}(-\omega^2)) + k_i(N_{ie}(-\omega^2)) \\ & = \omega^2 D_{jo}(-\omega^2) \end{aligned} \tag{7}$$

and

$$\begin{aligned} & k_p(N_{ie}(-\omega^2)) + k_i(N_{io}(-\omega^2)) \\ & = -D_{je}(-\omega^2) \end{aligned} \tag{8}$$

After denoting

$$\begin{aligned} F_i(\omega) &= -\omega^2 N_{io}(-\omega^2) \\ G_i(\omega) &= N_{ie}(-\omega^2) \\ H_i(\omega) &= N_{ie}(-\omega^2) \\ I_i(\omega) &= N_{io}(-\omega^2) \\ J_j(\omega) &= \omega^2 D_{jo}(-\omega^2) \\ K_j(\omega) &= -D_{je}(-\omega^2) \end{aligned} \tag{9}$$

(7), (8) and (9) can be written as

$$\begin{aligned} k_pF_i(\omega)+k_iG_i(\omega) &= J_j(\omega)\\ k_pH_i(\omega)+k_iI_i(\omega) &= K_j(\omega) \end{aligned} \qquad (10)$$

From these equations, parameters of the PI controller are expressed in the form

$$k_p = \frac{J_j(\omega)I_i(\omega)-K_j(\omega)G_i(\omega)}{F_i(\omega)I_i(\omega)-G_i(\omega)H_i(\omega)} \qquad (11)$$

and

$$k_i = \frac{K_j(\omega)F_i(\omega)-J_j(\omega)H_i(\omega)}{F_i(\omega)I_i(\omega)-G_i(\omega)H_i(\omega)} \qquad (12)$$

Consider one of the systems (4), where $i = 2$ and $j = 3$

$$G_{23}(s) = \frac{0.0428s-0.0156}{s^2+0.6349s+0.2084} \qquad (13)$$

Then

$$\begin{aligned} k_p &= \frac{a_2^+b_0^+\omega^2-a_0^-b_0^+-a_1^-b_1^+\omega^2}{(b_1^+)^2\omega^2+(b_0^+)^2}\\ k_i &= \frac{a_1^-\omega^2+k_pb_1^+\omega^2}{b_0^+} \end{aligned} \qquad (14)$$

The stability boundary of the closed loop with the system (13) in the (k_p,k_i)-plane for $\omega = [0, 0.6267]$ is plot in the Figure 1. Then parameters k_p and k_i of the stabilizing controller are chosen from the stable region.

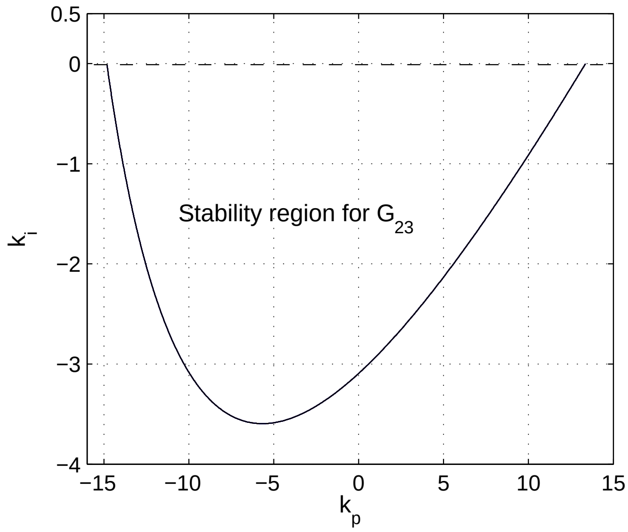


Fig. 1. Stability region of parameters k_p, k_i for the system G_{23}

Stable regions for all 16 Kharitonov systems are obtained alike. In the Figure 2 are shown stable regions for 16 Kharitonov systems (4). The controller which stabilizes all 16 Kharitonov systems has to be found in the intersection of all stable regions (the intersection lies in the red rectangle), which is in detail displayed in the Figure 3.

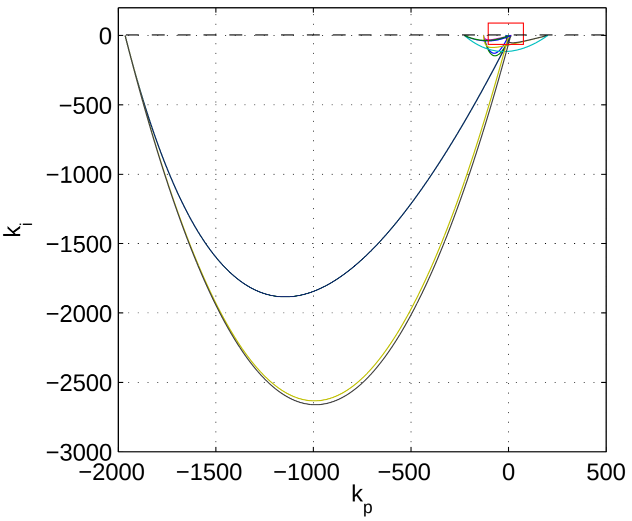


Fig. 2. Stability regions for 16 Kharitonov plants

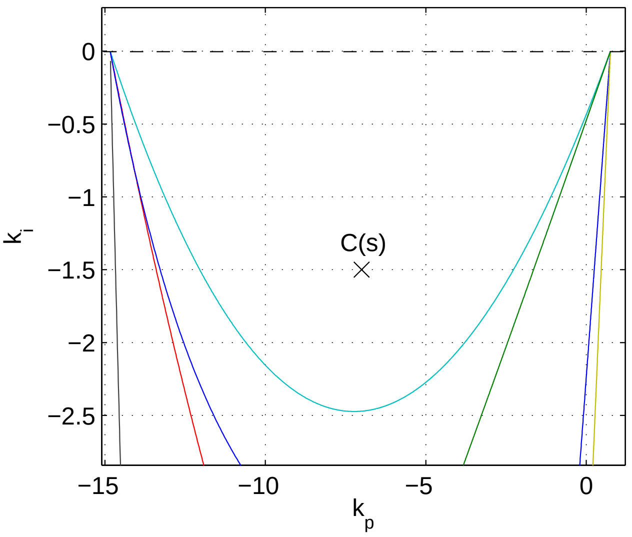


Fig. 3. Detail of the stability region for 16 Kharitonov plants

The parameters of the robust PI controller for control of the laboratory reactor (15) were chosen from the stable region of parameters k_p, k_i according to simulation results obtained for various choices of the PI controllers.

$$C(s) = \frac{k_ps+k_i}{s} = \frac{-7s-1.5}{s} \qquad (15)$$

The designed PI controller was used for control of the laboratory reactor. The controlled variable $y(t)$ was the conductivity G [mS] of the NaCl solution, the control variable $u(t)$ was the water flow rate F [dm^3/s] and the reference $w(t)$ was the conductivity of the NaCl solution which corresponded to the required concentration of the NaCl solution.

Obtained experimental results are presented in the Figures 4 and 5. Robustness of the designed PI controller (15) was tested by setting the reference value in a wider area. Control responses of the reactor are shown in Figure 4 for $w \in [12; 32]$ mS and in the Figure 5 for $w \in [18; 30]$ mS.

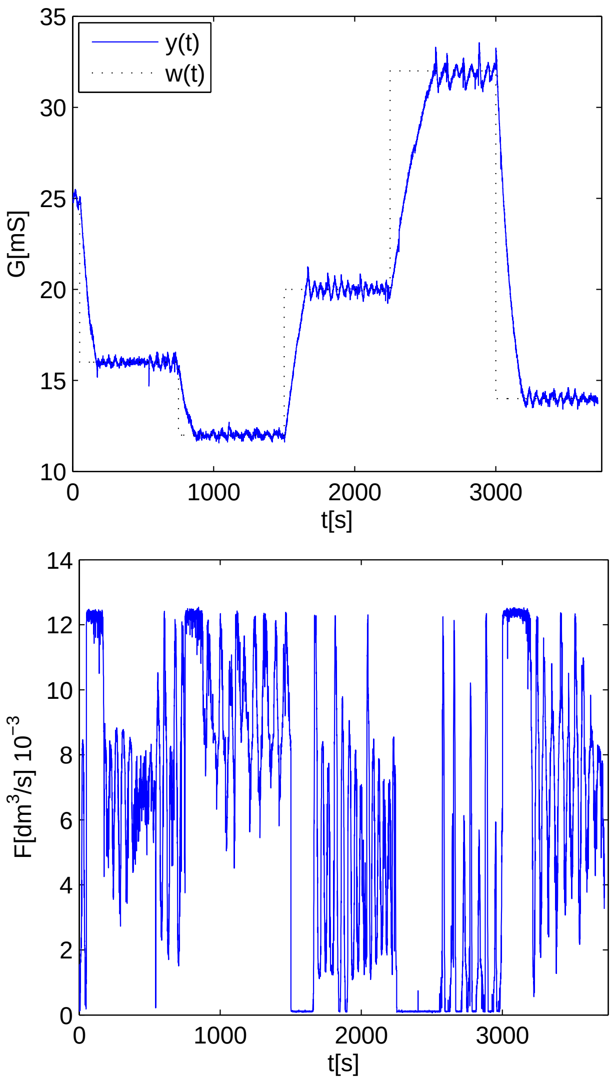


Fig. 4. Control of the reactor with robust PI controller

5. CONCLUSION

The robust PI controller was designed for control of the laboratory continuous stirred tank reactor. A simple robust synthesis was used which was based on plotting the stability boundary locus in the (k_p, k_i)-plane and the sixteen plant theorem. The robust PI controller was chosen from the stable region of the (k_p, k_i)-plane. The designed controller was tested experimentally by control of a laboratory reactor. Obtained experimental results confirmed that the designed robust PI controller successfully controlled the laboratory reactor where controlled variable - conductivity G [mS] of NaCl, was controlled by water flow rate F [dm^3/s]. The varying reference was always reached. The control responses were without overshoots and fast enough.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0537/10, 1/0095/11, and by the Slovak Research and Development Agency under the project APVV-0029-07.

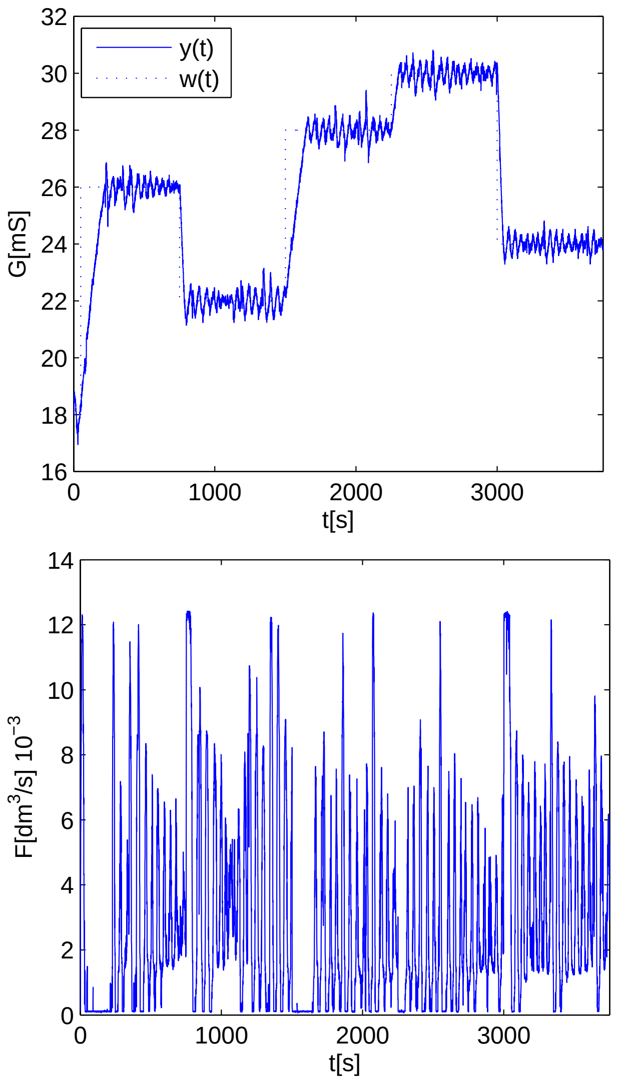


Fig. 5. Control of the reactor with robust PI controller

REFERENCES

Alvarez-Ramirez, J. and Femat, R. (1999). Robust PI stabilization of a class of chemical reactors. *Sytems Control Letter*, 38, 219–225.

Armfield (2005). *Instruction manual PCT40*, 4 edition.

Armfield (2006a). *Instruction manual PCT41*, 3 edition.

Armfield (2006b). *Instruction manual PCT42*, 2 edition.

Barmish, B. (1994). *New Tools for Robustness of Linear Systems*. Macmillan Publishing Company.

Gerhard, J., Monningmann, M., and Marquardt, W. (2004). Robust stable nonlinear control and design of a CSTR in a large operating range. In *Proc. 7th Int. Symp. Dynamics and Control of Process Systems*, Cambridge, Massachusset, USA, July 5 – 7. CD-ROM 92.

Mikleš, J. and Fikar, M. (2007). *Process modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg.

Tan, N. and Kaya, I. (2003). Computation of stabilizing pi controllers for interval systems. In *Medditeranean Conference on Conntrol and Automation*. Rhodes, Greece.

Čirka, Ľ. and Fikar, M. (2007). *A Dynamical System Identification Toolbox, Selected Topics in Modelling and*

Control. Bratislava: STU Press, 58–62.

Vojtešek, J., Dostal, J., and Matušů, R. (2007). Multifunctional process control teaching system pct40. In *16th Int. Conferences Process Control*, 15f.pdf. Štrbské pleso, Slovakia.

Závacká, J., Bakošová, M., and Vaneková, K. (2008). Design of robust pi, pid controllers for control of systems with parametric uncertainty. In *V 18th International Congress of Chemical and Process Engineering*, 0457–1–0457–10. Prague, Czech Republic.

Prediction of critical processes in nuclear power plant using genetically trained neural networks

Matej Petrík [*] **Štefan Kozák** [**]

[*] *Matej Petrík, Institute of Applied Informatics at FIIT SUT (Tel: +421 910 904 516; e-mail: matej.petrik@gmail.com)*
[**] *Štefan Kozák, Institute of Automatic control systems at FEI SUT (Tel: +421 905 581 323; e-mail: stefan.kozak@stuba.sk)*

Abstract: Neural network is one of many models used in power engineering process prediction. In most cases, the accuracy of prediction models is critical in operational safety or is used to support human irreversible decisions. We use neural network, when the real model of process is unknown, or it is too difficult to identify them in domain specific environment. Neural networks training algorithms are different. Typically, measured data are divided into 2 sets called train and test set. On the train set, algorithms set neural network parameters so that network simulate process on train interval. On the test set is network tested if it can generalize the process from train set. We take a look on special genetic training and compare it with algorithm used today to generalize the process.

Keywords: neural network genetic plant energetics

1. INTRODUCTION

Nuclear power plant performance is the result of complex system composed of nuclear reactor, warm and cold water, radioactive water, turbines and electricity generator. Source of generated energy are graphite rods containing uranium. Uranium is fuel for controlled nuclear reactions, which results into non-linear amount of irradiated heat. This heat is transfered through two system of warm water to large generator turbines - figure 11. Even though we know the physical background of the nuclear chain reaction and thermodynamic laws, we are not able to compute physical model because of many (but measurable) external impacts. Instead of assembling mathematical model of physical laws affecting the outputs in nuclear power plant, we are trying to create soft-computing methods to solve the prediction problem. Performance modeling is critical in 2 cases: security and economic. To prevent and avoid critical situation, we can't generate more energy in reactor than the water system can absorb. From economic reasons we have to generate just enough energy, which is possible to consume by customers.

Described process is nonlinear dynamic system, where the current state strongly depends on previous situation and outside conditions. It is possible to model dynamic systems in many ways, but with neural networks we hold a great tool - universal approximator. Offline training (which is specific for neural networks with supervisor) bring us opportunity to research new algorithms without affecting the real world. The only important point on generated neuro-model is generalization quality - the ratio between test and train prediction error. Classical algorithms use during the training process prediction error as error signal to calculate new network parameters. In this work, we are going to describe and show special genetic algorithm for training neural network, and compare them with standard methods used nowadays in nuclear plants for training neural networks.

Main area of the proposed paper is creation of effective dynamic models for prediction of possible critical situations. Created system can be used in supervisor mode for supporting decision processes by human professionals. Measured data were obtained from collecting the real system values in nuclear plant.

Neural network models used nowadays in nuclear power plants are described in APVV project documentation for VVER 400 reactor. They are trained with iterative - newton methods provided by toolbox NNSYSID because of low prediction error in real usage.

There are many publications about neural networks, but for purposes of this paper we are using only NNSYSID toobox, which is fully described at NNSYSID homepage.

2. PROBLEM FORMULATION

Because of previously used models of neural network implementations in nuclear plant[1] we are going to train recurrent network - multilayered percepton in toolbox NNSYSID. This architecture of recurrent connections allows connect previous outputs only at input layer of neurons, not hidden layers of network in input vector, as shown in figure 1

As an activation function we are using hyperbolic tangent sigmoid:

[1] Project APVV - Application of artificial intelligent methods in modeling and control of critical processes in power industry.

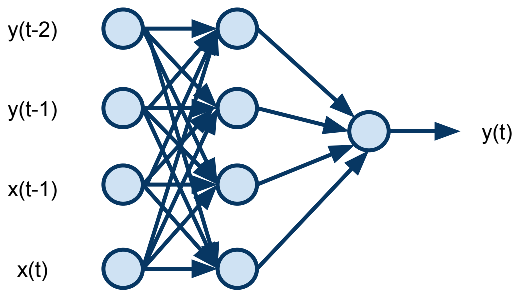


Fig. 1. Neural network architecture, x - input, y - output, t - time sample

$$y = f(\overline{x}) = \tanh\left(\sum_{i=1}^{n} w_i x_i - w_{i+1}\theta\right) \tag{1}$$

where:
$\overline{x}$ is input vector
N is number of connections to given neuron
w_i is weight, which affects input signal
θ is threshold
w_{i+1} is weight for threshold

It is possible to use another activation function, for example:

$$y = \frac{1}{1+e^{-x}} \tag{2}$$

where:
e is Euler's number

Because of the gradients of those functions are similar, it is not necessary to determine which function is better. Only the interval used to initialize network with random weights correlate with this gradient.

The whole network activity is computed by connected layers of neurons into one intelligent object and the resulting signal is computed by (for MISO neuro model):

$$\overline{x} = x_1, ..., x_N y_1, ..., y_{N-1} \tag{3}$$

$$y(t) = f_{act}\left(\sum_{i=1}^{i=L} w_{out} f_{act}\left(\sum_{i=1}^{i=K} w_{hid}^{i}\overline{x}\right)\right) \tag{4}$$

where:
$x_1 ... x_N$ are input values with delay
$y_1 ... y_{N-1}$ are previous network outputs
f_{act} is activation sigmoid function
w_{out} is vector of output layer weights
w_{hid} is matrix of hidden neurons weights

For simulating a dynamic system, we are searching for specific hidden and output layer weights combination. The solution can be interpolated into n-dimensional space (where n is number of weights in network). Than we can simply represent the solution or state of network with one point:

$$y(t) = F(W_{w_{1hid}}, ..., W_{w_{Khid}} W_{w_{1out}}, ..., W_{w_{Lhid}}) \tag{5}$$

where:
$F()$ is transfer function to y(t) from n-dimensional space
$\overline{W}$ is vector of all weights in network

To specify the sufficient condition for evaluating a quality of prediction we use prediction error and average prediction error for one sample:

$$\epsilon = \sum_{i=1}^{N} \frac{1}{2}(F(x) - G(x))^2 \tag{6}$$

$$\epsilon_{avg} = \frac{\sum_{i=1}^{N} \frac{1}{2}(F(x) - G(x))^2}{N} \tag{7}$$

where:
ϵ is general error
ϵ_{avg} is average error for 1 sample
N is number of samples used in test set
$F(x)$ is computed output of simulated system
$G(x)$ is real output of the system

Different training algorithms comparison should be done trough monitoring the neural network state in n-dimensional space after each iteration (if the algorithm is iterative-based, but most are). Next preference of algorithm is one step Hamming distance between the iterations. Stochastic model evaluating the next-step position in n-dimensional state is useful for measurement the probability of Hamming distance:

$$x(t+1)_{p_1 ... p_N} \in \mathrm{sph}(x(t)_{p_1 ... p_N}; \overline{\delta}) \tag{8}$$

$$\overline{\delta} = \Delta_{max} p_1^2; ...; \Delta_{max} p_N^2 \tag{9}$$

where:
$\mathrm{sph}(\overline{x}, \overline{y})$ is n-dimensional spheroid function
$x(t)_{p_1 ... p_N}$ are neural network state coordinates
$\overline{\delta}$ is vector of n-dimensional spheroid dimensions
$x(t+1)_{p_1 ... p_N}$ is possible state in next iteration
$\Delta_{max} p$ is maximum possible change in current weight

Main idea - nuclear plant performance prediction, should be computed from reactivity, neutron flux (in nuclear reactor) and warm and cold balance valves temperatures [2] . Progress examples of these variables are displayed at figures 2, 3, 4, 5. The output of system (which is also the expected output of our model) - performance is displayed at figure 6.

As seen on figures (and also all industry data) data are recorded with some noise (additive white noise, $1/f$ pink noise, gray noise [3]). To avoid network from learning useless noise, it is necessary to filter all input signals before putting it to neural network inputs. The mostly used filtration methods to smooth a noise are: moving average and Savitzky-Golay filter. One good example of moving average is exponential moving average:

$$EMA_N = \frac{\sum_{i=N-n}^{N} w\alpha^{N-i} m(i)}{\sum_{i=N-n}^{N} w\alpha^{N-i}} \tag{10}$$

[2] Same, as were trained models in project APVV
[3] In this case we have a white noise on our data

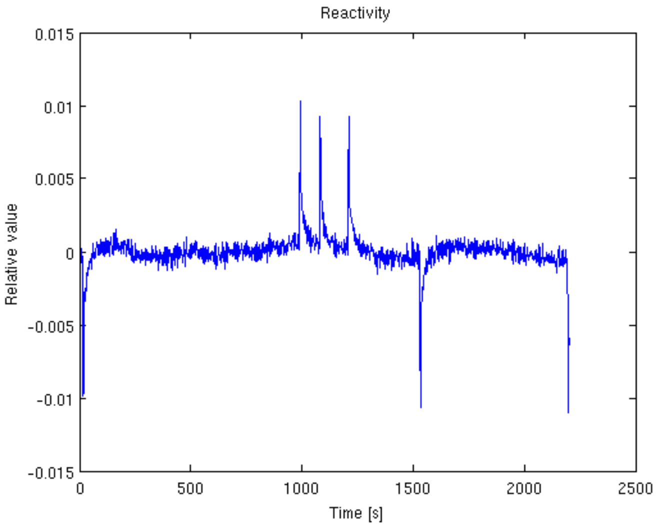


Fig. 2. Time response of reactivity

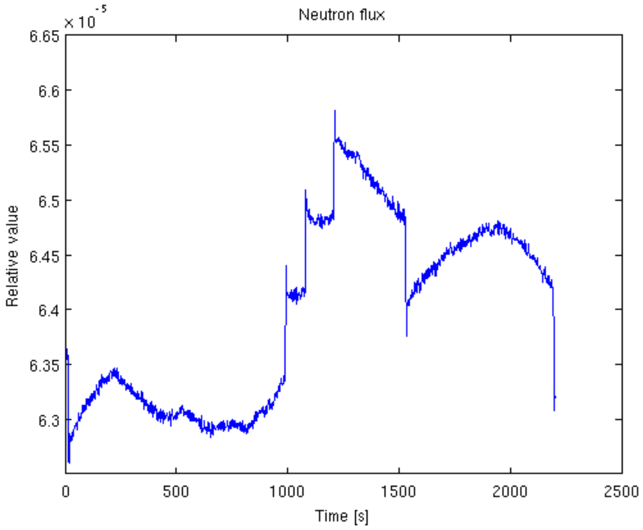


Fig. 3. Time response of neuron flux

where:
N is index of measurment
w is weight of previous measurment
α is decreasing coefficient of previous weights
m is vector of measured values
Savitzky-Golay filter is able to represent local extremes on noisy data.

3. CASE STUDY: MODEL IMPLEMENTATION

The goal is to explore new methods in neural network training, reduce known disadvantages of currently used learning algorithms. When we use Newton optimization methods, training process strongly depends on Gradients, Hessians, local extremes, process characteristic. These properties have major impact on train / test set prediction error ratio. Genetic optimizations are different. The main difference is, that the steps between iterations are the same [4] and the error signal doesn't affect the weights directly, but networks with bad results are replaced with better ones. Before solving an optimization problem with genetic (evolutionary) algorithm is necessary to create transfer function from problem state to gene. After a transfer function is created, we have to specify a fitness function. Fitness function is used to evaluation given gene. After that, the optimization begins with M [5] randomly initialized genes. These genes are evaluated with fitness function, and sorted from the best to worst. After sort, we take some better genes for the application of genetic methods - crossover, mutation [6] . This process is repeated in iterations while we found acceptable solution.

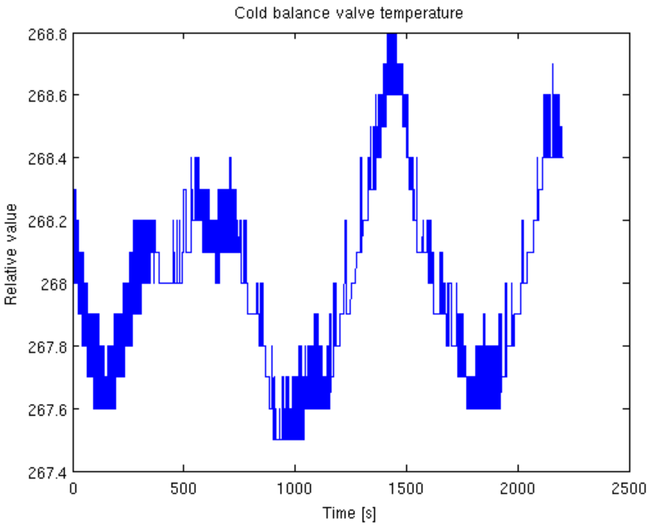


Fig. 4. Time response of cold valve temperature

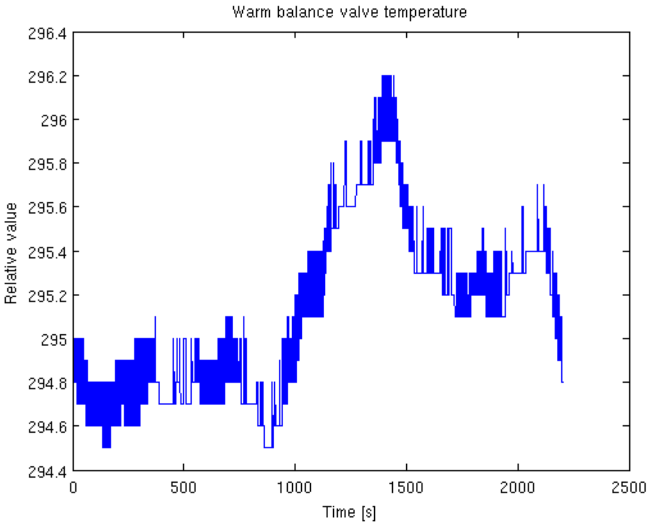


Fig. 5. Time response of warm valve temperature

In neural networks, genes should be simple all the weights, which represent the state. So the transfer function is simple - matrix form hidden and output layers weights. Fitness function should be prediction error on train data.

[4] not strictly, but comparing to newton optimization methods

[5] Typically M is between 20 - 40

[6] These are the most used in evolutionary programming, typically each genetic optimization problem need separate analysis of appropriate methods

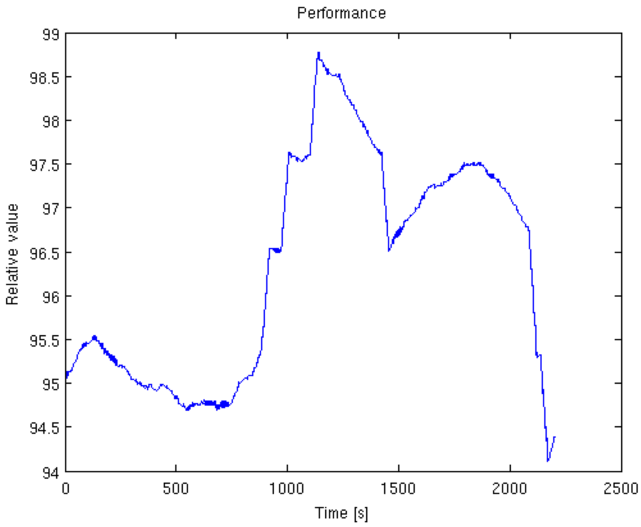


Fig. 6. Time response of performance

For better understanding both approaches, lets assume that the acceptable neuro-model solution is situated in n-dimensional space in hypercube with dimensions $[-1_{w_1} ... -1_{w_N}; 1_{w_1} ... 1_{w_N}]$, or simply all the weights are from interval $< -1, 1 >$ [7]. In Newton optimization methods are the next-step Hamming distance of weights smaller and strongly depends on error signal, so the state where can next state occur in n-dimensional space is very irregular and small spheroid [8]. Otherwise, in genetic algorithm we use mutation - random change in random weights in some genes, so the next state, where can network occur is situated into sphere, not spheroid, and this sphere covers the whole hypercube of possible solutions [9].

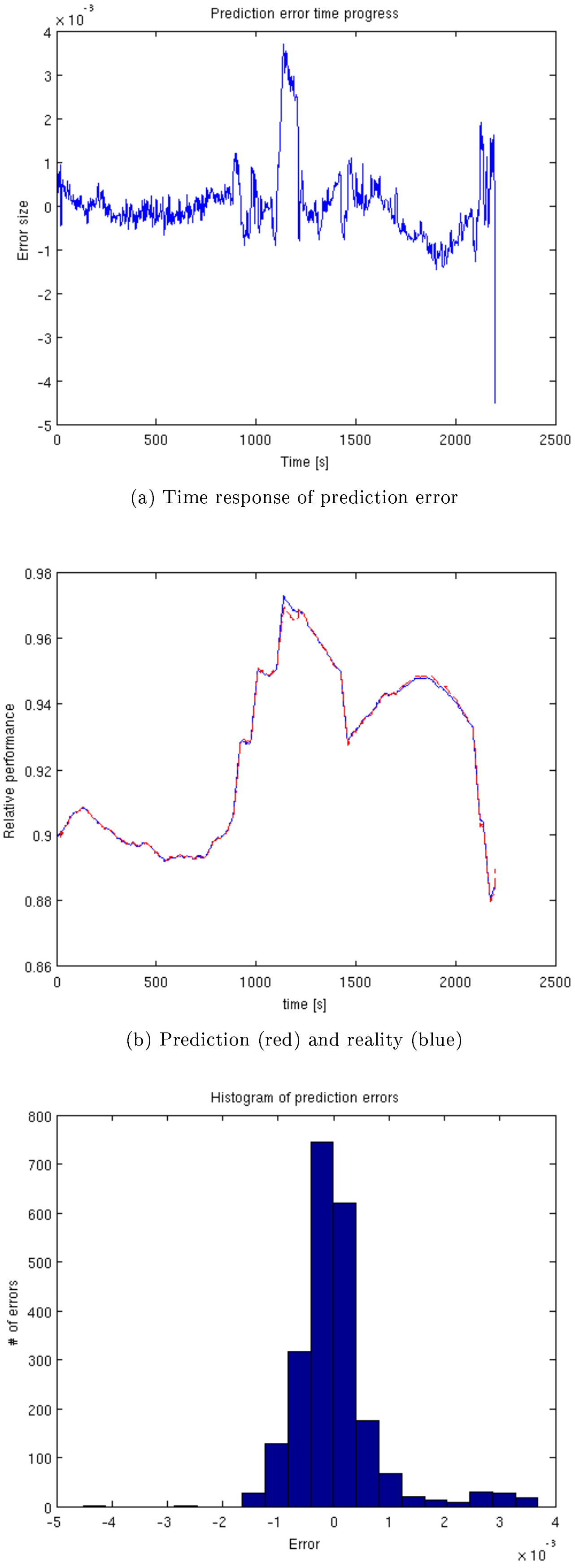
Before training, we must specify how much neurons should network have, and how much earlier measured data is necessary to put on network input. Experimentally we set 40 as number of hidden layer neurons, 3 past inputs and previous outputs on input.

First, let's look at neural network trained with Newton optimization method - Levenberg-Marquadt. This method is implemented in toolbox NNSYSID used with Matlab environment. We took some 35 minute data as a train set [10] and we started the training process. The result is shown at figure 7.

After that, we test the model on some randomly selected data (more than 5 hours). The result on test set is shown at figure 8

In genetic training, first we have to specify crossover function and mutation functions(s). For crossover we choose this function:

$$G_{new} = MG_{rand_1} + \mathrm{inv}(m)G_{rand_2} \tag{11}$$



(a) Time response of prediction error

(b) Prediction (red) and reality (blue)

(c) Prediction error histogram

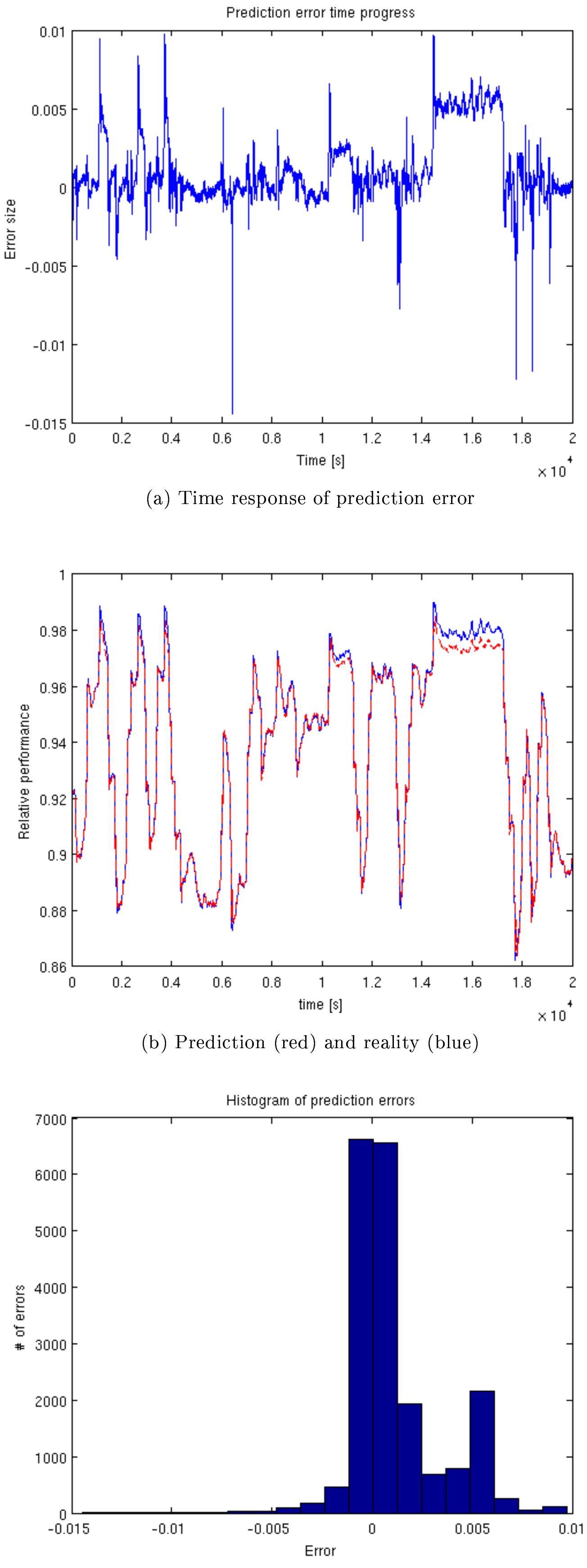
Fig. 7. Levenberg-Marquadt method, train set

[7] It's not necessary to think about bigger interval, because of fast convergence of used sigmoid functions

[8] Something like "irregular spheroid" doesn't exist, but the radiuses of every dimensions are different - also depends on error signal

[9] The probability of next state in sphere is smaller with his increasing radius, but always possible

[10] Where were performance decreasing and increasing



(a) Time response of prediction error

(b) Prediction (red) and reality (blue)

(c) Prediction error histogram

Fig. 8. Levenberg-Marquadt method, test set

where:
G_{new} new gene matrix, which represent neural network state
G_{rand_1}, G_{rand_2} are two randomly chosen genes
M is randomly generated binary matrix
inv is inverse function

And for mutation, we choose 2 mutation: classic and output layer moving mutation:

$$G_{new} = M\text{rand}(-1,1)G_{rand} \tag{12}$$

where:
G_{new} is new gene matrix
G_{rand} are randomly chosen gene from population
M is randomly generated binary matrix
rand is random function

$$G_{new} = rand(-1,1)\text{hid}(G_{rand}) \tag{13}$$

where:
G_{new} is new gene matrix
G_{rand} are randomly chosen gene from population
rand() is random function
hid() is function for extracting only hidden layer weights

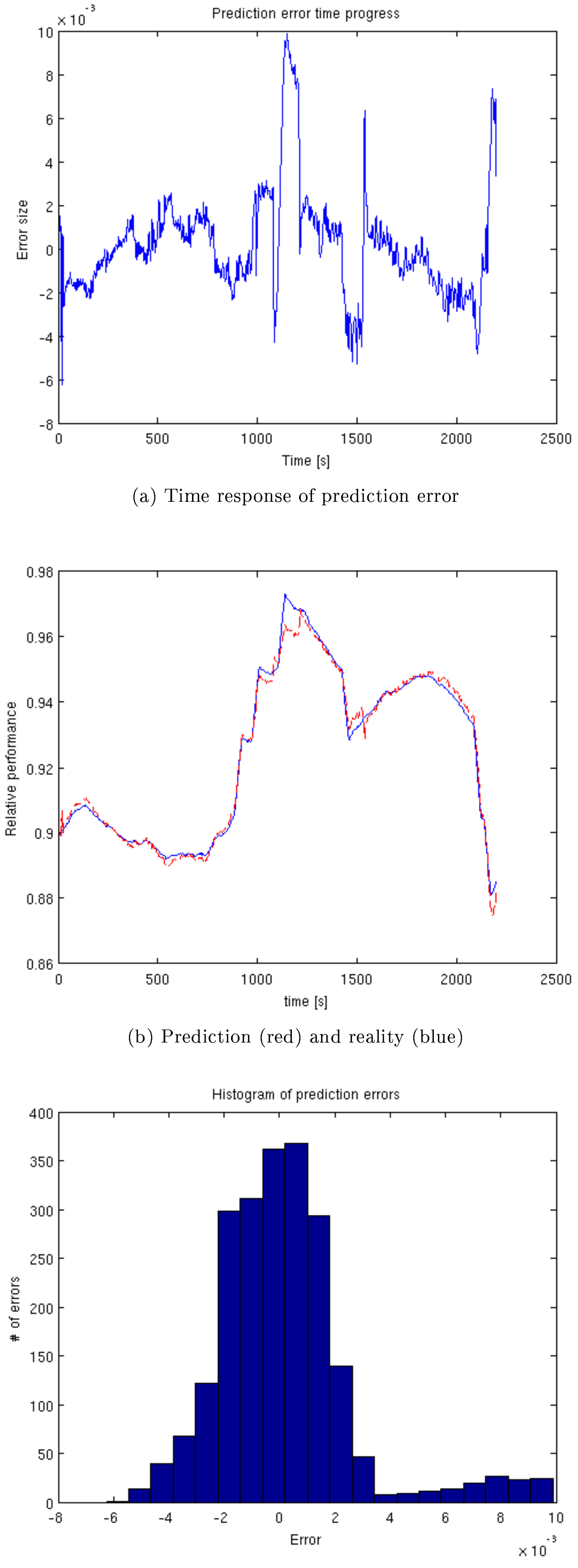
Our algorithm creates 40 randomly generated genes and sorts it in every iteration, after application of genetic methods shown above. We stop the algorithm after 1370000 iterations and it takes more than 8 days to run. To better comparison, we let the train and test set the same as in Levenberg-Marquadt method. The result of genetic method is shown at figure 9 (genetic training) and 10 (test set). Comparison of prediction errors between genetic and Levenberg-Marquadt method is shown at table 1.

Although that the prediction error with genetic training is greater, the important thing is, that the ratio between train/test set error is smaller. From this point of view we can declare, that genetic algorithm has better ability to learn process characteristic from train set and it is valuable alternative to another training methods.

This genetic algorithm was implemented in Matlab environment, integrated with NNSYSID toolbox using the same data structures describing the neural network state as have been chosen by NNSYSID toolbox authors. Algorithm has constant memory requirements, uses all CPU resources and was tested in Matlab 2010. Both approaches were tested with NNSYSID function NNVALID.

Table 1. Levenberg-Marquadt and genetic training method comparison

	Genetic alg.	LM alg.
Prediction error (Train)	3.0420×10^{-6}	3.0521×10^{-7}
Prediction error (Test)	1.5191×10^{-5}	3.2994×10^{-6}
Prediction error ratio	5	10
Runtime	8 days	approx. 15 min.

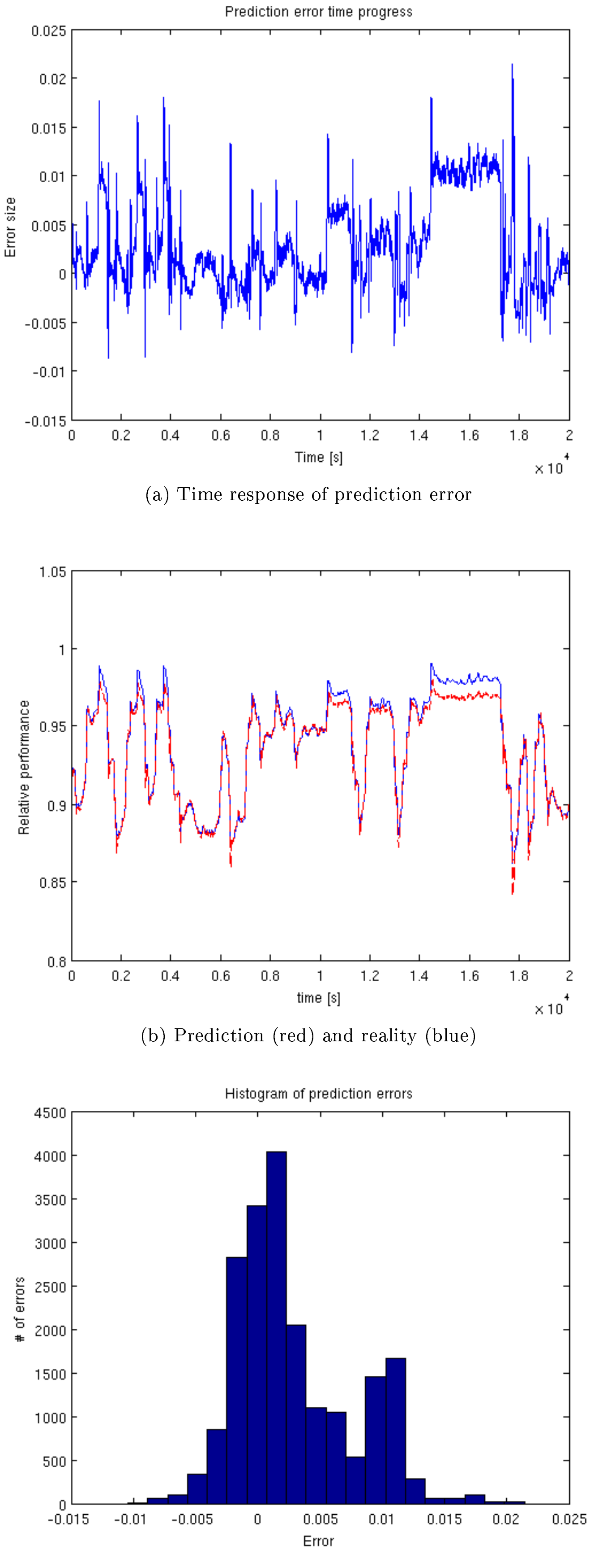


(a) Time response of prediction error

(b) Prediction (red) and reality (blue)

(c) Prediction error histogram

Fig. 9. Genetic method, train set



(a) Time response of prediction error

(b) Prediction (red) and reality (blue)

(c) Prediction error histogram

Fig. 10. Genetic method, test set

4. CONCLUSION

In this paper, we provide comparison of real implemented soft-computing genetic algorithm for neural network training with often used Levenbeg-Marquadt optimization method. With these methods were 2 neural networks trained with real data, on same test and train intervals. The result is, that genetic neural network training is acceptable alternative to other training methods with respect on better overlearning resistance. Modeled system provide prediction of nuclear plant power performance, and the prediction error were calculated from real data.

Solution should help with prediction of critical situations in nuclear engineers decisions processes with supporting and precalculating the state of nuclear reactor and other power systems. Main advatage of using neural networks in proces prediction is, that there is no need to strictly define process identification parameters of models (the model shall be also very good after adding more neurons and neuro-connections) unlike in many other system identification techniques.

ACKNOWLEDGMENTS

This project has been supported by VEGA project N^{O} 1/1105/11. This support is very gratefully acknowledged.

REFERENCES

NNSYSID toolbox, http://www.iau.dtu.dk/research/control/nnsysid.html

Kvasnička, V., Pospíchal, J., Kozák, Š., Návrat, P.: *Umelá inteligencia a kognitívna veda.* 1st edition. Bratislava: Slovenská technická univerzita v Bratislave. Fakulta informatiky a informačných technológií, 2009, 457

Sekaj, I.: *Evolučné výpočty a ich využitie v praxi.* 1st edition. Bratislava: IRIS, 2005, 157

Kvasnička, V., Pospíchal, J., Tiňo, P.: *Evolučné algoritmy.* 1st edition. Bratislava: Slovenská technická univerzita v Bratislave, 2000, 215

Rosinová, D., Dúbravská, M.: *Optimalizácia.* 1st edition. Bratislava: Slovenská technická univerzita v Bratislave, 2007, 195

Návrat, P. a kol.: *Umelá inteligencia.* 1st edition. Bratislava: Slovenská technická univerzita v Bratislave, 2007, 393

Bratko, R.: *Matlab II. Optimalizácia.* 1st edtion. Praha: VŠCHT, 2008, 227

National Instruments: *Selecting a Model Structure in the System Identification Process* http://zone.ni.com/devzone/cda/tut/p/id/4028#toc6, 2.5.2010

Rail, R.: *Neural Networks A Systematic Introduction* 1st edition. New York: Springer, 1996, 722

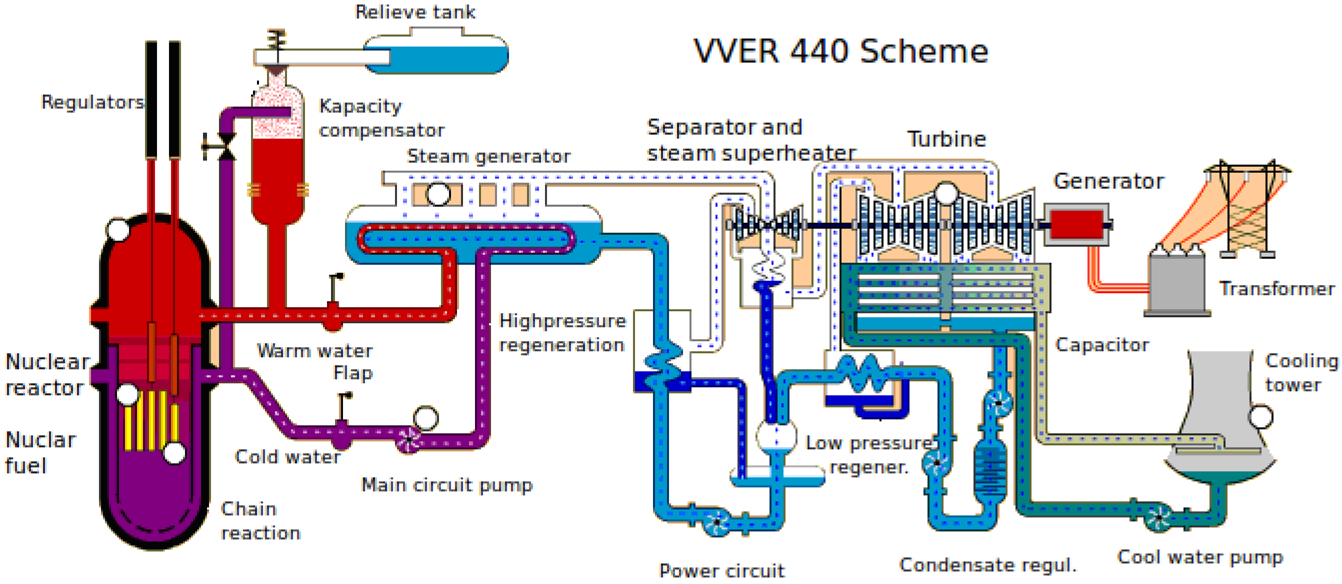


Fig. 11. Nuclear plant scheme - reactor VVER 440 Scheme

Dual Kalman Estimation of Wind Turbine States and Parameters

Petar Mateljak Vlaho Petrović Mato Baotić

Faculty of Electrical Engineering and Computing
University of Zagreb, Croatia
(Tel: +385-1-6129-529; e-mail: petar.mateljak@fer.hr,
vlaho.petrovic@fer.hr, mato.baotic@fer.hr)

Abstract: Modern wind turbines operate in a wide range of wind speeds. Power contained in the wind is proportional to the third power of wind speed and therefore increases rapidly with increase of wind speed. To enable wind turbine operation in such a variety of operating conditions, sophisticated control and estimation algorithms are needed. In this paper, a method for wind turbine state and parameter estimation is proposed. The described estimation is experimentally tested on laboratory wind turbine.

Keywords: Wind turbine, state estimation, parameter estimation, dual Kalman filter.

1. INTRODUCTION

Modern wind turbines operate in a wide range of wind speeds, typically from 3 m/s to 25 m/s. Power contained in wind is proportional to the third power of wind speed and therefore increases rapidly with increased wind speed, Burton et al. (2001). To enable wind turbine operation in such a variety of operating conditions, a sophisticated control system is needed. During weak winds, the control system has to optimise wind energy conversion by using appropriate generator torque. On the other hand, during strong winds, wind turbine power has to be constrained. An efficient way to constrain wind energy capture is pitching the rotor blades around their longitudinal axis, i.e. pitch control.

To employ more complex control algorithms, state estimation is often needed, Simon (2006). Due to wind turbine high nonlinearity and parameters uncertainty, it is not possible to use a linear model for control design and state estimation in the whole operating region. Therefore, Kalman filter for state and parameter estimation is proposed in this paper. The estimation is experimentally verified in Laboratory for Renewable Energy Sources (LARES) on Faculty of Electrical Engineering and Computing, University of Zagreb. Brief description of LARES can be found in Section 2. In Section 3, mathematical model of the wind turbine is given. State and parameter estimation method based on Kalman filter theory is presented in Section 4. Wind speed estimation is described in Section 5. Finally, in Section 6, experimental results are shown with brief conclusion.

2. DESCRIPTION OF THE LABORATORY

Laboratory for Renewable Energy Sources is located at the Faculty of Electrical Engineering and Computing, University of Zagreb. The research in the laboratory is focused on three areas: (i) wind energy, (ii) solar energy and (iii) energy storage using hydrogen fuel cell stack with metal hydride storage. Besides the research of each area independently, their connection into a micro grid is also being investigated, see Perić et al. (2010) for more details. The principle scheme of the laboratory is shown in Figure 1.

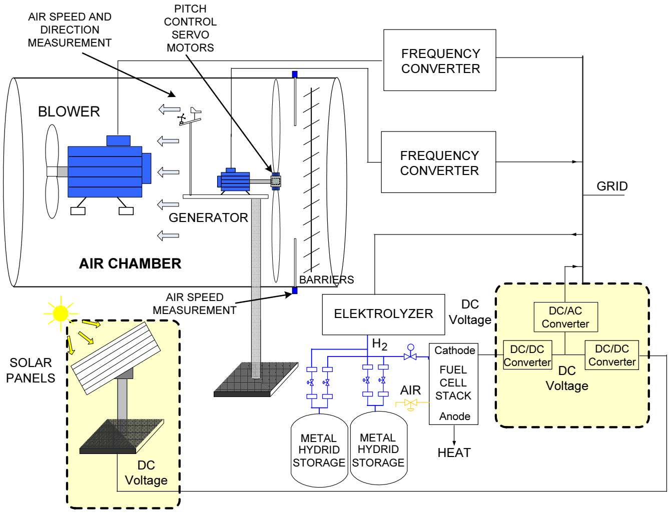


Fig. 1. Principle scheme of the Laboratory for Renewable Energy Sources, Perić et al. (2010).

The laboratory's wind turbine set-up consists of scaled wind turbine placed in air chamber with wind blower for generating wind, as shown in Figure 2.

The laboratory wind turbine is specially constructed so all aerodynamic relations present at MW-scale wind turbines are preserved. Furthermore, construction of laboratory wind turbine tower enables oscillatory fore-aft tower motion. To be able to use the same control strategies as on MW-scale wind turbine, the laboratory wind turbine is equipped with: (i) three servo drives that enable individual pitch control and (ii) synchronous generator with

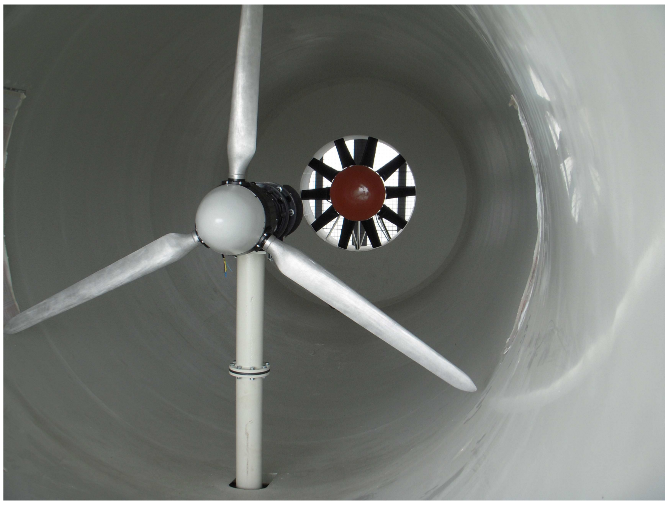


Fig. 2. The laboratory wind turbine set-up.

frequency converter for torque control. Control algorithms are implemented on PC using LabVIEW (*Laboratory Virtual Instrumentation Engineering* Workbench) platform, Johnson and Jennings (2006). Measurements of the system variables and control signal generation are obtained by a specialized input-output PXI and cRIO circuits, as shown in Figure 3.

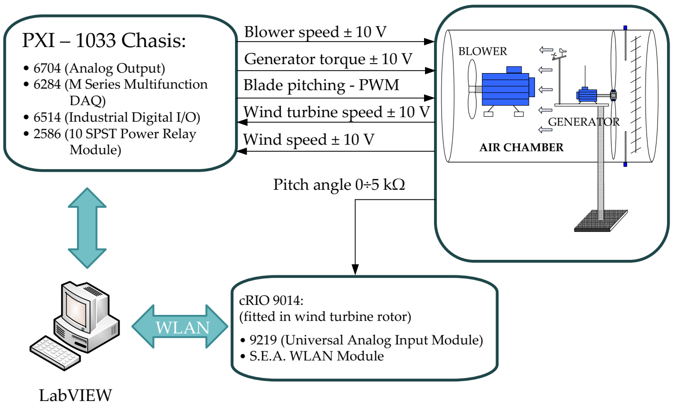


Fig. 3. The basic wind turbine control scheme.

3. WIND TURBINE MATHEMATICAL MODEL

Very common method used for wind turbine modelling is blade element and momentum theory which yields reliable and detailed wind turbine model (Burton et al. (2001)). However, such models describe wind turbine behaviour with implicit equations which are not suitable for controller design. Therefore, a simpler model that uses quasi steady state relations for aerodynamic phenomena is preferred. Such model can easily be used for control and estimator design, but it is still detailed enough to offer necessary insight into physics of the wind turbine.

The motion of the rotor can be described with equation:

$$J_t\dot{\omega} = M_a - M_g, \tag{1}$$

where ω is rotor speed, M_g is generator electromagnetic torque, M_a is aerodynamic torque and J_t is turbine moment of inertia. The aerodynamic torque can be computed as:

$$M_a = \frac{\pi}{2}\rho_a R_b^3 \bar{v}_w^2 C_Q(\bar{v}_w, \omega, \beta), \tag{2}$$

where $\bar{v}_w$ is effective wind speed on wind turbine rotor, R_b is blade length, and ρ_a is air density and C_Q is torque coefficient. Torque coefficient C_Q describes the steady state dependence of aerodynamic torque on wind speed, rotor speed and pitch angle β, Hau (2006).

It should be noted that wind is not uniform and its speed varies over the rotor area. Therefore effective wind speed $\bar{v}_w$ used in this model is not wind speed in any particular point on rotor area. The effective wind speed $\bar{v}_w$ is defined as wind speed of uniform wind that would have the same effect on wind turbine as real nonuniform wind. For most applications, information about effective wind speed is more useful than information about wind speed on any particular point on rotor area, cf. van der Hooft and van Engelen (2003).

Important parts of wind turbine dynamics are tower oscillations. Namely, wind turbine structure is very flexible due to great dimensions of its components and need for their moderate mass. In this paper, only tower flexibility is considered, while blades are assumed to be completely stiff. This assumption is valid for most MW-scale wind turbines that are in use at the present time and for the laboratory wind turbine described in Section 2. However, for larger turbines, (5 MW and more) we should probably take into account flexing of the blades as well. The first harmonic in tower oscillations is the most dominant, so tower dynamics can be approximated by (van Engelen et al. (2007)):

$$M\ddot{x}_t + D\dot{x}_t + Cx_t = F_t, \tag{3}$$

where x_t is the tower top deflection and M, D and C are the modal mass, damping and stiffness, respectively. The aerodynamic thrust force is defined similarly to aerodynamic torque:

$$F_t = \frac{\pi}{2}\rho_a R_b^2 \bar{v}_w^2 C_T(\bar{v}_w, \omega, \beta), \tag{4}$$

where thrust coefficient C_T describes the steady state dependence of aerodynamic thrust force on wind speed, rotor speed and pitch angle, Hau (2006).

Due to tower motion, wind turbine rotor is not influenced by absolute wind speed v_w, but by apparent wind speed that is derived from absolute wind speed and tower top speed:

$$\bar{v}_w = v_w - \dot{x}_t. \tag{5}$$

Tower oscillations are small in magnitude compared to wind speed, but they significantly contribute to wind turbine behaviour and pose main limitation to wind turbine pitch control (Jelavić and Perić (2009)).

For parameter estimation, it is more convenient to have a linear discrete-time state space model of the system. To this end we introduce state vector $x \in \mathbb{R}^3$ and input vector $u \in \mathbb{R}^3$ as follows:

$$x = \begin{bmatrix} \omega \\ x_t \\ \dot{x}_t \end{bmatrix}, \quad u = \begin{bmatrix} v_w \\ \beta \\ M_g \end{bmatrix}. \tag{6}$$

The nonlinear mathematical model (1) – (5) can now be linearised around an operating point (O.P. $\equiv (x^0, u^0)$) and written in the state space form:

$$\dot{x} = Ax + Bu, \tag{7}$$

with matrices $A \in \mathbb{R}^{3\times3}$ and $B \in \mathbb{R}^{3\times3}$ as follows:

$$A = \begin{bmatrix} \dfrac{M_\omega}{J_t} & 0 & -\dfrac{M_v}{J_t} \\ 0 & 0 & 1 \\ \dfrac{F_\omega}{M} & -\dfrac{C}{M} & -\dfrac{F_v + D}{M} \end{bmatrix}, \tag{8}$$

$$B = \begin{bmatrix} \dfrac{M_v}{J_t} & \dfrac{M_\beta}{J_t} & -\dfrac{1}{J_t} \\ 0 & 0 & 0 \\ \dfrac{F_v}{M} & \dfrac{F_\beta}{M} & 0 \end{bmatrix}, \tag{9}$$

The coefficients M_ω, F_ω, M_v, F_v, M_β and F_β, introduced in (8) and (9), are partial derivatives of aerodynamic torque and thrust force around operating point:

$$\begin{aligned} M_\omega &= \left.\frac{\partial M_a}{\partial \omega}\right|_{O.P.}, \; F_\omega = \left.\frac{\partial F_t}{\partial \omega}\right|_{O.P.}, \\ M_v &= \left.\frac{\partial M_a}{\partial v_w}\right|_{O.P.}, \; F_v = \left.\frac{\partial F_t}{\partial v_w}\right|_{O.P.}, \\ M_\beta &= \left.\frac{\partial M_a}{\partial \beta}\right|_{O.P.}, \; F_\beta = \left.\frac{\partial F_t}{\partial \beta}\right|_{O.P.}. \end{aligned} \tag{10}$$

Partial derivatives (10) and other wind turbine parameters used in this section can be obtained from professional wind turbine simulation tools, e.g., GH Bladed Bossanyi (2009).

Finally, a discrete time model of the wind turbine in the form:

$$x(k+1) = \Phi x(k) + \Gamma u(k), \tag{11}$$

can be obtained from the continuous-time model (7)–(10) with the following approximation (Franklin et al. (1997)):

$$\Phi = e^{AT} \approx I + AT, \tag{12}$$

$$\Gamma = \int_0^T e^{A\tau} B d\tau \approx BT, \tag{13}$$

where T is the sampling time.

4. STATE AND PARAMETER ESTIMATION

As shown in Section 3, wind turbine is a highly nonlinear system. For this reason we have chosen the extended Kalman filter (EKF) as an algorithm for the state and parameter estimation, cf. Simon (2006).

In general, the discrete-time EKF considers the nonlinear system in the state space form:

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}), \\ y_k &= h_k(x_k, v_k), \end{aligned} \tag{14}$$

where $x_k \in \mathbb{R}^n$ is the state vector, $u_k \in \mathbb{R}^m$ is the input vector, $y_k \in \mathbb{R}^p$ is the output (measurement) vector, w_k is the process noise, and v_k is the measurement noise at time step k. The process and measurement noise are assumed to be zero-mean stochastic variables with normal distribution, i.e.,

$$\begin{aligned} w_k &\sim (0, Q_k), \\ v_k &\sim (0, R_k), \end{aligned} \tag{15}$$

where Q_k and R_k are corresponding covariances.

The basic idea of the extended Kalman filter is to linearise (i.e., compute the first order Taylor approximation of) the nonlinear system (14) around the Kalman filter estimate of the states. At the same time the Kalman filter estimate of the states is based on the obtained linearised system.

Kalman filter is a recursive estimator that can be decomposed into two phases: prediction and correction, which are performed at every time instant k. In the prediction phase a priori estimation of the state ($\hat{x}_k^-$) is obtained from the system model (14) as if there was no process noise. In the correction phase an improved state estimate is found ($\hat{x}_k^+$) by utilizing the actual measurements. This update is achieved via the so-called Kalman gain matrix, K_k. The quality of the estimation is captured in the error covariance matrix P_k, which is also updated in two phases, i.e., we have a priori value P_k^- and a posteriori value P_k^+.

In the following we give brief summary of the computational steps for discrete-time extended Kalman filter, see Simon (2006) for more details:

(1) Initialize the filter

$$\begin{aligned} \hat{x}_0^+ &= \mathrm{E}(x_0), \\ P_0^+ &= \mathrm{E}\left[(x_0 - \hat{x}_0^+) \cdot (x_0 - \hat{x}_0^+)^T\right], \\ k &= 1. \end{aligned} \tag{16}$$

(2) *Prediction phase*: compute the time update of the state estimate and estimation-error covariance

$$\begin{aligned} \hat{x}_k^- &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0), \\ P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T, \end{aligned} \tag{17}$$

where

$$\begin{aligned} F_{k-1} &:= \left.\frac{\partial f_{k-1}}{\partial x}\right|_{(\hat{x}_{k-1}^+, u_{k-1}, 0)} \\ L_{k-1} &:= \left.\frac{\partial f_{k-1}}{\partial w}\right|_{(\hat{x}_{k-1}^+, u_{k-1}, 0)} \end{aligned} \tag{18}$$

(3) *Correction phase*: compute the measurement update of the state estimate and estimation-error covariance

$$\begin{aligned} K_k &= P_k^- H_k^T \left(H_k P_k^- H_k^T + M_k R_k M_k^T\right)^{-1}, \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k \left[y_k - h_k(\hat{x}_k^-, 0)\right], \\ P_k^+ &= (I - K_k H_k) P_k^-, \end{aligned} \tag{19}$$

where

$$\begin{aligned} H_k &:= \left.\frac{\partial h_k}{\partial x}\right|_{(\hat{x}_k^-, 0)} \\ M_k &:= \left.\frac{\partial h_k}{\partial v}\right|_{(\hat{x}_k^-, 0)} \end{aligned} \tag{20}$$

(4) Increase k and go to step 2.

The estimation process is illustrated in Figure 4.

As already mentioned, besides estimation of the signals, the extended Kalman filter can be used for parameter

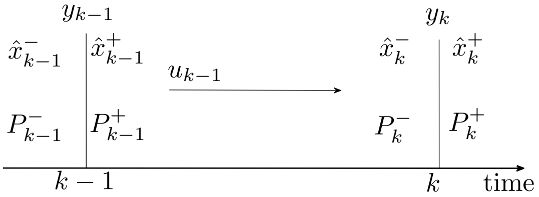


Fig. 4. Kalman filter estimation process.

estimation as well, cf. Wan and Nelson (2001). Basic idea is to define state vector with parameter values and perform estimation procedure.

In control systems one usually assumes that the parameter changes are slow, i.e., that one can write:

$$p_{k+1} = p_k + \delta_k, \tag{21}$$

where p_k is the parameter vector and δ_k is the parameter uncertainty at discrete time step k. The parameter uncertainty δ_k is defined as a stochastic variable with a zero mean value and standard deviation Q_δ.

In case of a wind turbine, the parameter changes cannot be observed as in (21), because parameter values depend on wind speed. So, mathematical model, given in (12) and (13) is used to obtain expected parameter values. In this paper, both of these approaches are used in order to obtain a priori estimation as it is proposed in following equation:

$$p^-_k = \alpha \cdot p^+_{k-1} + (1-\alpha) \cdot \tilde{p}_k, \tag{22}$$

where p^-_k is a priori parameter estimation at discrete time step k, p^+_{k-1} is a posteriori parameter estimation at discrete time step $(k-1)$, $\tilde{p}_k$ is expected parameter value based on mathematical model and α is tuning coefficient.

It must be noted that a posteriori estimation is calculated using Kalman gain matrix as in case of state estimation.

Wind turbine is a highly nonlinear system whose dynamics strongly depend on wind speed, so its parameters are changing in time. The goal of this paper is to estimate the changing parameters and use obtained estimates to improve estimation of the system states. To this end we employ extended Kalman filter, which is suitable for parallel estimation of the system states and parameters, see Wan and Nelson (2001) for more details.

To carry out the dual task of state and parameter estimation it is necessary to enable synchronous operation of two Kalman filters as it is shown in Figure 5. This is achieved using the state estimator and available measurements, which will converge towards the correct estimation values. Furthermore, these values will be used as an input data to parameter estimator. Finally, calculated values are returned in the first estimator using feedback. In this way, one gets two estimators, which interact with each other, and thus provide two types of data from the process, state and parameters values. These data are usually hard or even impossible to measure. It is necessary to point out that possible estimation problems are expected because of limited number of signals that are actually measurable.

Note that, in general, the algorithm is able to calculate all parameters and state values. However, the obtained solution may not be accurate, because estimated values may be only one of possibly many solutions that satisfy system dynamics and initial conditions. So, there are limitations to the number of signals which can be reliably estimated.

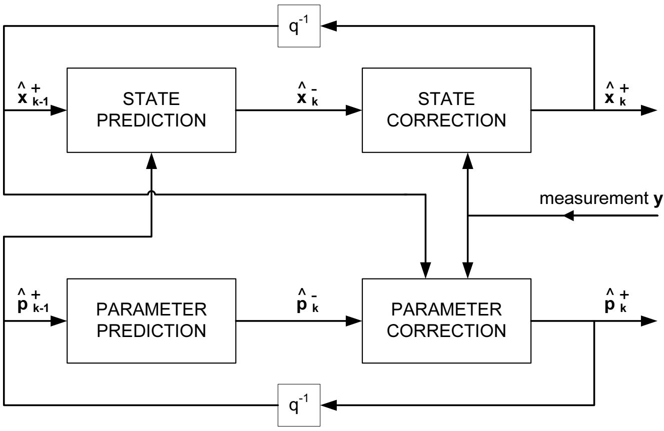


Fig. 5. Dual Kalman filter scheme.

5. WIND SPEED ESTIMATION

As it can be seen in (6), wind speed is one of process inputs and it can be shown that it contributes significantly to wind turbine behaviour. Although it would be possible to use the wind speed measurement in the Laboratory, such information about wind speed on the rotor is typically not available on a MW-scale wind turbines. Namely, anemometers are placed on wind turbine nacelle, so wind speed measurement has a significant time lag. Also, wind measured by the anemometer on a MW-scale turbine is deformed due to passing through the wind turbine rotor. Therefore, wind speed estimation is used in this paper instead of the real wind speed measurement.

Furthermore, in Section 4, it is mentioned that there may be a limit on a number of states and parameters one could estimate sufficiently well depending on the number of measured outputs. Therefore, instead of augmenting EKF from Section 4 to estimate wind speed, another approach is used. As it is shown by van der Hooft and van Engelen (2003), the effective wind speed can be estimated from (1) that describes the wind turbine rotor motion. By using expression for aerodynamic torque (2), one can readily form the following nonlinear function:

$$f_w(\hat{v}_w) = \frac{\pi}{2}\rho_a R_b^3 \hat{v}_w C_Q(\hat{v}_w, \omega, \beta) - M_g - J_t\dot{\omega}. \tag{23}$$

Pitch angle β and electromagnetic torque M_g can be easily measured, while rotor speed ω can be obtained from Kalman filter described in Section 4. The torque coefficient C_Q is based on the aerodynamic characteristics of the turbine, and can be calculated using professional simulation tools, e.g. GH Bladed, Bossanyi (2009). Clearly, when the estimated wind speed $\hat{v}_w$ is equal to the effective wind speed on the wind turbine rotor $\bar{v}_w$ then the function (23) has a zero value. Therefore, the wind speed estimation can be done by numerically solving the nonlinear equation:

$$f_w(\hat{v}_w) = 0. \tag{24}$$

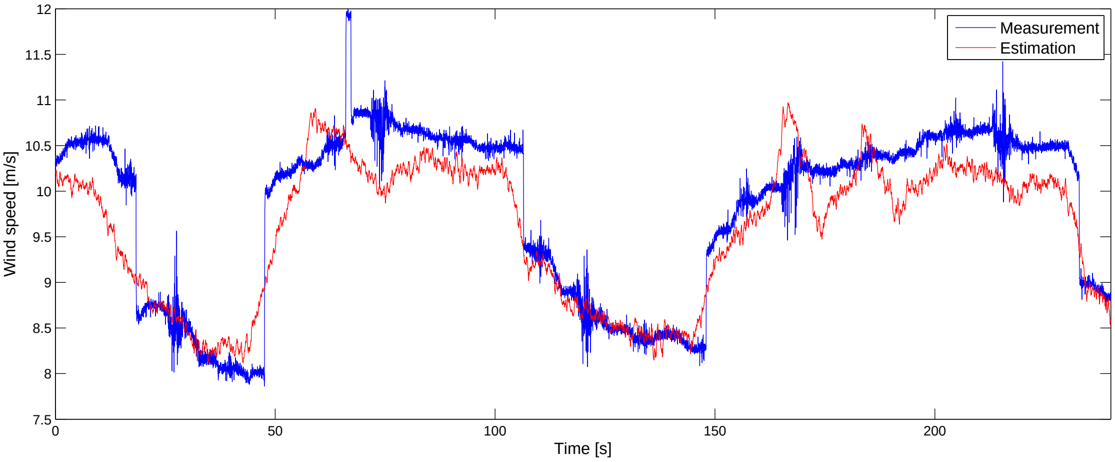


Fig. 6. Comparison between wind speed estimation and wind speed measurement.

Figure 6 shows comparison between wind speed estimation and measurement.

6. EXPERIMENTAL RESULTS

The state and parameter estimation method described in Section 4 was experimentally tested in Laboratory for Renewable Energy Sources (LARES) at the Faculty of Electrical Engineering and Computing, University of Zagreb. The wind turbine in question has the following parameters:

$$\begin{aligned} J_t &= 4\,\mathrm{Nms}^2, \\ M &= 2.321\cdot 10^4\,\mathrm{Ns}^2/\mathrm{m}, \\ D &= 4.672\cdot 10^3\,\mathrm{Ns/m}, \\ C &= 8.468\cdot 10^5\,\mathrm{N/m}. \end{aligned}$$

In the Laboratory setup the control system comprises two separated control loops. First controller is used below rated wind speed to set up the moment reference. The second controller is used above rated wind speed to obtain pitching of the rotor blades around their longitudinal axis in order to constrain the capturing of wind energy. The goal of the experiment was to verify the state and parameter estimation based on the Kalman filter theory.

Two series of experiments were made. In the first set of experiments the state and parameter estimation is performed below rated wind speed. In that case the blade pitch angle is constant and the moment controller is active. In the second set of experiments, during strong winds above rated wind speed, generator torque is on its rated value and pitch controller is active.

Estimated wind speed and rotor speed in the experiment below rated wind speed are shown in Figure 7 and Figure 8, respectively. Initial values of parameters in (10) are

$$\begin{aligned} M_\omega &= -0.082663\,\mathrm{Nms/rad}, \\ F_\omega &= -19.774\,\mathrm{Ns/rad}, \\ M_v &= 1.26908\,\mathrm{Ns}, \\ F_v &= 5.06419\,\mathrm{Ns/m}, \\ M_\beta &= 0.048625\,\mathrm{Nm/rad}, \\ F_\beta &= -19.774\,\mathrm{N/rad}, \end{aligned}$$

and initial continuous-time model (7) with

$$A = \begin{bmatrix} -0.0207 & 0 & -0.3173 \\ 0 & 0 & 1 \\ -8.55\cdot 10^{-4} & -36.627 & -0.2023 \end{bmatrix},$$

$$B = \begin{bmatrix} 0.3173 & 0.012156 & -0.25 \\ 0 & 0 & 0 \\ 2.19\cdot 10^{-4} & -6.364\cdot 10^{-4} & 0 \end{bmatrix},$$

is discretized with the sampling time of 20 ms to get a discrete-time model (11).

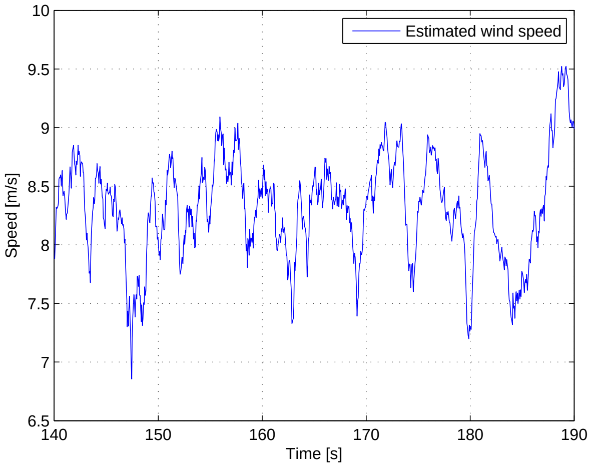


Fig. 7. Wind speed estimation below rated wind speed.

Representative results in experiments above rated wind speed are reported in Figure 9 (estimated wind speed) and Figure 10 (estimated rotor speed).

Estimation results show that the measurement noise is suppressed and estimation error is negligible in both cases – both below and above rated wind speed.

For illustration, in Figure 11 we report one representative sample of the parameter estimation results: estimation of parameter $\Gamma_{1,1}$. The blue line represents the expected value of $\Gamma_{1,1}$ based on the mathematical model (13). A posteriori estimation using the extended Kalman filter is plotted with a red line.

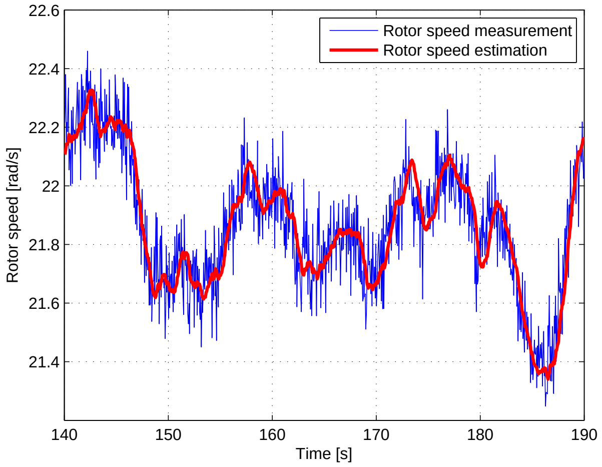


Fig. 8. Rotor speed estimation below rated wind speed.

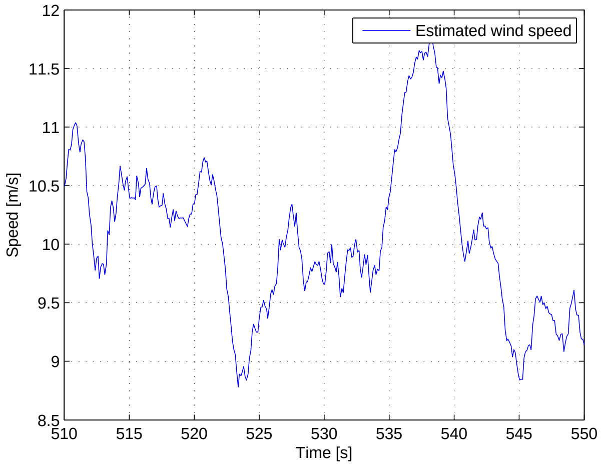


Fig. 9. Wind speed estimation above rated wind speed.

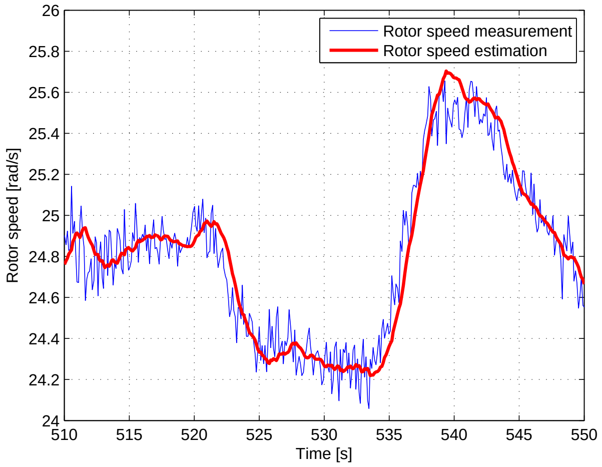


Fig. 10. Rotor speed estimation above rated wind speed.

7. CONCLUSION

In this paper, wind turbine state and parameter estimation based on a dual Kalman filter theory is implemented and experimentally tested in Laboratory for Renewable Energy Sources. A modification of the parameter estimation is used to improve a priori estimation of the state vector.

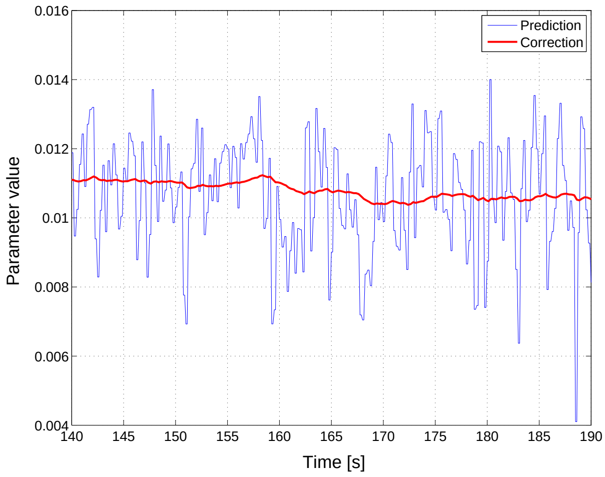


Fig. 11. Parameter estimation below rated wind speed, parameter $\Gamma_{1,1}$.

We report results for wind speed estimation and rotor speed estimation. Experiment is performed in two different operating modes, below and above rated wind speed. It is shown that the rotor speed can be estimated with a high quality in both cases. It is also possible to implement this estimation method for other states, e.g. tower top deflection.

ACKNOWLEDGEMENTS

This research has been financially supported by the Ministry of Science, Education and Sports of the Republic of Croatia under grant No 036-0361621-3012; by the National Foundation for Science, Higher Education and Technological Development of the Republic of Croatia; and by Končar - Electrical Engineering Institute. This support is gratefully acknowledged.

REFERENCES

Bossanyi, E.A. (2009). GH Bladed – User Manual, Version 3.81. Technical Report 282/BR/010, Garrad Hassan.

Burton, T., Sharpe, D., Jankings, N., and Bossanyi, E. (2001). *Wind Energy Handbook*. John Wiley & Sons.

Franklin, G.F., Powell, J.D., and Workman, M.L. (1997). *Digital Control of Dynamic Systems*. Prentice Hall.

Hau, E. (2006). *Wind Turbines: Fundamentals, Technologies, Application, Economics*. Springer, 2nd edition.

Jelavić, M. and Perić, N. (2009). Wind turbine control for highly turbulent winds. *Automatika - Journal for Control, Measurement, Electronics, Computing and Communications*, 50(3-4), 135–151.

Johnson, G.W. and Jennings, R. (2006). *LabVIEW Graphical Programming*. McGraw-Hill, New York, 4th edition.

Perić, N., Jelavić, M., Željko Ban, Domitrović, H., Matijašević, B., Kostelac, M., and Mikac, S. (2010). Wind turbine control research in Laboratory for Renewable Energy Sources. In *Proceedings of the European Wind Energy Conference, EWEC 2010*. Warsaw, Poland.

Simon, D. (2006). *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches.* John Wiley & Sons, New Jersey.

van der Hooft, E.L. and van Engelen, T.G. (2003). Feed Forward Control of Estimated Wind Speed. Technical Report ECN-C-03-137, ECN, Netherlands.

van Engelen, T.G., Markou, H., Buhl, T., and Marrant, B. (2007). Morphological Study of Aeroelastic Control Concepts for Wind Turbines. Technical report, STABCON Task-7 Report, EU-Contract ENK5-CT-2002-00627.

Wan, E.A. and Nelson, A.T. (2001). Dual Extended Kalman Filter Methods. In S. Haykin (ed.), *Kalman Filtering and Neural Networks*, 123–173. John Wiley & Sons.

Linear Matrix Inequalities Based H_∞ Control of Gantry Crane using Tensor Product Transformation

Š. Ileš [*] F. Kolonić [*] J. Matuško [*]

[*] *Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000, Zagreb, Croatia (e-mail: sandor.iles@fer.hr, fetah.kolonic@fer.hr, jadranko.matusko@fer.hr)*

Abstract: This paper describes a H_∞ controller design procedure for tensor product based model of gantry crane augumented with friction model in order to minimize friction effects. The Tensor Product (TP) model transformation is a recently proposed technique for transforming given Linear Parameter Varying (LPV) state-space models into polytopic model form, namely, to parameter varying convex combination of Linear Time Invariant (LTI) systems. H_∞ controller guarantee stability and L_2 norm bound constraint on disturbance attenuation. H_∞ controller is found using relaxed LMIs which have proof of asymptotic convergence to the global optimal controller under quadratic stability. Control algorithm is experimentally tested on single pendulum gantry (SPG).

Keywords: Parallel Distributed Compensation, Linear Matrix Inequalities, Tensor Product (TP) model transformation, gantry crane control, friction compensation.

1. INTRODUCTION

In modern industrial system, gantry cranes are widely used for the heavy loads transfer. Fast load positioning and load swinging minimization are conflicting requirements imposed to the traveling crane control systems.

For the position and anti-sway control of travelling cranes, there are several solutions, i.e., by fuzzy control, optimal control, pole placement, etc. and each of them is reported to be effective (Popadić et al., 2005), (Nalley and Trabia, 2000), (Omar, 2003). All these approaches are based on linear/linearized model of gantry crane.

Recently, nonlinear control approach based on tensor product model representation (TP) of the process is proposed (Baranyi et al., 2003), (Petres, 2006) and successfully applied to control of Single Pendulum Gantry process (Kolonić et al., 2006). The TP model represents the Linear Parameter Varying (LPV) state-space models by the parameter varying combination of Linear Time Invariant (LTI) models.

However, none of these approaches takes into account friction effect, which is unavoidable in real applications (Olsson et al., 1998). This effect may seriously degrade the performance of the control system, specially when high precision positioning is required. As a consequence steady state error due to static friction is common for all these approaches. Introducing integral action in control loop, to eliminate steady state error due to friction effects, may result in limit cycling called hunting phenomenon, see Hensen et al. (2003).

One way to cope with friction phenomena is to introduce friction compensator based on friction model. The friction model may be a-priory known or learned in an on-line manner (Lee and Tomizuka, 1996), (Huang et al., 2000), (Matuško et al., 2010), (Boras et al., 2010).

Another approach consider the friction effect as an unknown disturbance and design the robust controller. In Cheang and Chen (2000), Burul et al. (2010), H_∞ controller for linearised model is synthesized using loop-shaping methodology, and experimentally proved effective.

In this paper a combination of these two approaches is used. It is assumed that friction model is partially known, and nominal model is identified, while its associated uncertainty is considered as a disturbance. In order to cope with nonlinear nature of the gantry crane model, as well as friction model, non-linear model of gantry crane augmented with nominal friction model is rewritten in LPV form, suitable for TP transformation. As a result of TP model transformation, polytopic model of process is obtained. For such model representation, LMI control approach is used to design a controller (Tanaka and Wang, 2001), (Boyd et al., 1994), (Gahinet et al., 2002). In this paper such approach is used to synthesize H_∞ controller.

The paper is organized as follows: Section II discusses the theoretical background of TP model transformation-based control design. Section III introduces the LPV model augmented with friction and its TP transformation. Section IV discuses the LMI relaxations used in controller design. Section V presents the experimental results obtained on the single pendulum gantry (SPG) experimental model, and Section VI concludes this paper.

2. TENSOR PRODUCT MODEL TRANSFORMATION-BASED CONTROL DESIGN METHODOLOGY

Consider the linear parameter-varying state-space model

$$\begin{pmatrix}\dot{x}(t)\\ y(t)\end{pmatrix} = S(p(t))\begin{pmatrix}x(t)\\ u(t)\end{pmatrix} \tag{1}$$

with input $u(t) \in R^k$, output $y(t) \in R^l$ and state vector $x(t) \in R^m$. The system matrix

$$S(p(t)) = \begin{pmatrix}A(p(t)) & B(p(t))\\ C(p(t)) & D(p(t))\end{pmatrix} \in R^{(m+k)\times(m+l)} \tag{2}$$

is a parameter-varying object, where $p(t) \in \Omega$ is time varying parameter vector, where Ω is a closed hypercube in R^N, $\Omega = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_N, b_N]$. Parameter $p(t)$ can also include the elements of the state vector $x(t)$, therefore LPV system given in Eq. (1) is considered in the class of non-linear dynamic state space models.

The main idea of TP model transformation is to discretize the given LPV model given in Eq. (1) over hyper rectangular grid M in Ω, then via executing Higher Order Singular Value Decomposition, the tensor product structure of given model is obtained. By ignoring singular values, TP model of reduced complexity and accuracy can be obtained. For more details se Petres (2006) and Tikk et al. (2004).

Tensor product structure can be written as follows

$$\begin{aligned}S(p(t)) &= \mathcal{S}\underset{n=1}{\overset{N}{\boxtimes}} w_n(p_n)\\ &= \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_n=1}^{I_N}\prod_{n=1}^{N} w_{n,i_n}(p_n) S_{i_1,i_2\cdots,i_N},\end{aligned} \tag{3}$$

where $\mathcal{S} \in R^{I_1\times I_2\times\cdots I_N\times(m+k)\times(m+l)}$ denotes obtained tensor, I_n denotes number of LTI systems in n-th dimension Ω, $\boxtimes$ denotes multiple n-mode product of a tensor by a matrix, w_n is row vector containing $w_{n,i_n}(p_n) \in [0, 1]$ which is corresponding one variable weighting function defined on the n-th dimension of Ω and

$$S_{i_1,i_2\cdots i_N} = \begin{pmatrix}A_{i_1,i_2\cdots i_N} & B_{i_1,i_2\cdots i_N}\\ C_{i_1,i_2\cdots i_N} & D_{i_1,i_2\cdots i_N}\end{pmatrix}, \tag{4}$$

is LTI system matrix obtained by TP model transformation.

Controller is determined in same form as TP model. Control signal is given by

$$u = -\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_n=1}^{I_N}\prod_{n=1}^{N} w_{n,i_n}(p_n) K_{i_1,i_2\cdots,i_N} x \tag{5}$$

where the $K_{i_1,i_2\cdots,i_N}$ are corresponding LTI feedback gains.

By using i as linear index, equivalent to the multilinear array index with the size of $I_1 \times I_2 \times \cdots I_N$, TP model (3) and control signal (5) can be rewritten in standard polytopic form

$$S(p) = \sum_{i=1}^{R} w_i(p) S_i, \tag{6}$$

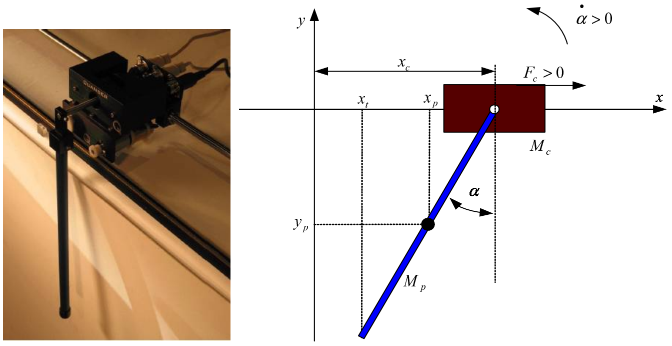
$$u = -\sum_{i=1}^{R} w_i(p) K_i, \tag{7}$$

where $R = I_1 + I_2 + \cdots + I_N$ and $w_i(p)$ is corresponding weighting function.

3. TP MODEL-BASED CONTROLLER DESIGN APPLIED TO THE SINGLE PENDULUM GANTRY CRANE EXPERIMENTAL MODEL

3.1 Mathematical model of the gantry crane

Experimental laboratory model of Single Pendulum Gantry (SPG) is used to emulate industrial crane application, see Fig 1.



(a) (b)

Fig. 1. SPG photo in mechatronics laboratory a), and schematics of the model b)

Non-linear model of single pendulum gantry (SPG) can be described by following equations [1] :

$$\begin{aligned}&(M_c + M_p)\ddot{x}_c + M_p l_p \ddot{\alpha}\cos(\alpha) - M_p l_p \dot{\alpha}^2 \sin(\alpha) =\\ &= F_m - F_f - B_{eq}\dot{x}_c,\\ &(I_p + M_p l_p^2)\ddot{\alpha} + M_p l_p \ddot{x}_c \cos(\alpha) + M_p g l_p \sin(\alpha) = -B_p\dot{\alpha},\end{aligned} \tag{8}$$

where F_m is DC motor force while F_f is the friction force. Driving force obtained from DC motor is given by:

$$F_m = \frac{\eta_g \eta_m K_g K_t}{r_{mp}} \frac{1}{R_m} U_m. \tag{9}$$

The meanings and the values of other parameters in equations (8) and (9) are given in Table 1.

In this paper Stribeck friction model is used, see Olsson et al. (1998). It can be described by

$$F_f(\dot{x}_c) = \operatorname{sgn}(\dot{x}_c)\left[F_C + (F_S - F_C)e^{-(\frac{\dot{x}_c}{v_s})^{\delta}}\right], \tag{10}$$

[1] Derivation using Lagrangian formulation is omitted for brevity and can be found in (QUANSER User Manual, 1999)

where F_C is Coulomb friction, F_S is static friction force, v_s is Stribeck velocity, and δ is emprirical exponent.

The parameters of the model (10) are experimentally identified and following values are obtained:

$$F_C = 2.70, \qquad F_S = 3.10, \qquad v_s = 0.10, \qquad \delta = 2. \tag{11}$$

3.2 LPV model of single pendulum gantry (SPG)

Letting $x = [x_1\ x_2\ x_3\ x_4]^T = [x_c\ \dot{x}_c\ \alpha\ \dot{\alpha}]^T$, $u = U_m$ and w norm bounded disturbance, the equations of motion in linear parameter varying state space form are:

$$\begin{aligned} \dot{x} &= A(x(t))x + B(x(t))u + E(x(t))w, \\ y &= C(x(t))x + D(x(t))u + F(x(t))w. \end{aligned} \tag{12}$$

The system matrix in LPV form for the model (12) can be written as:

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & a_1/a_x & a_2/a_x & a_3/a_x & b_{11}/a_x & e_1/a_x \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & a_4/a_x & a_5/a_x & a_6/a_x & b_{12}/a_x & e_2/a_x \\ 1 & 0 & c_1 & 0 & 0 & 0 \end{pmatrix}, \tag{13}$$

where :

$$\begin{aligned} a_1 &= -(I_p + M_p l_p^2) \cdot \left(\frac{\eta_g \eta_m K_g^2 K_t K_m}{r_{mp}^2 R_m} + B_{eq} \right) \\ a_2 &= \frac{M_p^2 l_p^2 g \sin(x_3) \cos(x_3)}{x_3} \\ a_3 &= (M_p^2 l_p^3 + l_p M_p l_p) \sin(x_3) x_4 + M_p l_p B_p \cos(x_3) \\ a_4 &= M_p l_p \cos(x_3) \cdot \left(B_{eq} - \frac{\eta_g \eta_m K_g^2 K_t K_m}{r_{mp}} \frac{1}{R_m} \right) \\ a_5 &= -\frac{(M_c + M_p) M_p l_p \sin(x_3)}{x_3} \\ a_6 &= -(M_c + M_p) B_p - M_p^2 l_p^2 \cos(x_3) \sin(x_3) x_4 \\ a_x &= (M_c + M_p) I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin^2(x_3) \\ b_{11} &= -(I_p M_p l_p)^2 \mathrm{sgn}(\dot{x}_c) \left[F_C + (F_S - F_C) e^{-(\frac{\dot{x}_c}{v_s})^\delta} \right] \\ b_{12} &= -M_p l_p \cos(x_3) \mathrm{sgn}(\dot{x}_c) \left[F_C + (F_S - F_C) e^{-(\frac{\dot{x}_c}{v_s})^\delta} \right] \\ e_1 &= -(I_p M_p l_p)^2 \frac{\eta_g K_g \eta_m K_t}{R_m r_{mp}} \\ e_2 &= -M_p l_p \cos(x_3) \frac{\eta_g K_g \eta_m K_t}{R_m r_{mp}} \\ c_1 &= -0.614 \frac{\sin x_3}{x_3} \end{aligned} \tag{14}$$

3.3 Single pendulum ganty TP model representation

Operating area for single pendulum gantry, is selected as

$$\begin{aligned} \Omega &= [\dot{x}_{cmin}, \dot{x}_{cmax}] \times [\alpha_{min}, \alpha_{max}] \times [\dot{\alpha}_{min}, \dot{\alpha}_{max}] \\ &= [-0.6, 0.6] \times [-0.0873, 0.0873] \times [-0.8, 0.8]. \end{aligned} \tag{15}$$

Applying the TP transformation to the model (13) yield to the TP model representation consisting of 20 LTI models. The LTI system matrices of the TP model are:

$$\begin{aligned} S_{1,1,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.009566 & 1.529 & 1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.09477 & -3.509 & -2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{2,1,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.009566 & 1.529 & -1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.09477 & -3.509 & 2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{1,2,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.000189 & 1.529 & 1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.07325 & -3.509 & -2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{2,2,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.000189 & 1.529 & -1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.07325 & -3.509 & 2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{1,3,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.517 & 0.008731 & 1.529 & 1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.78 & -26.08 & -0.0929 & -3.517 & -2.653 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{16}$$

$$\begin{aligned} S_{2,3,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.517 & 0.008731 & 1.529 & -1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.78 & -26.08 & -0.0929 & -3.517 & 2.653 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \\ S_{1,4,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.65 & 1.517 & -0.001939 & 1.53 & 1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.79 & -26.08 & -0.06834 & -3.518 & -2.654 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \\ S_{2,4,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.65 & 1.517 & -0.001939 & 1.53 & -1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.79 & -26.08 & -0.06834 & -3.518 & 2.654 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \\ S_{1,5,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.66 & 1.525 & 0.006239 & 1.531 & 1.155 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.89 & -26.13 & -0.0872 & -3.532 & -2.665 \\ 1.0 & 0 & 0.6143 & 0 & 0 & 0 \end{pmatrix} \\ S_{2,5,1} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.66 & 1.525 & 0.006239 & 1.531 & -1.155 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.89 & -26.13 & -0.0872 & -3.532 & 2.665 \\ 1.0 & 0 & 0.6143 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{17}$$

$$\begin{aligned} S_{1,1,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.000189 & 1.529 & 1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.07325 & -3.509 & -2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{2,1,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.000189 & 1.529 & -1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.07325 & -3.509 & 2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{1,2,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.009566 & 1.529 & 1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.09477 & -3.509 & -2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{2,2,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.512 & 0.009566 & 1.529 & -1.153 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.71 & -26.05 & -0.09477 & -3.509 & 2.647 \\ 1.0 & 0 & 0.6132 & 0 & 0 & 0 \end{pmatrix} \\ S_{1,3,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.517 & 0.001048 & 1.529 & 1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.78 & -26.08 & -0.07522 & -3.517 & -2.653 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{18}$$

Table 1. PARAMETERS OF THE SPG SYSTEM

Symbol	Description	Value	Unit
U_m	DC motor voltage		V
I_m	DC motor armature current		A
R_m	DC motor armature resistance	2.6	Ω
L_m	DC motor armature inductance	0.18	mH
K_t	Motor torque constant	0.007 67	N m A^{-1}
η_m	Motor efficiency	1	
K_m	Back electro-motive force constant	0.007 67	V s/rad
J_m	Rotor moment of inertia	3.9001×10^{-7}	kg m^2
I_p	Pendulum moment of inertia	0.007 883 8	kg m^2
K_g	Planetary gearbox gear ratio	3.71	
η_g	Planetary gearbox efficiency	1	
M_c	Lumped mass of the cart system	1.0731	kg
l_p	Pendulum length from pivot to COG	0.3302	m
M_p	Pendulum mass	0.23	kg
r_{mp}	Motor pinion radius	0.006 35	m
B_{eq}	Equivalent viscous damping coeff.	5.4	N m s rad^{-1}
B_p	Viscous damping coefficient	0.0024	N m s rad^{-1}
F_c	Driving force		N
g	Gravitational constant of earth	9.81	m/s^2
α	Pendulum angle		rad
x_c	Chart position		mm

$$
\begin{aligned}
S_{2,3,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.64 & 1.517 & 0.001048 & 1.529 & -1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.78 & -26.08 & -0.07522 & -3.517 & 2.653 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \\
S_{1,4,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.65 & 1.517 & 0.01172 & 1.53 & 1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.79 & -26.08 & -0.09978 & -3.518 & -2.654 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \\
S_{2,4,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.65 & 1.517 & 0.01172 & 1.53 & -1.154 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.79 & -26.08 & -0.09978 & -3.518 & 2.654 \\ 1.0 & 0 & 0.6136 & 0 & 0 & 0 \end{pmatrix} \\
S_{1,5,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.66 & 1.525 & 0.003582 & 1.531 & 1.155 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.89 & -26.13 & -0.08108 & -3.532 & -2.665 \\ 1.0 & 0 & 0.6143 & 0 & 0 & 0 \end{pmatrix} \\
S_{2,5,2} &= \begin{pmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & -11.66 & 1.525 & 0.003582 & 1.531 & -1.155 \\ 0 & 0 & 0 & 1.0 & 0 & 0 \\ 0 & 26.89 & -26.13 & -0.08108 & -3.532 & 2.665 \\ 1.0 & 0 & 0.6143 & 0 & 0 & 0 \end{pmatrix}
\end{aligned} \tag{19}
$$

Weighting functions of the TP model are given in Fig. 2.

4. CONTROLLER DESIGN

4.1 Linear Matrix Inequalities

Recently a class of numerical optimization problems called linear matrix inequality (LMI) problems has received significant attention. These optimization problems can be solved in polynomial time and hence are tractable, at least in a theoretical sense. Interior-point methods, developed for these problems, have been found to be extremely efficient in practice. For systems and control, the importance of LMI optimization stems from the fact that a wide variety of system and control problems can be recast as LMI problems. Except for a few special cases these problems do not have analytical solutions. However, the main point is that through the LMI framework they can be efficiently solved numerically in all cases. Therefore recasting a control problem as an LMI problem is equivalent to finding a solution to the original problem.

Generally, a linear matrix inequality (LMI) has the form

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i > 0, \tag{20}$$

where $x \in R^m$ is the variable and the symmetric matrices $F_i = F_i^T$ are given. The inequality symbol in (20) means that $F(x)$ is positive definite

4.2 Control objective

In Kolonić et al. (2006) the control objective was to find stabilizing controller under quadratic stability (QS) with prescribed decay rate with minimal overshoot and constrained control signal. In order to obtain stabilizing controller Lyapunov stability approach is used, with Lyapunov function candidate given by:

$$V(x) = x^T P x > 0. \tag{21}$$

The speed of response is related to decay rate α, that is, the largest Lyapunov exponent. Therefore, the condition for desired decay rate can be written as

$$\dot{V}(x) \leq -2\alpha V(x). \tag{22}$$

The equilibrium of the continuous system in polytopic form (3) is globally asymptotically stable if there exists a common positive definite matrix P such that

$$A_i^T P + P A_i + 2\alpha P < 0; i \in (1, R). \tag{23}$$

Next, let us consider the stability of the closed-loop control system of TP model of single pendulum gantry. It is globally asymptotically stable if there exists a common positive definite matrix P such that

$$
\begin{gathered}
G_{ii}^T P + P G_{ii} + 2\alpha P < 0, \\
\left(\frac{G_{ij} + G_{ji}}{2}\right)^T P + P \left(\frac{G_{ij} + G_{ji}}{2}\right) + 2\alpha P \leq 0, i < j,
\end{gathered} \tag{24}
$$

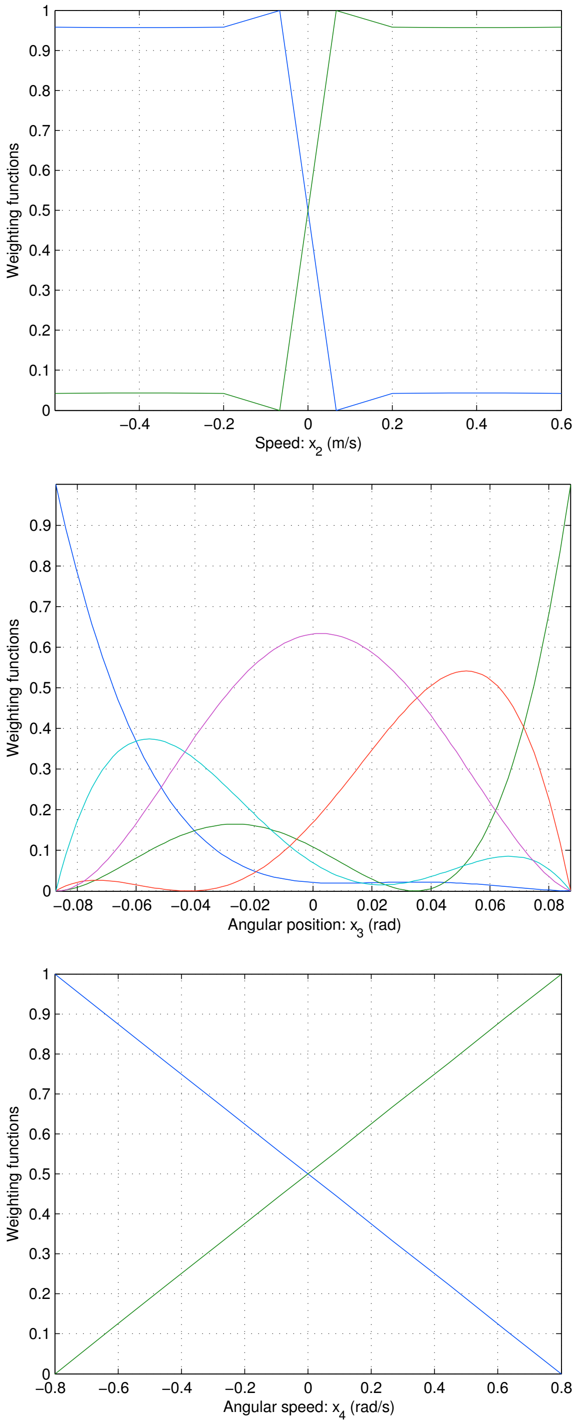


Fig. 2. Weighting functions of the TP model

where

$$G_{ij} = A_i + B_i F_j, \tag{25}$$

denotes closed loop state matrix.

Largest possible decay rate can be found by solving generalized eigenvalue minimization problem (GEVP):

$$\begin{aligned}
&\text{maximize } \alpha\\
&\text{subject to}\\
&\qquad X > 0\\
&\qquad -XA_i^T - A_iX + M_i^TB_i^T + B_iM_i + 2\alpha X > 0\\
&\qquad -XA_i^T - A_iX - XA_j^T - A_jX + M_j^TB_i^T\\
&\qquad + B_iM_j + M_i^TB_j^T + B_jM_i - 4\alpha X \geq 0
\end{aligned} \tag{26}$$

where $X = P^{-1}$ and $M_i = F_iX$.

In order to satisfy the constraints on control input and output constraints, the following LMIs are added to the (26).

Constraint on the control value:

Assume that initial condition $x(0)$ is unknown, but its upper bound $\|x(0)\| \leq \phi$ is known, which can be recast as following LMI

$$\phi^2 I \leq X, \tag{27}$$

the constraint $\|u\|_2 \leq \mu$ is enforced $\forall t \geq 0$ if the following LMI holds

$$\begin{pmatrix} X & M_i^T \\ M_i & \mu^2 I \end{pmatrix} \geq 0. \tag{28}$$

Constraint on the output:

Assume that condition (27) is satisfied, the constraint $\|y(t)\|_2 \leq \lambda$ is enforced, $\forall t \geq 0$, if the following LMI holds

$$\begin{pmatrix} X & XC_i^T \\ C_iX & \lambda^2 I \end{pmatrix} \geq 0. \tag{29}$$

LMI conditions (26) - (29) guarantee stability, constrained control signal and constrained output, however since friction effects were neglected it resulted in steady state error. In order to minimize steady state error, H_∞ norm was minimized since it is related to the capacity of the closed-loop system to reject energy bounded disturbance.

If there exist Lyapunov function (21) such that

$$\dot{V}(x) + y^T y - \gamma^2 w^T w \leq 0, \tag{30}$$

closed loop system has guaranteed H_∞ disturbance attenuation less than γ,

$$\|y(t)\|_2 \leq \gamma \|w(t)\|_2, \tag{31}$$

besides being quadratically stable. Condition (30) can be rewritten as

$$\sum_{i=1}^{R} w_i^2(p) T_{ii} + \sum_{i=1}^{R-1} \sum_{j=i+1}^{R} w_i(p) w_j(p) (T_{ij} + T_{ji}) < 0, \tag{32}$$

where

$$T_{ij} = \begin{pmatrix} \begin{pmatrix} A_iX + XA_i^T + B_iM_j + \\ +M_j^TB_i^T + E_iE_i^T \end{pmatrix} & \star \\ C_iX + F_iE_i^T + D_iM_j & F_iF_i^T - \gamma^2 I \end{pmatrix}, \tag{33}$$

which is in literature known as LMI representation of bounded real lemma (Boyd et al., 1994), (Fridman and Shaked, 2001).

Since γ is related to the level of disturbance attenuation it is of great interest to compute controller which ensures minimum value of γ.

$$\begin{aligned} &\text{minimize } \gamma \\ &\text{subject to (33)} \end{aligned} \tag{34}$$

Classical LMI conditions test the negative definiteness of (33), by imposing that the coefficients $T_{ii} < 0$ and $T_{ij} + T_{ji} < 0$, which is obviously only sufficient condition. Instead we use condition with proved convergence towards (33) in following theorem

Theorem 1: The TP model in (3) is quadratically stabilizable by means of all linear parameter-dependent state feedback control gain $K(p) = \sum_{i=1}^{R} w_i(p)K_i$ with an $\mathcal{H}_\infty$ guaranteed cost $\gamma > 0$ if and only if there exist a symmetric positive definite matrix $X \in \mathbb{R}^{n\times n}$, matrices $M_i \in \mathbb{R}^{m\times n}$, $i = 1, ..., N$, matrices $X_k \in \mathbb{R}^{2n+p+q\times n+q}$, $k \in \mathcal{N}(g)$, a degree $g \geq 1$, $g \in \mathbb{N}$, and a sufficiently large $d \in \mathbb{N}$ such that

$$\sum_{\substack{k'\in\mathcal{N}(d)\\ k\succeq k'}} \left(\sum_{i\in\{1,...,N\} k_i > k'_i} \frac{d!}{\pi(k')} \left(X_{k-k'-e_i}\mathcal{B}_i + \mathcal{B}_i^T X_{k-k'-e_i}^T \right) + \right.$$
$$+ \sum_{\substack{i,j\in\{1,...,N\}\\ k-k'-e_i-e_j\succeq 0}} \frac{d!}{\pi(k')} \cdot \frac{(g-1)!}{\pi(k-k'-e_i-e_j)}$$
$$\left. \times \begin{bmatrix} B_iM_j + M_j^T B_i^T & X & 0 & M_i^T D_j^T \\ \star & 0 & 0 & 0 \\ \star & \star & I & 0 \\ \star & \star & \star & -\gamma^2 I \end{bmatrix} \right) < 0,$$
$$\forall k \in \mathcal{N}(g+d+1), \tag{35}$$

with

$$\mathcal{B}_i = \begin{bmatrix} A_i^T & -I & 0 & C_i^T \\ E_i^T & 0 & -I & F_i^T \end{bmatrix}, \tag{36}$$

where $\mathcal{N}(g)$ is set of N-tuples obtained as all possible combinations of nonnegative integers k_i, $i \in [1, N]$, such that $\sum_{i=1}^{N} k_i = g.$, for N-tuples k, k', comparation, summation and substraction are defined componentwise, N-tuple e_i denotes N-tuple with all components equal 0, except i-th component which equals 1, and where $\pi(k)$ is defined as $\pi(k) = (k_1!)(k_2!)\ldots(k_N!)$.

Minimising γ subject to LMI constraints proposed in Theorem 1., have the asymptotic convergence to the minimum $\mathcal{H}_\infty$ guaranteed cost under quadratic stabilizability, for more details and proof of Theorem 1 see Montagner et al. (2009), `http://www.dt.fee.unicamp.br/~ricfow/robust.htm`.

In the affirmative case, local feedback gains are given by $K_i = M_iX^{-1}$, $i = 1, ..., N$.

In order to ensure constrained control input (27) and (28) are added to (35).

By using Yalmip [2] and Sedumi [3] 1.3, the following feasible solution and feedback gains are obtained:

$$\begin{aligned}
K_{1,1,1} &= \begin{pmatrix} 64.55 & 22.34 & -5.605 & 8.128 \end{pmatrix}^T \\
K_{2,1,1} &= \begin{pmatrix} 64.55 & 22.34 & -5.605 & 8.128 \end{pmatrix}^T \\
K_{1,2,1} &= \begin{pmatrix} 64.52 & 22.32 & -5.616 & 8.118 \end{pmatrix}^T \\
K_{2,2,1} &= \begin{pmatrix} 64.52 & 22.32 & -5.616 & 8.118 \end{pmatrix}^T \\
K_{1,3,1} &= \begin{pmatrix} 65.29 & 22.52 & -5.287 & 8.213 \end{pmatrix}^T \\
K_{2,3,1} &= \begin{pmatrix} 65.29 & 22.52 & -5.287 & 8.213 \end{pmatrix}^T \\
K_{1,4,1} &= \begin{pmatrix} 65.34 & 22.51 & -5.267 & 8.212 \end{pmatrix}^T \\
K_{2,4,1} &= \begin{pmatrix} 65.34 & 22.51 & -5.267 & 8.212 \end{pmatrix}^T \\
K_{1,5,1} &= \begin{pmatrix} 66.71 & 22.86 & -4.684 & 8.374 \end{pmatrix}^T \\
K_{2,5,1} &= \begin{pmatrix} 66.71 & 22.86 & -4.684 & 8.374 \end{pmatrix}^T \\
K_{1,1,2} &= \begin{pmatrix} 64.52 & 22.32 & -5.616 & 8.118 \end{pmatrix}^T \\
K_{2,1,2} &= \begin{pmatrix} 64.52 & 22.32 & -5.616 & 8.118 \end{pmatrix}^T \\
K_{1,2,2} &= \begin{pmatrix} 64.55 & 22.34 & -5.605 & 8.128 \end{pmatrix}^T \\
K_{2,2,2} &= \begin{pmatrix} 64.55 & 22.34 & -5.605 & 8.128 \end{pmatrix}^T \\
K_{1,3,2} &= \begin{pmatrix} 65.27 & 22.5 & -5.296 & 8.206 \end{pmatrix}^T \\
K_{2,3,2} &= \begin{pmatrix} 65.27 & 22.5 & -5.296 & 8.206 \end{pmatrix}^T \\
K_{1,4,2} &= \begin{pmatrix} 65.38 & 22.54 & -5.251 & 8.225 \end{pmatrix}^T \\
K_{2,4,2} &= \begin{pmatrix} 65.38 & 22.54 & -5.251 & 8.225 \end{pmatrix}^T \\
K_{1,5,2} &= \begin{pmatrix} 66.7 & 22.86 & -4.687 & 8.371 \end{pmatrix}^T \\
K_{2,5,2} &= \begin{pmatrix} 66.7 & 22.86 & -4.687 & 8.371 \end{pmatrix}^T
\end{aligned} \tag{37}$$

4.3 Results

Simulation and experimental results are shown on Fig 3. During simulation and experimental tests proposed H_∞ approach is compared to original TP model based approach (QS) described in Kolonić et al. (2006). It can be seen that proposed approach has successfully minimized steady state error of chart position from 14.55 mm to 2.4 mm in simulation, and from 7mm to 0.4 mm in experimental results. However, slower decay rate as well as oscillatory angle response is obtained due to increased control action needed to overcome static friction force.

5. CONCLUSION

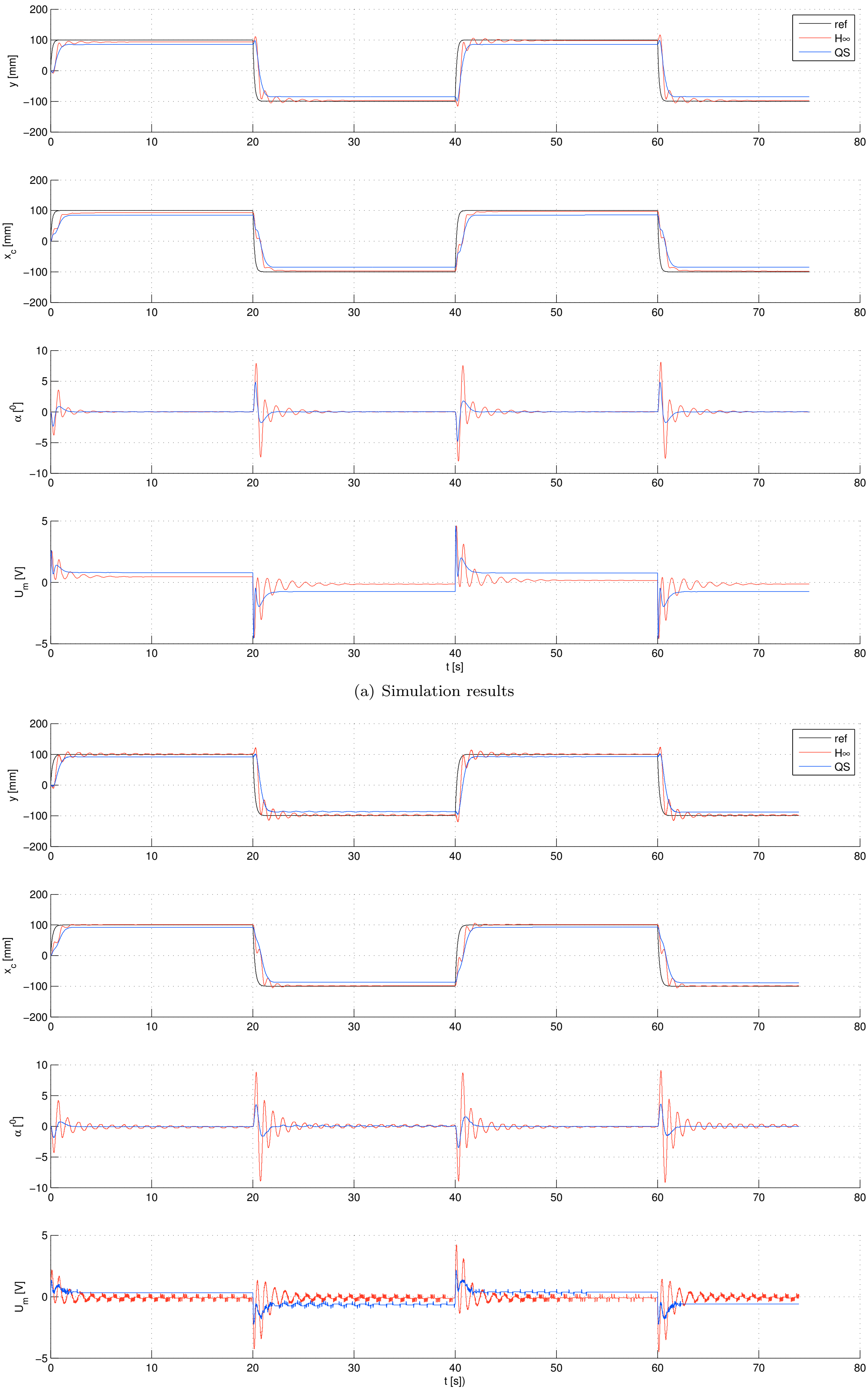
TP transform (HOSVD) creates polytopic model suitable for PDC controller synthesis via LMIs. Presented results for gantry crane are obtained by using exact TP transform. Simulation and experimental results verify that H_∞ controller has successfully minimized steady state error due to friction effect, however it resulted in oscillatory angle response and slower decay rate. In order to minimize oscillations H_∞ controller could be extended with robust pole placement inside LMI regions.

ACKNOWLEDGEMENT

This research has been supported by the Ministry of Science, Education and Sports of the Republic of Croatia under Grant No 036-0363078-1629.

[2] YALMIP is a modelling language for advanced modeling and solution of convex and nonconvex optimization problems.

[3] SeDuMi: a Matlab toolbox for optimization over symmetric cones



(a) Simulation results

(b) Experimental results

Fig. 3. Simulation and experimental results after step reference position, where y is pendulum tip position, x_c is cart position, α is pendulum angle and U_m is control input, respectively

REFERENCES

Baranyi, P., Tikk, D., Yam, Y., and Patton, R. (2003). From differential equations to PDC controller design via numerical transformation. *Computers in Industry*, 51(3), 281–297.

Boras, M., Lešić, V., Matuško, J., and Kolonić, F. (2010). B-spline neuronski kompenzator nelinearnosti sustava. In *33rd International Convention on Information and Communication Technology, Electronics and Microelectronics*.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. Society for Industrial Mathematics.

Burul, I., Kolonić, F., and Matuško, J. (2010). The control system design of a gantry crane based on H_∞ control theory. In *MIPRO, 2010 Proceedings of the 33rd International Convention*, 183–188. IEEE.

Cheang, S. and Chen, W. (2000). Stabilizing control of an inverted pendulum system based on H_∞ loop shaping design procedure. In *Intelligent Control and Automation, 2000. Proceedings of the 3rd World Congress on*, volume 5, 3385–3388. IEEE.

Fridman, E. and Shaked, U. (2001). New bounded real lemma representations for time-delay systems and their applications. *Automatic Control, IEEE Transactions on*, 46(12), 1973–1979.

Gahinet, P., Nemirovskii, A., Laub, A., and Chilali, M. (2002). The LMI control toolbox. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 3, 2038–2041. IEEE.

Hensen, R., Van de Molengraft, M., and Steinbuch, M. (2003). Friction induced hunting limit cycles: A comparison between the LuGre and switch friction model* 1. *Automatica*, 39(12), 2131–2137.

Huang, S., Tan, K., and Lee, T. (2000). Adaptive friction compensation using neural network approximations. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30(4), 551–557.

Kolonić, F., Poljugan, A., and Petrovič, I. (2006). Tensor product model transformation-based controller design for gantry crane control system–an application approach. *Acta Polytechnica Hungarica*, 3(4).

Lee, H. and Tomizuka, M. (1996). Robust motion controller design for high-accuracy positioning systems. *Industrial Electronics, IEEE Transactions on*, 43(1), 48–55.

Matuško, J., Kolonić, F., Ileš, Š., and Slutej, A. (2010). Friction compensation of gantry crane model based on the B-spline neural compensator. In *14th International Power Electronics and Motion Control Conference, EPE-PEMC 2010*.

Montagner, V., Oliveira, R., and Peres, P. (2009). Convergent LMI relaxations for quadratic stabilizability and H_∞ control of Takagi-Sugeno fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 17(4), 863–873.

Nalley, M. and Trabia, M. (2000). Control of overhead cranes using a fuzzy logic controller. *Journal of Intelligent and Fuzzy systems*, 8(1), 1–18.

Olsson, H., Astrom, K., Canudas De Wit, C., Gafvert, M., and Lischinsky, P. (1998). Friction models and friction compensation. *European Journal of Control*, 4, 176–195.

Omar, H. (2003). *Control of gantry and tower cranes*. Ph.D. thesis, Virginia Polytechnic Institute and State University.

Petres, Z. (2006). Polytopic Decomposition of Linear Parameter-Varying Models by Tensor-Product Model Transformation Ph. D. Thesis Booklet.

Popadić, T., Kolonić, F., and Poljugan, A. (2005). A fuzzy control scheme for the gantry crane position and load swing control.

QUANSER User Manual, L. (1999). Linear Experiment 4: Pole Placement. *Linear Motion Servo Plants, Instructor Handout*.

Tanaka, K. and Wang, H. (2001). *Fuzzy control systems design and analysis: a linear matrix inequality approach*. Wiley-Interscience.

Tikk, D., Baranyi, P., Patton, R., and Tar, J. (2004). Approximation Capability of TP model forms. *Australian Journal of Intelligent Information Processing Systems*, 8(3), 155–163.

Design of feedback control for unstable processes with time delay

D. Vrančić* and M. Huba**

**Department of Systems and Control, J. Stefan Institute, Ljubljana, Slovenia
e-mail: damir.vrancic@ijs.si*

***Faculty of Electrical Eng. and Inform. Tech., Ilkovičova 3, 81219 Bratislava, Slovakia
e-mail: mikulas.huba@stuba.sk*

Abstract: In this paper, the tuning method, based on characteristic areas and Magnitude Optimum (MO) criterion for some unstable processes is presented. The proposed approach is to use inner compensator, of the first or the second order, to stabilise the process. The stabilised process is controlled by 2-DOF PI controller, tuned by using MOMI or DRMO tuning method (depending on desired tracking or disturbance-rejection performance). The proposed method was tested on five linear process models. The responses were relatively fast and without oscillations, all according to the MO criterion.

Keywords: unstable processes, internal feedback, PID control

1. INTRODUCTION

Most processes in chemical and process control industries are stable and can be controlled by various types of controller structures and relatively wide range of controller parameters. However, some types of processes, like continuous stirred reactors, bioreactors or polymerisation reactors are inherently unstable. Those processes require closer attention, since, to stabilise them, controller structure and parameters should be carefully chosen (Lee et al., 2010).

Several tuning rules for different types of unstable processes have been proposed so far. Some of the methods are dedicated to PI(D) controller design for unstable processes. Jacob and Chidambaram (1996) provided tuning formulas for the first-order unstable process with delay (FODUP) for PI controllers by using model reference method, synthesis method and internal model control (IMC) method. Park et al., (1998) proposed inner proportional feedback loop for stabilising the process and outer loop with PID controller. The proposed approach is equivalent to using 2-degrees-of-freedom (2-DOF) PID controller, which is also used by Prashanti and Chidambaram (2000) to reduce process overshoots. Construction of PID controller with lead/lag filter for integrating and FODUP processes was proposed by Shamsuzzoha and Lee (2008). Additional set-point filter was applied to reduce the overshoots. Panda (2009) designed PID controller for integrating and unstable processes, based on IMC design.

The proposed approach in this paper is to use internal feedback loop to stabilise the system, similar to Park et al. (1998). However, the inner compensator is of the first or the second order. The parameters of the compensator are calculated so as to equalise characteristic areas of the actual and desired closed-loop transfer functions. Then, Magnitude Optimum Multiple Integration (MOMI) or Disturbance Rejection Magnitude Optimum (DRMO) tuning rules (Vrančić et al.; 1999a, 2001, 2004) are applied to calculate PI(D) controller parameters for such stabilised process. The proposed control scheme is given in Figure 1.

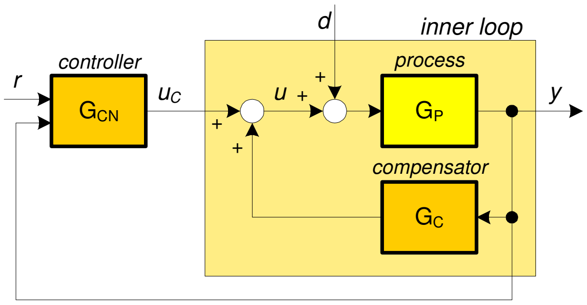


Fig. 1. Block diagram of the proposed closed-loop control.

2. DESIGN OF INTERNAL COMPENSATOR

The purpose of the compensator (Fig. 1) is to stabilise the process by forming the inner closed-loop. The compensator parameters depend on desired closed-loop properties. Let us assume that the process transfer function is the following:

$$G_P = K_{PR} \frac{num}{den} e^{-sT_{del}} , \tag{1}$$

where

$$\begin{aligned} num &= 1 + b_1 s + b_2 s^2 + b_3 s^3 + \cdots \\ den &= 1 + a_1 s + a_2 s^2 + a_3 s^3 + \cdots \end{aligned} , \tag{2}$$

and K_{PR} and T_{del} are process gain and time-delay, respectively. Let us choose the following compensator's structure:

$$G_C = K_C \frac{num_C}{den_C} , \tag{3}$$

where

$$\begin{aligned} num_C &= 1 + d_1 s + d_2 s^2 + d_3 s^3 + \cdots \\ den_C &= 1 + c_1 s + c_2 s^2 + c_3 s^3 + \cdots \end{aligned} . \tag{4}$$

Then, the closed-loop transfer function of the inner feedback loop (between signals u_C and y in Fig. 1) is:

$$G_{CL} = K_{PR} \frac{num \cdot den_C \cdot e^{-sT_{del}}}{den \cdot den_C - K_1 num \cdot num_C \cdot e^{-sT_{del}}} . \tag{5}$$

Let us define a desired closed-loop transfer function of the inner loop to have the same steady-state gain, numerator and pure time-delay as in (5):

$$G_{CLD} = \frac{K_{PR}}{1 - K_1} \frac{num \cdot e^{-sT_{del}}}{den_R} , \tag{6}$$

where

$$K_1 = K_{PR} K_C , \tag{7}$$

and den_R is a desired closed-loop denominator:

$$den_R = 1 + e_1 s + e_2 s^2 + e_3 s^3 + \cdots . \tag{8}$$

In order to make G_{CL} (5) and G_{CLD} (6) equal, the following sub-functions should become equal:

$$\begin{aligned} G_1 &= \frac{den_C}{\frac{den \cdot den_C}{1 - K_1} - \frac{K_1}{1 - K_1} num \cdot num_C \cdot e^{-sT_{del}}} \\ G_2 &= \frac{1}{den_R} \end{aligned} . \tag{9}$$

However, exact matching of G_1 and G_2 is not possible, since G_1 contains numerator and pure time-delay in denominator, which cannot be developed into finite number of terms. One possibility to make G_1 as close as possible to G_2 is to make them equal in lower frequency region by equating terms of their "characteristic areas" (Rake, 1987; Vrančić et al, 1999a). Namely, for the following transfer function:

$$G = \frac{K_P \left(1 + \beta_1 s + \beta_2 s^2 + \beta_2 s^3 + \cdots\right)}{\left(1 + \alpha_1 s + \alpha_2 s^2 + \alpha_2 s^3 + \cdots\right)} e^{-sT_d} , \tag{10}$$

the characteristic areas can be calculated as (Vrančić et al., 1999a,b):

$$\begin{aligned} A_0 &= K_P \\ A_1 &= K_P \left(\alpha_1 - \beta_1 + T_d\right) \\ A_2 &= K_P \left(\beta_2 - \alpha_2 - \beta_1 T_d + 0.5 \cdot T_d^2\right) + A_1 \alpha_1 \\ A_3 &= K_P \left(\alpha_3 - \beta_3 + \beta_2 T_d - 0.5 \beta_1 T_d^2 + \frac{T_d^3}{6}\right) + A_2 \alpha_1 - A_1 \alpha_2 \\ &\vdots \end{aligned} . \tag{11}$$

Function G_1 (9) can be expressed in terms of parameters α_i and β_i (10) by applying Taylor's expansion of time-delay term in denominator:

$$e^{-sT_{del}} \approx 1 - sT_{del} + \frac{s^2 T_{del}^2}{2!} - \frac{s^3 T_{del}^3}{3!} + \cdots , \tag{12}$$

as follows:

$$\begin{aligned} \alpha_1 &= \frac{K_1 \left(b_1 + d_1 - T_{del}\right) - a_1 - c_1}{K_1 - 1} \\ \alpha_2 &= \frac{\begin{array}{l} K_1 \left(-T_{del} d_1 - T_{del} b_1 + d_2 + b_1 d_1 + b_2 + 0.5 T_{del}^2\right) - \\ - c_2 - a_1 c_1 - a_2 \end{array}}{\left(K_1 - 1\right)} \\ \alpha_3 &= \frac{\begin{array}{l} K_1 \left(\begin{array}{l} d_3 + b_1 d_2 + b_2 d_1 + b_3 - \frac{T_{del}^3}{6} + 0.5 T_{del}^2 \left(b_1 + d_1\right) - \\ - T_{del} \left(d_2 + b_2 + b_1 d_1\right) \end{array}\right) \\ - c_3 - a_1 c_2 - a_2 c_1 - a_3 \end{array}}{\left(K_1 - 1\right)} \\ &\vdots \\ \beta_1 &= c_1 \\ \beta_2 &= c_2 \\ \beta_3 &= c_3 \\ &\vdots \end{aligned} . \tag{13}$$

Function G_2 (9) can be simply expressed in terms of parameters α_i and β_i (10) as follows:

$$\begin{aligned} \alpha_1 = e_1,\ \alpha_2 = e_2,\ \alpha_3 = e_3, \cdots \\ \beta_1 = 0,\ \beta_2 = 0,\ \beta_3 = 0, \cdots \end{aligned} . \tag{14}$$

In order to simplify derivations, denominator den_C (3) will be chosen a-priori. Its main task is to filter out the process output noise signal, so it should be of the same or higher order (n) than the numerator:

$$den_C = \left(1 + sT_F\right)^n , \tag{15}$$

where T_F can be chosen as several times smaller than absolute values of the process time constants.

Now, the internal compensator's parameters can be calculated by equating characteristic areas (11) of G_1 and G_2 (9). In order to simplify practical realisation of the compensator, the first- and the second-order numerator (num_C) will be derived (note that it does not limit us to calculate higher-order compensators). When choosing the first-order compensator's numerator, the first two areas (11) of sub-processes G_1 (13) and G_2 (14) should be equal. The following parameters are obtained:

$$\begin{aligned} K_1 &= \frac{T_{del}\left(e_1-a_1\right)+a_1b_1-b_1e_1+e_2+e_1c_1-a_1c_1-a_2}{\begin{array}{c}b_1^2+e_2-b_2-b_1e_1+e_1c_1-b_1c_1+c_2+\\ +T_{del}\left(e_1+c_1-b_1\right)+0.5T_{del}^2\end{array}} \\ d_1 &= T_{del}-b_1+c_1+e_1+\frac{a_1-e_1}{K_1} \end{aligned} \quad . \tag{16}$$

By equating the first three areas, we get:

$$\begin{aligned} K_1 &= \frac{\begin{bmatrix} e_3-a_3+b_1a_2+c_2e_1-b_1e_2+c_1e_2-a_1c_2+a_1b_2-a_2c_1- \\ -b_2e_1-b_1c_1e_1+a_1b_1c_1-b_1^2a_1+b_1^2e_1+0.5T_{del}^2\left(e_1-a_1\right)+ \\ +T_{del}\left(c_1e_1+a_1b_1-a_1c_1-b_1e_1+e_2-a_2\right) \end{bmatrix}}{\begin{bmatrix} c_3+e_3-b_3+c_2e_1+c_1e_2+2b_2b_1-b_1e_2-c_1b_2-c_2b_1- \\ -b_2e_1+c_1b_1^2+b_1^2e_1-b_1c_1e_1-b_1^3+ \\ +T_{del}\left(c_1e_1-b_1e_1+b_1^2-b_2+e_2+c_2-c_1b_1\right)+ \\ +0.5T_{del}^2\left(c_1+e_1-b_1\right)+\frac{T_{del}^3}{6} \end{bmatrix}} \\ d_1 &= T_{del}-b_1+c_1+e_1+\frac{a_1-e_1}{K_1} \\ d_2 &= c_1e_1+c_2-b_2+e_2+\left(T_{del}-b_1\right)\left(e_1+c_1-b_1\right)+0.5T_{del}^2+ \\ &\quad +\frac{a_1c_1+a_2+T_{del}a_1-T_{del}e_1-b_1a_1+b_1e_1-e_2-e_1c_1}{K_1} \end{aligned} \tag{17}$$

Compensator gain K_C can be calculated from (7) as:

$$K_C = \frac{K_1}{K_{PR}} \, . \tag{18}$$

Note that the areas (11) can also be calculated in time-domain by integrating the process input and output signals after changing the process (10) set-point (Vrančić et al., 1999b).

Illustrative example

Let us calculate compensator's parameters for the following process transfer function (Panda, 2009; Park et al., 1998):

$$G_P = \frac{e^{-0.5s}}{(1+0.5s)(1-2s)} \, . \tag{19}$$

The desired closed-loop denominators (8) are chosen to be of the same order as the process denominator. The first one has been chosen to have the same *absolute* time constants as the process, while the second one has faster response:

$$\begin{aligned} den_{R1} &= (1+0.5s)(1+2s) \\ den_{R2} &= (1+0.7s)^2 \end{aligned} \quad . \tag{20}$$

According to expression (6), the desired closed-loop transfer functions, for both denominators, are:

$$\begin{aligned} G_{CLD1} &= \frac{1}{1-K_1}\frac{e^{-0.5s}}{(1+0.5s)(1+2s)} \\ G_{CLD2} &= \frac{1}{1-K_1}\frac{e^{-0.5s}}{(1+0.7s)^2} \end{aligned} \quad . \tag{21}$$

Note that K_1 is not known a-priori. However, it does not have any influence on stability (when $K_1 \neq 1$). The a-priori chosen denominator of the compensator (to filter out high-frequency noise) is:

$$den_C = (1+0.1s)^3 \, . \tag{22}$$

Let us now calculate the remaining compensator's parameters by using expressions (17) and (18). The compensators become:

$$\begin{aligned} G_{C1} &= \frac{1.72\left(1+0.98s+0.29s^2\right)}{(1+0.1s)^3} \\ G_{C2} &= \frac{2.32\left(1+0.95s+0.28s^2\right)}{(1+0.1s)^3} \end{aligned} \, . \tag{23}$$

Both compensators were tested in the closed-loop configuration, as shown in Fig 1 (without controller gain G_{CN}). Response on unity step-change of signal u_C is shown in Fig. 2.

It is clear that the obtained responses (solid lines) are very close to desired responses, defined by function G_{CLD} (6).

3. DESIGN OF CONTROLLER

Since the process is already stabilised by the compensator, a controller design is not very critical. Therefore, relatively simple controller structures can be used. In this paper, due to simplicity, the 2-DOF PI controller structure has been chosen:

$$U_C = \left(bK + \frac{K_i}{s}\right)R - \left(K + \frac{K_i}{s}\right)Y \, , \tag{24}$$

where K, K_i and b are proportional gain, integral gain and proportional weighting factor, respectively. Note that other types of controllers can be applied as well. A Magnitude-Optimum-Multiple-Integration (MOMI) tuning method for PI controllers has been chosen for tracking, since it usually results in a relatively fast closed-loop responses without oscillations for different types of process models (Vrančić et al., 1999a; 2001). If disturbance rejection properties are more important, a DRMO method (modified MOMI method for improving disturbance rejection performance) can be applied (Vrančić et al., 2004).

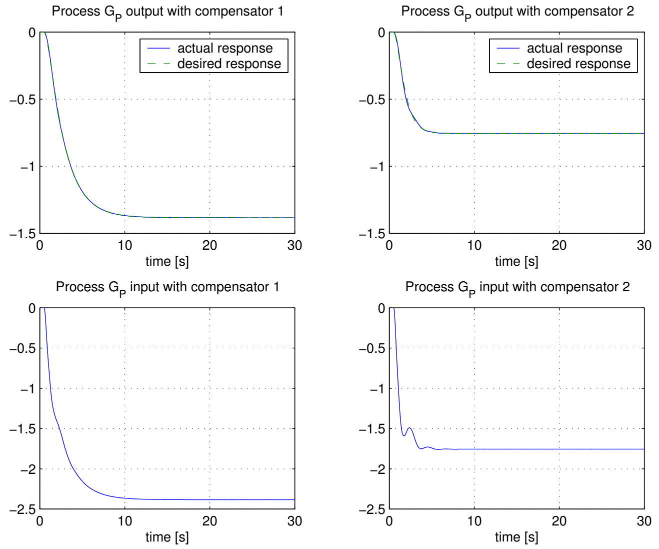


Fig. 2. Response of the inner loop when using both compensators.

The tuning rule for MOMI method is the following (see Vrančić et al., 1999a; 2001):

$$\begin{aligned} K &= \frac{A_3}{2\left(A_1A_2 - A_0A_3\right)} \\ K_i &= \frac{0.5 + A_0K}{A_1} \\ b &= 1 \end{aligned} \quad . \tag{25}$$

The tuning rule for DRMO method is (Vrančić et al., 2004):

$$\begin{aligned} &\xi_1 K^2 + 2\xi_2 K + A_3 = 0 \\ &\xi_1 = A_0^2A_3 - 2A_0A_1A_2 + A_1^3 \\ &\xi_2 = A_0A_3 - A_1A_2 \\ &K_i = \frac{\left(1 + A_0K\right)^2}{2A_1} \\ &b = 0 \end{aligned} \quad , \tag{26}$$

where K can be calculated from the second-order equation in (26). Areas A_0 to A_3 in (25) and (26) can be calculated from expression (11) if the controlled process is given by expression (10). However, note that the controlled process from controller's viewpoint is the desired closed-loop transfer function (6). Also note that parameters α_i and β_i can be expressed by equating expressions (6) and (10):

$$\begin{aligned} &K_P = \frac{K_{PR}}{1-K_1}, T_d = T_{del} \\ &\alpha_1 = e_1, \alpha_2 = e_2, \alpha_3 = e_3, \cdots, \beta_1 = b_1, \beta_2 = b_2, \beta_3 = b_3, \cdots \end{aligned} \quad . \tag{27}$$

The Matlab toolset, which performs the calculation of the compensator's and PI controller parameters for the chosen and arbitrary linear process models, is available on-line (Vrančić, 2010).

Illustrative example

Let us calculate the PI controller parameters for the same process (19) and compensators (23) as in the previous example. The PI controller is actually controlling the closed-loop transfer function (5) which is similar to desired closed-loop transfer function (21). The areas of desired transfer functions can be calculated from expressions (11) and (27) for both compensators:

$$\begin{aligned} &G_{CLD1}: A_0 = -1.38, A_1 = -4.15, A_2 = -9.17, A_3 = -18.79 \\ &G_{CLD2}: A_0 = -0.76, A_1 = -1.43, A_2 = -1.73, A_3 = -1.74 \end{aligned} \quad . \tag{28}$$

The PI controller parameters are calculated by using MOMI (25) or DRMO (26) tuning method for both compensators:

$$\begin{aligned} G_{CLD1}: \quad & K_i = -0.38, K = -0.78, b = 1 \;\; (MOMI) \\ & K_i = -0.57, K = -0.85, b = 0 \;\; (DRMO) \\ G_{CLD2}: \quad & K_i = -0.74, K = -0.74, b = 1 \;\; (MOMI) \\ & K_i = -0.89, K = -0.79, b = 0 \;\; (DRMO) \end{aligned} \quad . \tag{29}$$

The closed-loop response is given in Fig. 3. It can be seen that responses, when using MOMI method, have faster tracking responses, while DRMO method results in better disturbance rejection performance. Naturally, compensator 2 also gives faster closed-loop responses than compensator 1.

4. EXAMPLES

The proposed method will be tested on the following process models:

$$\begin{aligned} G_{P1} &= \frac{2e^{-0.3s}}{\left(1-3s\right)\left(1-s\right)} \qquad G_{P2} = \frac{4e^{-2s}}{\left(1-4s\right)} \\ G_{P3} &= \frac{e^{-s}}{\left(1-2s\right)\left(1+0.5s\right)} \qquad G_{P4} = \frac{e^{-0.4s}}{\left(1-s\right)} \end{aligned} \quad , \tag{30}$$

which were tested by some other authors (see Jacob and Chidambaram, 1996; Panda, 2010; Park et al., 1998; Prashanti and Chidambaram, 2000; Shamsuzzoha and Lee, 2008). The desired denominators are:

$$\begin{aligned} den_{R1} &= \left(1+2s\right)\left(1+s\right) \quad den_{R2} = \left(1+4s\right) \\ den_{R3} &= \left(1+s\right)^2 \qquad den_{R4} = \left(1+s\right) \end{aligned} \quad . \tag{31}$$

The calculated compensators, by using the proposed method, are the following:

$$
\begin{aligned}
G_{C1} &= \frac{0.16\left(1-18.1s-5.97s^{2}\right)}{\left(1+0.1s\right)^{3}}\\
G_{C2} &= \frac{0.39\left(1+1.105s\right)}{\left(1+0.1s\right)^{2}}\\
G_{C3} &= \frac{1.69\left(1+1.13s+0.434s^{2}\right)}{\left(1+0.1s\right)^{3}}.\\
G_{C4} &= \frac{1.64\left(1+0.228s\right)}{\left(1+0.1s\right)^{2}}
\end{aligned}
\tag{32}
$$

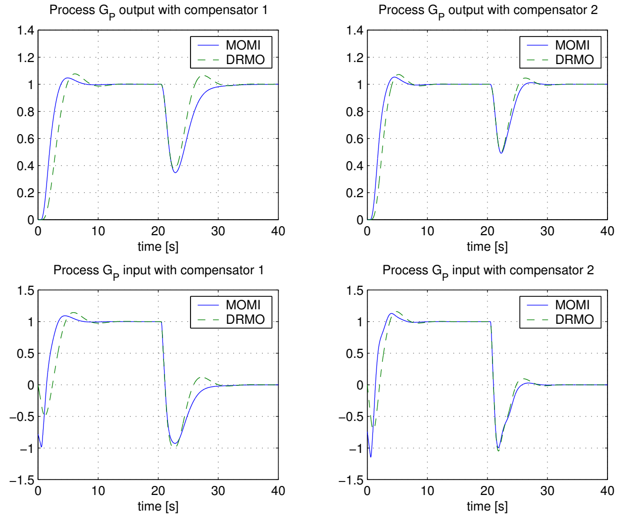


Fig. 3. Closed-loop response when using both compensators when using MOMI and DRMO method.

The calculated controller parameters, for all four process models with compensators, are given in Table 1. Note that Matlab toolset, which performs the calculation of all the parameters for the given process models, is given in Vrančić (2010).

Table 1. PI controller parameters

	MOMI			DRMO		
	K$_i$	***K***	***b***	***K$_i$***	***K***	***b***
G$_{P1}$	0.15	0.325	1	0.21	0.34	0
G$_{P2}$	-0.037	-0.148	1	-0.056	-0.166	0
G$_{P3}$	-0.227	-0.336	1	-0.266	-0.361	0
G$_{P4}$	-0.81	-0.816	1	-1.359	-0.92	0

The closed-loop responses for all four process models are given in Figs. 4-7. The difference between the desired and the actual inner closed-loop responses are relatively small for all four processes. The closed-loop responses with controller are relatively fast, without oscillations, and with relatively small overshoots, all according to the MO tuning criterion. The tracking performance is better when using MOMI method, while disturbance rejection performance is better with DRMO method.

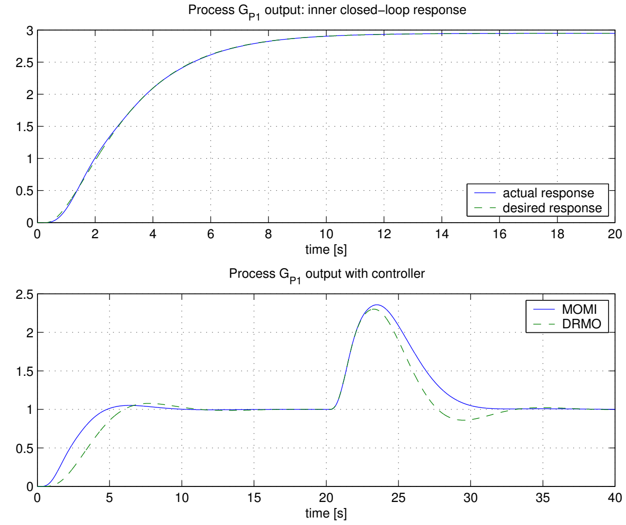


Fig. 4. Closed-loop responses of the process G$_{P1}$ when using MOMI and DRMO method.

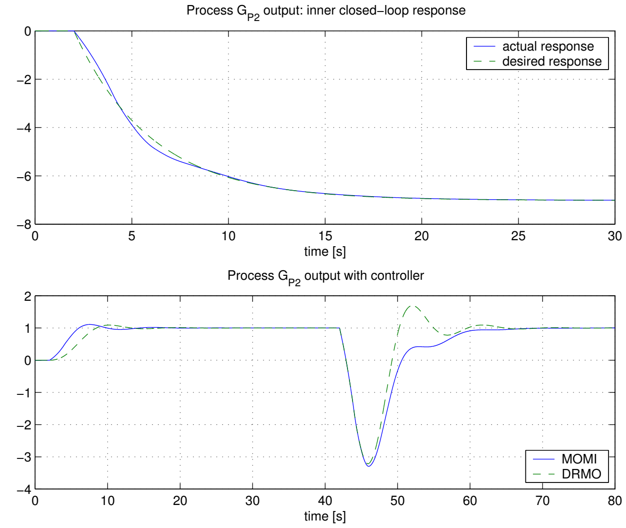


Fig. 5. Closed-loop responses of the process G$_{P2}$ when using MOMI and DRMO method.

5. CONCLUSIONS

Controller design is divided into two stages. The first stage is design of inner compensator by means of equating characteristic areas of the actual and desired inner closed-loop transfer function. The comparison of both responses in five examples confirms the efficiency the compensator.

The second stage is design of outer 2-DOF PI controller by applying MOMI or DRMO tuning method. According to all five examples, the proposed approach resulted in a relatively fast responses without oscillations.

The advantages of the proposed method are that it is not limited to the first- or the second-order processes models. Moreover, the method can be extended to higher order compensators or different controller structures (e.g. PID controllers or Smith predictors).

Disadvantage of the proposed method is that it requires, similar to other methods, the a-priori definition of desired closed-loop transfer function. In our case, the desired closed-loop time constants have been chosen to be the same or slightly faster to absolute values of process time constants.

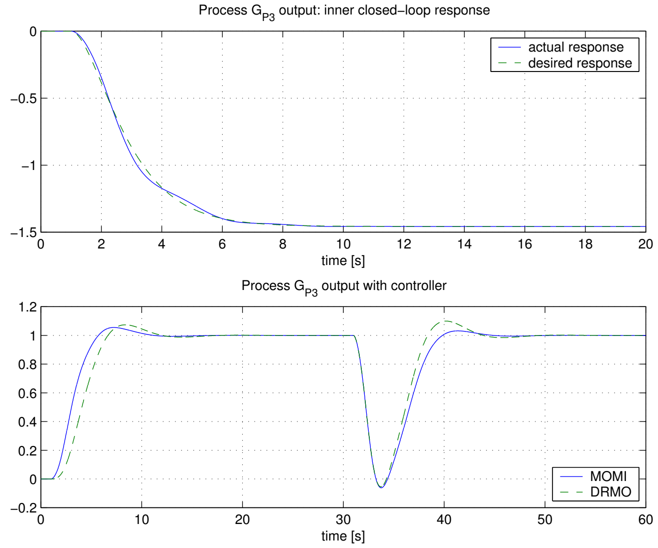


Fig. 6. Closed-loop responses of the process G_{P3} when using MOMI and DRMO method.

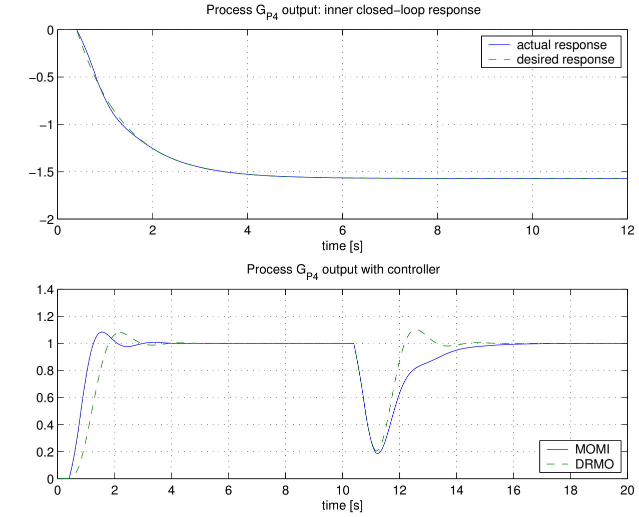


Fig. 7. Closed-loop responses of the process G_{P4} when using MOMI and DRMO method.

In our further work we will investigate robustness of the proposed tuning approach.

ACKNOWLEDGEMENT

This work has been financially supported by Slovene Research Agency. This support is gratefully acknowledged.

REFERENCES

Jacob, E. F., and Chidambaram M. (1996). Design of controllers for unstable first-order plus time delay systems. *Computers chem. Engineering*, Vol. 20, No. 5, pp. 579-584.

Lee S. C., Wang Q-G., and Nguyen L. B. (2010). Stabilizing control for a class of delay unstable processes. *ISA Transactions*, Vol. 49, pp.318-325, 2010.

Panda, R. C. (2009). Synthesis of PID controller for unstable and integrating processes. *Chemical Engineering Science*, Vol. 64, pp. 2807-2816.

Park, J. H., Sung, S. W., and Lee, I-B. An enhanced PID control Strategy for Unstable Processes. *Automatica*, Vol. 34, No. 6, pp. 751-756, 1998.

Prashanti G., and Chidambaram, M. (2000). Set-point weighted PID controllers for unstable systems. *Journal of The Franklin Institute*, Vol. 337. pp. 201-215.

Rake, H. (1987). Identification: transient and frequency response methods. In: M. G. Singh (Ed.), *Systems and control encyclopedia; theory, technology, applications*, Oxford: Pergamon Press.

Shamsuzzoha, M., and Lee, M. (2008). Analytical design of enhanced PID filter controller for integrating and first order unstable processes with time delay. *Chemical Engineering Science*, Vol. 63, pp. 2717-2731.

Vrančić, D., Peng, Y., and Strmčnik, S. (1999a). A new PID controller tuning method based on multiple integrations. *Control Engineering Practice*, Vol. 7, pp. 623-633.

Vrančić, D., Juričić, Đ., Strmčnik, S., and Hanus, R. (1999b). Closed-loop tuning of the PID controller by using MOMI method. *Proc. Of the American Control Conference*, San Diego, pp. 3352-3356.

Vrančić, D., Strmčnik, S., and Juričić, Đ. (2001). A magnitude optimum multiple integration tuning method for filtered PID controller, *Automatica*, Vol. 37, pp. 1473-1479.

Vrančić, D., Strmčnik, S., and Kocijan, J. (2004). Improving disturbance rejection of PI controller by means of the magnitude optimum. *ISA Transactions*, Vol. 43, pp. 73-84.

Vrančić, D. (2010). Matlab Toolset for Unstable Processes, Available on
http://dsc.ijs.si/damir.vrancic/tools.html.

Equivalent Representations of Bounded Real Lemma

D. Krokavec and A. Filasová

Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Košice, Slovakia
fax: ++ 421 55 625 3574
e-mail: dusan.krokavec@tuke.sk, anna.filasova@tuke.sk

Abstract: The paper concerns the problem of the bounded real lemma for linear continuous-time systems. Using free weighting matrices to express the relationship between the terms of the system state equation a modified equivalent LMI approach to bounded-real-lemma representation is presented. Immediate extension to design method of a memory-free feedback controller, which performs H_∞ properties of the closed-loop system, is formulated as a feasibility problem and expressed over a set of LMIs. Numerical example is included to illustrate the feasibility and properties of the proposed representations.

Keywords: Bounded real lemma, H_∞ performance, continuous-time systems, linear matrix inequality representation.

1. INTRODUCTION

Over the past decade, H_∞ theory seems to be one of the most sophisticated frameworks for robust control system design. Based on concept of quadratic stability which attempts to find a quadratic Lyapunov function (LF), H_∞ norm computation problem is transferred into a standard linear matrix inequality (LMI) optimization task, which includes bounded real lemma (BRL) formulation (Hermann et al. (2007), Veselý and Rosinová (2009), Wu et al. (2010)). A number of more or less conservative analysis methods are presented to assess robust stability for linear systems using a fixed Lyapunov function.

The first version of the BRL presents simple conditions under which a transfer function is contractive on the imaginary axis. Using it, it was possible to determine the H_∞ norm of a transfer function, and the BRL became a significant element to shown and prove that the existence of feedback controllers (that results in a closed loop transfer matrix having the H_∞ norm less than a given upper bound), is equivalent to the existence of solutions of certain LMIs (Boyd *et al.* (1994), Filasová et al. (2010)). Linear matrix inequality approach based on convex optimization algorithms is extensively applied to solve the above mentioned problem (Jia (2003), Pipeleers et al. (2009)) since it can be solved numerically efficiently by using developed interior-point algorithm.

In this paper, equivalent LMI representations of BRL for linear continuous-time systems are introduced. Motivated by the underlying ideas in Filasová and Krokavec (2009), Wu and Duan (2006), and Xie (2008) a simple technique for the BRL representation of linear systems is presented, and used modifications are explained in a context. The proposed LMI representations are proven to be necessary and sufficient and their extensions to state feedback controller design, performing system H_∞ properties is immediate. Translating into LMI framework the closed-loop system stability is characterized in the terms of convex LMIs.

2. PROBLEM DESCRIPTION

Through this paper the task is concerned with the computation of a state feedback $\boldsymbol{u}(t)$, which control the linear dynamic system given by the set of equations

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{D}\boldsymbol{u}(t) \tag{2}$$

where $\boldsymbol{q}(t) \in I\!R^n$, $\boldsymbol{u}(t) \in I\!R^r$, and $\boldsymbol{y}(t) \in I\!R^m$ are vectors of the state, input and measurable output variables, respectively, nominal system matrices $\boldsymbol{A} \in I\!R^{n\times n}$, $\boldsymbol{B} \in I\!R^{n\times r}$, $\boldsymbol{C} \in I\!R^{m\times n}$ and $\boldsymbol{D} \in I\!R^{m\times r}$ are real matrices.

Problem of the interest is to design asymptotically stable closed-loop system with the linear memoryless state feedback controller of the form

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{q}(t) \tag{3}$$

where matrix $\boldsymbol{K} \in I\!R^{r\times n}$ is a gain matrix.

3. BASIC PRELIMINARIES

Proposition 1. (Bounded real lemma) System (1), (2) is asymptotically stable if there exist a symmetric positive definite matrix $\boldsymbol{P} > 0$ and a positive scalar $\gamma > 0$ such that

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} & \boldsymbol{P}\boldsymbol{B} & \boldsymbol{C}^T \\ * & -\gamma^2\boldsymbol{I}_r & \boldsymbol{D}^T \\ * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{4}$$

where $\boldsymbol{I}_r \in I\!R^{r\times r}$, $\boldsymbol{I}_m \in I\!R^{m\times m}$ are identity matrices, respectively, Hereafter, $*$ denotes the symmetric item in a symmetric matrix.

Proof. (see. e.g. Krokavec and Filasová (2008)) Defining Lyapunov function as follows

$$v(\boldsymbol{q}(t)) = \boldsymbol{q}^T(t)\boldsymbol{P}\boldsymbol{q}(t) + \\ + \int_0^t (\boldsymbol{y}^T(r)\boldsymbol{y}(r) - \gamma^2\boldsymbol{u}^T(r)\boldsymbol{u}(r))\mathrm{d}r > 0 \tag{5}$$

where $\boldsymbol{P} = \boldsymbol{P}^T > 0$, $\boldsymbol{P} \in I\!R^{n\times n}$, $\gamma > 0 \in I\!R$, and evaluating the derivative of $v(\boldsymbol{q}(t))$ with respect to t along a system trajectory then it yields

$$\dot{v}(\boldsymbol{q}(t)) = \dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t) + \\ + \boldsymbol{y}^T(t)\boldsymbol{y}(t) - \gamma^2\boldsymbol{u}^T(t)\boldsymbol{u}(t) < 0 \tag{6}$$

Thus, substituting (1), (2) into (6) gives

$$\dot{v}(\boldsymbol{q}(t)) = (\boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t))^T\boldsymbol{P}\boldsymbol{q}(t) + \\ + \boldsymbol{q}^T(t)\boldsymbol{P}(\boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t)) - \gamma\boldsymbol{u}^T(t)\boldsymbol{u}(t) + \\ + (\boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{D}\boldsymbol{u}(t))^T(\boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{D}\boldsymbol{u}(t)) < 0 \tag{7}$$

and with the next notation

$$\boldsymbol{q}_c^T(t) = \begin{bmatrix} \boldsymbol{q}^T(t) & \boldsymbol{u}^T(t) \end{bmatrix} \tag{8}$$

it is obtained

$$\dot{v}(\boldsymbol{q}(t)) = \boldsymbol{q}_c^T(t)\boldsymbol{P}_c\boldsymbol{q}_c(t) < 0 \tag{9}$$

where

$$\boldsymbol{P}_c = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} & \boldsymbol{P}\boldsymbol{B} \\ * & -\gamma^2\boldsymbol{I}_r \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{C}^T\boldsymbol{D} \\ * & \boldsymbol{D}^T\boldsymbol{D} \end{bmatrix} < 0 \tag{10}$$

Since

$$\begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{C}^T\boldsymbol{D} \\ * & \boldsymbol{D}^T\boldsymbol{D} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}^T \\ \boldsymbol{D}^T \end{bmatrix} [\, \boldsymbol{C} \;\; \boldsymbol{D} \,] \geq 0 \tag{11}$$

Schur complement property implies

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}^T \\ * & \boldsymbol{0} & \boldsymbol{D}^T \\ * & * & -\boldsymbol{I}_m \end{bmatrix} \geq 0 \tag{12}$$

and using (12) the LMI condition (10) can be written compactly as (4). This concludes the proof. ■

4. IMPROVED BRL REPRESENTATION

Theorem 1. System (1), (2) is asymptotically stable if there exist a symmetric positive definite matrix $\boldsymbol{P} > 0$, $\boldsymbol{P} \in I\!R^{n\times n}$, matrices $\boldsymbol{S}_1, \boldsymbol{S}_2 \in I\!R^{n\times n}$, and a positive scalar $\gamma > 0$, $\gamma \in I\!R$ such that

$$\begin{bmatrix} -\boldsymbol{S}_1\boldsymbol{A} - \boldsymbol{A}^T\boldsymbol{S}_1^T & -\boldsymbol{S}_1\boldsymbol{B} & \boldsymbol{P} + \boldsymbol{S}_1 - \boldsymbol{A}^T\boldsymbol{S}_2^T & \boldsymbol{C}^T \\ * & -\gamma^2\boldsymbol{I}_r & -\boldsymbol{B}^T\boldsymbol{S}_2^T & \boldsymbol{D}^T \\ * & * & \boldsymbol{S}_2 + \boldsymbol{S}_2^T & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{13}$$

Proof. Since (1) implies

$$\dot{\boldsymbol{q}}(t) - \boldsymbol{A}\boldsymbol{q}(t) - \boldsymbol{B}\boldsymbol{u}(t) = \boldsymbol{0} \tag{14}$$

then with arbitrary square matrices $\boldsymbol{S}_1, \boldsymbol{S}_2 \in I\!R^{n\times n}$ it yields

$$\big(\boldsymbol{q}^T(t)\boldsymbol{S}_1 + \dot{\boldsymbol{q}}^T(t)\boldsymbol{S}_2\big)\big(\dot{\boldsymbol{q}}(t) - \boldsymbol{A}\boldsymbol{q}(t) - \boldsymbol{B}\boldsymbol{u}(t)\big) = 0 \tag{15}$$

Thus, adding (15), as well as its transposition to (6) and substituting (2) it can be written

$$\dot{v}(\boldsymbol{q}(t)) = -\gamma\boldsymbol{u}^T(t)\boldsymbol{u}(t) + \dot{\boldsymbol{q}}^T(t)\boldsymbol{P}\boldsymbol{q}(t) + \boldsymbol{q}^T(t)\boldsymbol{P}\dot{\boldsymbol{q}}(t) + \\ + (\boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{D}\boldsymbol{u}(t))^T(\boldsymbol{C}\boldsymbol{q}(t) + \boldsymbol{D}\boldsymbol{u}(t)) + \\ + \big(\dot{\boldsymbol{q}}(t) - \boldsymbol{A}\boldsymbol{q}(t) - \boldsymbol{B}\boldsymbol{u}(t)\big)^T\big(\boldsymbol{S}_1^T\boldsymbol{q}(t) + \boldsymbol{S}_2^T\dot{\boldsymbol{q}}(t)\big) + \\ + (\boldsymbol{q}^T(t)\boldsymbol{S}_1 + \dot{\boldsymbol{q}}^T(t)\boldsymbol{S}_2)(\dot{\boldsymbol{q}}(t) - \boldsymbol{A}\boldsymbol{q}(t) - \boldsymbol{B}\boldsymbol{u}(t)) < 0 \tag{16}$$

and using the notation

$$\boldsymbol{q}_c^T(t) = \begin{bmatrix} \boldsymbol{q}^T(t) & \boldsymbol{u}^T(t) & \dot{\boldsymbol{q}}^T(t) \end{bmatrix} \tag{17}$$

it can be obtained

$$\dot{v}(\boldsymbol{q}(t)) = \boldsymbol{q}_c^T(t)\boldsymbol{P}_c^\circ\boldsymbol{q}_c(t) < 0 \tag{18}$$

where

$$\boldsymbol{P}_c^\circ = \begin{bmatrix} \boldsymbol{C}^T\boldsymbol{C} & \boldsymbol{C}^T\boldsymbol{D} & \boldsymbol{0} \\ * & \boldsymbol{D}^T\boldsymbol{D} & \boldsymbol{0} \\ * & * & \boldsymbol{0} \end{bmatrix} + \\ + \begin{bmatrix} -\boldsymbol{S}_1\boldsymbol{A} - \boldsymbol{A}^T\boldsymbol{S}_1^T & -\boldsymbol{S}_1\boldsymbol{B} & \boldsymbol{P} + \boldsymbol{S}_1 - \boldsymbol{A}^T\boldsymbol{S}_2^T \\ * & -\gamma^2\boldsymbol{I}_r & -\boldsymbol{B}^T\boldsymbol{S}_2^T \\ * & * & \boldsymbol{S}_2 + \boldsymbol{S}_2^T \end{bmatrix} < 0 \tag{19}$$

Thus, analogously using (11), (12) the inequality (19) can be written compactly as (13). This concludes the proof. ■

Remark 1. Setting $\boldsymbol{S}_1 = -\boldsymbol{P}$ then (13) is transformed in

$$\begin{bmatrix} \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} & \boldsymbol{P}\boldsymbol{B} & -\boldsymbol{A}^T\boldsymbol{S}_2^T & \boldsymbol{C}^T \\ * & -\gamma^2\boldsymbol{I}_r & -\boldsymbol{B}^T\boldsymbol{S}_2^T & \boldsymbol{D}^T \\ * & * & \boldsymbol{S}_2 + \boldsymbol{S}_2^T & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{20}$$

Thus, inserting $\boldsymbol{S}_2 = -\delta\boldsymbol{I}$, where $\delta > 0$, $\delta \in I\!R$ gives

$$\begin{bmatrix} \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} & \boldsymbol{P}\boldsymbol{B} & \delta\boldsymbol{A}^T & \boldsymbol{C}^T \\ * & -\gamma\boldsymbol{I}_r & \delta\boldsymbol{B}^T & \boldsymbol{D}^T \\ * & * & -2\delta\boldsymbol{I}_n & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{21}$$

$$\begin{bmatrix} \boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{P} & \boldsymbol{P}\boldsymbol{B} & \boldsymbol{A}^T & \boldsymbol{C}^T \\ * & -\gamma\boldsymbol{I}_r & \boldsymbol{B}^T & \boldsymbol{D}^T \\ * & * & -2\delta^{-1}\boldsymbol{I}_n & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \tag{22}$$

respectively. Then (22) can be written as

$$\begin{bmatrix} \boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} & \boldsymbol{P}\boldsymbol{B} & \boldsymbol{C}^T \\ * & -\gamma\boldsymbol{I}_r & \boldsymbol{D}^T \\ * & * & -\boldsymbol{I}_m \end{bmatrix} + \\ + 0.5\,\delta \begin{bmatrix} \boldsymbol{A}^T \\ \boldsymbol{B}^T \\ \boldsymbol{0} \end{bmatrix} [\boldsymbol{A} \;\; \boldsymbol{B} \;\; \boldsymbol{0}] < 0 \tag{23}$$

Choosing δ as a sufficiently small positive scalar satisfying the condition

$$0 < \delta < 2\frac{\lambda_1}{\lambda_2} \tag{24}$$

$$\lambda_1 = \lambda_{min}\left\{ - \begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} \right\} \tag{25}$$

$$\lambda_2 = \lambda_{max}\left\{ \begin{bmatrix} A^T A & A^T B & 0 \\ B^T A & B^T B & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\} \tag{26}$$

(21) be negative definite for a feasible P of (4).

Corollary 1. Setting $S_1 = -P$, and $S_2 = \delta I$, where $0 < \delta \in \mathbb{R}$ then (20)-(22) implies

$$\begin{bmatrix} A^T P + PA & PB & C^T \\ * & -\gamma I_r & D^T \\ * & * & -I_m \end{bmatrix} - \\ -0.5\,\delta \begin{bmatrix} -A^T \\ -B^T \\ 0 \end{bmatrix} [-A \ -B \ 0] < 0 \tag{27}$$

and a feasible solution P of (4) is also a feasible solution of (27) for all $\delta > 0$, $\delta \in \mathbb{R}$.

Theorem 2. System (1), (2) is asymptotically stable if there exist a symmetric positive definite matrix $P > 0$, $P \in \mathbb{R}^{n \times n}$, matrices S_1, $S_2 \in \mathbb{R}^{n\times n}$, and a positive scalar $\gamma > 0$, $\gamma \in \mathbb{R}$ such that

$$\begin{bmatrix} PA + A^T P & PB & P + S_1 + A^T S_2 & C^T \\ * & -\gamma^2 I_r & B^T S_2 & D^T \\ * & * & S_2 + S_2^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{28}$$

Proof. Defining the congruence transform matrix

$$T_1 = \begin{bmatrix} I & & & \\ & I & & \\ A & B & I & \\ & & & I \end{bmatrix} \tag{29}$$

and multiplying right-hand side of (13) by T_1 and left-hand side of (13) by T_1^T then after tedious calculation (28) is obtained. This concludes the proof. ■

Remark 2. Setting $S_1 = -P$, $S_2 = -\delta P$ then (28) leads to

$$\begin{bmatrix} PA + A^T P & PB & -\delta A^T P & C^T \\ * & -\gamma^2 I_r & -\delta B^T P & D^T \\ * & * & -2\delta P & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{30}$$

$$\begin{bmatrix} PA + A^T P & PB & -A^T P & C^T \\ * & -\gamma^2 I_r & -B^T P & D^T \\ * & * & -2\delta^{-1} P & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{31}$$

respectively, and using Schur complement property then (31) can be rewritten as

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} + \\ + \frac{\delta}{2} \begin{bmatrix} -A^T P \\ -B^T P \\ 0 \end{bmatrix} P^{-1} [-PA \ -PA \ \ 0] < 0 \tag{32}$$

$$\begin{bmatrix} PA + A^T P & PB & C^T \\ * & -\gamma^2 I_r & D^T \\ * & * & -I_m \end{bmatrix} + \\ + \frac{\delta}{2} \begin{bmatrix} A^T PA & A^T PB & 0 \\ B^T PA & B^T PB & 0 \\ 0 & 0 & 0 \end{bmatrix} < 0 \tag{33}$$

respectively. Choosing δ satisfying (24), then with (25) and

$$\lambda_2 = \lambda_{max}\left\{ \begin{bmatrix} A^T PA & A^T PB & 0 \\ B^T PA & B^T PB & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\} \tag{34}$$

(31) be negative definite for a feasible P of (4). This concludes the proof. ■

Corollary 2. Considering (32), (33) it is evident that the inequality

$$\begin{bmatrix} PA + A^T P & PB & A^T P & C^T \\ * & -\gamma^2 I_r & B^T P & D^T \\ * & * & -2\delta^{-1} P & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{35}$$

and (31) are equivalent.

5. CONTROL LAW PARAMETER DESIGN

Theorem 3. Closed-loop system (1), (2), (3) is stable if there exists a symmetric positive definite matrix $X > 0$, $X \in \mathbb{R}^{n\times n}$, a regular square matrix $Z \in \mathbb{R}^{n\times n}$, a matrix $Y \in \mathbb{R}^{r\times n}$, and a scalar $\gamma > 0$, $\gamma \in \mathbb{R}$ such that

$$X = X^T > 0, \quad \gamma > 0 \tag{36}$$

$$\begin{bmatrix} \Pi_{11} & B & XA^T - Y^T B^T & XC^T - Y^T D^T \\ * & -\gamma^2 I_r & B^T & D^T \\ * & * & Z + Z^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{37}$$

$$\Pi_{11} = AX + XA^T - BY - Y^T B^T \tag{38}$$

The control law gain matrix is given as

$$K = YX^{-1} \tag{39}$$

Proof. Setting $S_1 = -P$ then (28) implies

$$\begin{bmatrix} PA + A^T P & PB & A^T S_2 & C^T \\ * & -\gamma^2 I_r & B^T S_2 & D^T \\ * & * & S_2 + S_2^T & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{40}$$

Supposing that $\det(S_2) \neq 0$ then it can be defined the congruence transform matrix

$$T_2 = \text{diag}\left[P^{-1} \ I_r \ S_2^{-1} \ I_m \right] \tag{41}$$

and pre-multiplying right-hand side of (40) by T_2, and left-hand side of (40) by T_2^T leads to

$$\begin{bmatrix} AP^{-1} + P^{-1} A^T & B & P^{-1} A^T & P^{-1} C^T \\ * & -\gamma^2 I_r & B^T & D^T \\ * & * & S_2^{-1} + S_2^{-T} & 0 \\ * & * & * & -I_m \end{bmatrix} < 0 \tag{42}$$

Thus, denoting

$$P^{-1} = X, \qquad S_2^{-1} = Z \tag{43}$$

(42) can be written as

$$\begin{bmatrix} \boldsymbol{AX}+\boldsymbol{XA}^T & \boldsymbol{B} & \boldsymbol{XA}^T & \boldsymbol{XC}^T \\ * & -\gamma^2\boldsymbol{I}_r & \boldsymbol{B}^T & \boldsymbol{D}^T \\ * & * & \boldsymbol{Z}+\boldsymbol{Z}^T & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \qquad (44)$$

Inserting $\boldsymbol{A} \leftarrow \boldsymbol{A}_c = \boldsymbol{A}-\boldsymbol{BK}$, and $\boldsymbol{C} \leftarrow \boldsymbol{C}_c = \boldsymbol{C}-\boldsymbol{DK}$ it yields

$$\begin{bmatrix} \boldsymbol{\Pi}_{11} & \boldsymbol{B} & \boldsymbol{X}(\boldsymbol{A}^T-\boldsymbol{K}^T\boldsymbol{B}^T) & \boldsymbol{X}(\boldsymbol{C}^T-\boldsymbol{K}^T\boldsymbol{D}^T) \\ * & -\gamma^2\boldsymbol{I}_r & \boldsymbol{B}^T & \boldsymbol{D}^T \\ * & * & \boldsymbol{Z}+\boldsymbol{Z}^T & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \; (45)$$

where

$$\boldsymbol{\Pi}_{11} = \boldsymbol{AX}+\boldsymbol{XA}^T-\boldsymbol{BKX}-\boldsymbol{XK}^T\boldsymbol{B}^T \qquad (46)$$

and with

$$\boldsymbol{Y} = \boldsymbol{KX} \qquad (47)$$

(45), (46) implies (37), (38), respectively. This concludes the proof. ■

Remark 3. Setting $\boldsymbol{Z} = -\delta\boldsymbol{X}$ then with $\boldsymbol{D} = \boldsymbol{0}$ the control law design condition (36)-(38) can be rewritten as

$$\boldsymbol{X} = \boldsymbol{X}^T > 0, \quad \delta > 0 \qquad (48)$$

$$\begin{bmatrix} \boldsymbol{\Pi}_{11} & \boldsymbol{B} & \boldsymbol{XA}^T-\boldsymbol{Y}^T\boldsymbol{B}^T & \boldsymbol{XC}^T \\ * & -\gamma^2\boldsymbol{I}_r & \boldsymbol{B}^T & \boldsymbol{0} \\ * & * & -2\delta\boldsymbol{I}_n & \boldsymbol{0} \\ * & * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \qquad (49)$$

$$\boldsymbol{\Pi}_{11} = \boldsymbol{AX}+\boldsymbol{XA}^T-\boldsymbol{BY}-\boldsymbol{Y}^T\boldsymbol{B}^T \qquad (50)$$

where feasible $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{\delta}$ implies gain matrix parameter (39).

Therefore, it is evident that the design standard form of BRL is

$$\begin{bmatrix} \boldsymbol{\Pi}_{11} & \boldsymbol{B} & \boldsymbol{X}(\boldsymbol{C}^T-\boldsymbol{K}^T\boldsymbol{D}^T) \\ * & -\gamma^2\boldsymbol{I}_r & \boldsymbol{0} \\ * & * & -\boldsymbol{I}_m \end{bmatrix} < 0 \qquad (51)$$

Note, other nontrivial solutions can be obtained using different setting of $\boldsymbol{S}_l,\ l = 1, 2$.

6. ILLUSTRATIVE EXAMPLE

The approaches given above are illustrated by the numerical example where the parameters of (1), (2) are

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -9 & -5 \end{bmatrix}, \ \boldsymbol{B} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 1 & 5 \end{bmatrix}, \ \boldsymbol{C}^T = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix}$$

Solving (48), (49) with respect to LMI matrix variables $\boldsymbol{X}$, $\boldsymbol{Y}$, γ, and δ using SeDuMi (Self-Dual-Minimization) package for Matlab (Peaucelle et al. (1994)) given task was feasible with

$$\boldsymbol{X} = \begin{bmatrix} 3.7160 & -2.6784 & 1.2147 \\ -2.6784 & 3.0184 & -1.8970 \\ 1.2147 & -1.8970 & 3.2896 \end{bmatrix}$$

$$\boldsymbol{Y} = \begin{bmatrix} 0.8937 & 2.1673 & -1.4078 \\ -0.0801 & -0.0207 & 0.5383 \end{bmatrix}$$

$$\gamma = 11.0242, \qquad \delta = 6.7040$$

and results the control system parameters

$$\boldsymbol{K} = \begin{bmatrix} 2.2731 & 3.0405 & 0.4860 \\ 0.0152 & 0.1662 & 0.2538 \end{bmatrix}$$

$$\rho(\boldsymbol{A}_c) = \{-0.9398, \ -3.1252, \ -11.2561\}$$

It is evident, that the eigenvalues spectrum $\rho(\boldsymbol{A}_c)$ of the closed control loop is stable.

Solving (48), (51) with respect to LMI matrix variables $\boldsymbol{X}$, $\boldsymbol{Y}$, and γ given task was feasible, too. Obtained LMI variables were

$$\boldsymbol{X} = \begin{bmatrix} 2.7637 & -1.7983 & 0.6386 \\ -1.7983 & 2.2479 & -1.2081 \\ 0.6386 & -1.2081 & 3.0925 \end{bmatrix}$$

$$\boldsymbol{Y} = \begin{bmatrix} 0.9127 & 1.6581 & -0.8163 \\ 0.2802 & 0.1304 & -0.2269 \end{bmatrix}$$

$$\gamma = 6.9412$$

and implies

$$\boldsymbol{K} = \begin{bmatrix} 1.7557 & 2.2852 & 0.2662 \\ 0.2837 & 0.2709 & -0.0261 \end{bmatrix}$$

$$\rho(\boldsymbol{A}_c) = \{-0.8968, \ -5.8435 \pm 1.7282\,\mathrm{i}\}$$

It is evident, that performance γ is less then one obtained with respect to (49) but this fetches worst dynamic properties.

It also should be noted, the cost value γ will not be a monotonously decreasing function with the decreasing of δ, if δ is chosen.

7. CONCLUDING REMARKS

This paper describes a simple technique for equi-valent BRL representation and its application to the H_∞ control of linear systems. Standard criterion is extended for a system with constant coefficient matrices employing free weighting matrices to take the relationship between the terms of the system equation into account in the structure of BRL. The method is further extended to the design of an H_∞ state-feedback controller. Numerical example demonstrates that principles described in this paper are effective, although some computational complexity is increases.

The advantage of this approach is that in Theorem 1 Lyapunov matrix $\boldsymbol{P}$ is separated from $\boldsymbol{A}$, $\boldsymbol{B}^T$, $\boldsymbol{C}$, and $\boldsymbol{D}^T$, i.e. there are no terms containing the product of $\boldsymbol{P}$ and any of them. This enables a new robust BRL to be derived for a system with polytopic uncertainties by using a parameter-dependent Lyapunov function, and to deal with linear systems with parametric uncertainties. It seems to be a useful extension to other control performance synthesis problems, too.

ACKNOWLEDGMENTS

The work presented in this paper was supported by VEGA, Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/0256/11. This support is very gratefully acknowledged.

REFERENCES

D. Boyd, L. El Ghaoui, E. Peron, and V. Bala-krishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM Society for Industrial and Applied Mathematics, Philadelphia, 1994.

A. Filasová and D. Krokavec. Global asymptotically stable control design for time-delay systems. *AT&P Journal Plus*, 2, 89-92, 2009.

A. Filasová, D. Gontkovič, and D. Krokavec. Output feedback control design using unified algebraic approach, In *Proceedings of the 8^{th} International Symposium on Applied Machine Intelligence and Informatics SAMI 2010*, Herľany, Slovakia, 259-262, 2010.

G. Herrmann, M.C. Turner, and I. Postlethwaite. Linear matrix inequalities in control, In *Mathematical Methods for Robust and Nonlinear Control*, Springer–Verlag, Berlin, 123-142, 2007.

Y. Jia. Alternative proofs for improved LMI representations for the analysis and the design of continuous-time systems with poolytopic type uncertainty: A predictive approach, *IEEE Transactions on Automatic Control*, 48:8, 1413-1416, 2003.

D. Krokavec and A. Filasová. *Discrete-Time Systems*, Elfa, Košice, 2008. (in Slovak)

D. Peaucelle, D. Henrion, Y. Labit, and K. Taitz. *User's Guide for SeDuMi Interface 1.04*, LAAS-CNRS, Toulouse, 2002.

G. Pipeleers, B. Demeulenaerea, J. Sweversa, and L. Vandenbergheb. Extended LMI characterizations for stability and performance of linear systems, *Systems & Control Letters*, 58, 510-518, 2009.

V. Veselý and A. Rosinová. Robust output model predictive control design. BMI approach, *International Journal of Innovative Computing, Information and Control*, 5:4, 1115-1123, 2009.

A.I. Wu and G.R. Duan. Enhanced LMI Representations for H_2 Performance of Polytopic Uncertain Systems: Continuous-time Case, *International Journal of Automation and Computing*, 3, 304-308, 2006.

M. Wu, Y. He, and J.H. She. *Stability Analysis and Robust Control of Time-Delay Systems*, Science Press, Beijing and Springer–Verlag, Berlin, 2010.

W. Xie. An equivalent LMI Representation of Bounded Real Lemma for Continous-Time Systems, Journal of Inequalities and Applications, 5, 2010.

MATLAB TOOLBOX FOR PWA IDENTIFICATION OF NONLINEAR SYSTEMS

J. Stevek [*] **S. Kozak** [**]

[*] *Slovak University of Technology, Faculty of Informatics and Information Technologies, Bratislava, Slovakia (e-mail: stevek@fiit.stuba.sk)*
[**] *Slovak University of Technology, Faculty of Electrical Engineering and Information Technology, Bratislava, Slovakia (e-mail: stefan.kozak@stuba.sk).*

Abstract: This paper is dedicated to issue of approximation of nonlinear functions and nonlinear dynamical systems by Piecewise Affine (PWA) linear model. The article presents new identification Matlab toolbox for modelling and simulation of nonlinear systems. Functions of the toolbox together with GUI application simplified and accelerates identification of so called PWA OAF model. Identification of nonlinear systems is based on novel method of PWA modelling by generalized Fourier series. The approach provides identification of nonlinear functions of an arbitrary number of variables and identification of nonlinear dynamical systems in ARX model structure fashion from input-output data.

Keywords: PWA systems, Generalized Fourier series, Matlab toolbox, Chebyshev polynomial, PWA identification.

1. INTRODUCTION

In the recent research many methods were developed for modelling of hybrid systems and general nonlinear functions at all (Roll et al., 2004; Ferrari-Trecate, 2005; Julian et al., 1999). Many model structures were developed for hybrid systems and nonlinear systems. Much attention is dedicated to system modeling in MLD (Mixed Integer Dynamical) form (Bemporad and Morari, 1999) and PWA (Piecewise Affine). In (Bemporad et al., 2000), the formal equivalence between MLD systems and PWA systems is established and also effective algorithms were developed for transformation from one model structure to another (Villa et al., 2004; Bemporad, 2002). In (Heemels et al., 2001ab), the equivalence between the following five classes of hybrid systems is, under certain conditions, established: MLD systems, Linear Complementarity (LC) systems, Extended Linear Complementarity (ELC) systems, PWA systems and Max-Min-Plus-Scaling (MMPS) systems. The important result of these equivalences is that derived theoretical properties and tools can easily be transferred from one class to another.

In this paper we present an effective tool for modeling of nonlinear systems by PWA using novel approach based on generalized Fourier series (Kozak and Stevek, 2010). This approach belongs to black-box identification methods of general nonlinear models (Sjöberg et al., 1995).

We use methodology of generalized Fourier series with orthogonal polynomials. In (Leondes, 1997), orthogonal polynomials were used as activation functions for special case of neural network with one hidden layer - Orthogonal Activation Function based Neural Network (OAF NN). For this type of neural network online and off-line training algorithm has been defined with fast convergence properties. After simple modification of OAF NN it is possible to use this technique for PWA approximation of a common nonlinear system.

The paper is divided in six sections. First, we formulate the identification and linearization problem of nonlinear function. Next, we present modeling of nonlinear process by OAF NN, topology of the fourier series (PWA OAF NN) and network transformation to state space PWA form. In Section 3 PWA OAF identification toolbox is presented on three case studies. In Section 3.3 is identified nonlinear dynamical system from input-output data and designed explicit mpc control law.

2. PROBLEM FORMULATION

PWA linear approximation of hybrid systems depends on defining guardlines of the PWA mappping. If guardlines are known, the problem of identifying PWA systems can easily be solved using standard techniques for linear systems (Roll et al., 2004). The method based on finding mapping guardlines is suitable for linear system with nonlinear discrete parts like switches which changes system behavior in step. Other methods a priori assume that the system dynamics is continuous (Ferrari-Trecate, 2005). Both mentioned approaches use for identification clustering-based algorithms.

As will be pointed out, nonlinear identification techniques can be used under specific conditions in order to obtain linear PWA model. Many neural network based identification techniques use nonlinear neuron functions of one variable which are easier linearizable than whole model of many variables. The key idea is based on linearization of nonlinear neural network functions of single variable. Similarly as Taylor series, it is possible to define any nonlinear function as a series of nonlinear functions. This approach leads to generalized Fourier polynomial series. Generalized Fourier series is based on a set of one-dimensional orthonormal functions $\phi_i^{(N)}$ defined as

$$\int_{x_1}^{x_2} \phi_i^{(N)}(x)\phi_j^{(N)}(x) = \delta_{ij} \qquad (1)$$

where δ_{ij} is the Kronecker delta function and $[x_1, x_2]$ is the domain of interest. Several examples of orthonormal functions are the normalized Fourier (harmonic) functions, Legendre polynomials, Chebyshev polynomials and Laguerre polynomials (Leondes, 1997). In this paper only Chebyshev polynomials will be discussed.

Orthogonal Activation Function based Neural Network (OAF NN) is employed in the task of nonlinear approximation. PWA approximation of every used orthonormal polynomial creates Piecewise Affine Orthogonal Activation Function based Neural Network (PWA OAF NN).

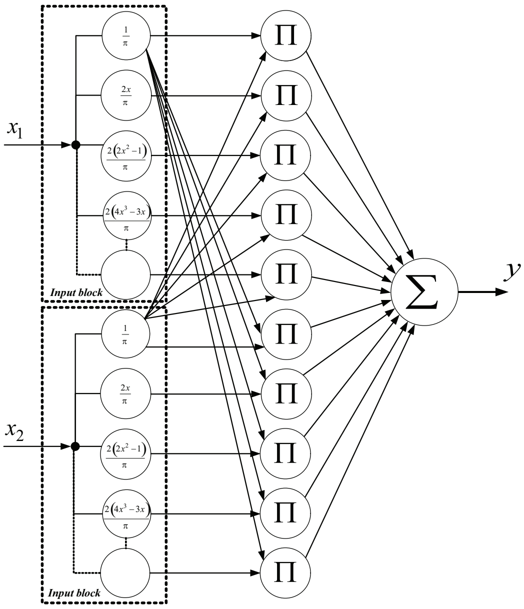


Fig. 1. Adjusted OAF NN structure

2.1 *Chebyshev polynomial*

The Chebyshev polynomials of the first kind can be defined by the trigonometric identity

$$T_n(x) = \cos(n \arccos(x)) \qquad (2)$$

with norm defined as follows

$$\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}}(T_n(x))^2 dx = \begin{cases} \pi & n = 0 \\ \pi/2 & n \neq 1 \end{cases} \qquad (3)$$

Recursive generating formula for Chebyshev polynomials:

$$\begin{aligned} T_0(x) &= 1, & (4)\\ T_1(x) &= x, & (5)\\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x), & (6)\\ T_n(x) &= U_{n+1}(x) - U_{n-1}(x). & (7) \end{aligned}$$

where U_n is the Chebyshev polynomial of the second kind generated by the recursive formula:

$$\begin{aligned} U_0(x) &= 1, & (8)\\ U_1(x) &= 2x, & (9)\\ U_{n+1}(x) &= 2xU_n(x) - U_{n-1}(x), & (10) \end{aligned}$$

The first few Chebyshev polynomials of the first kind are

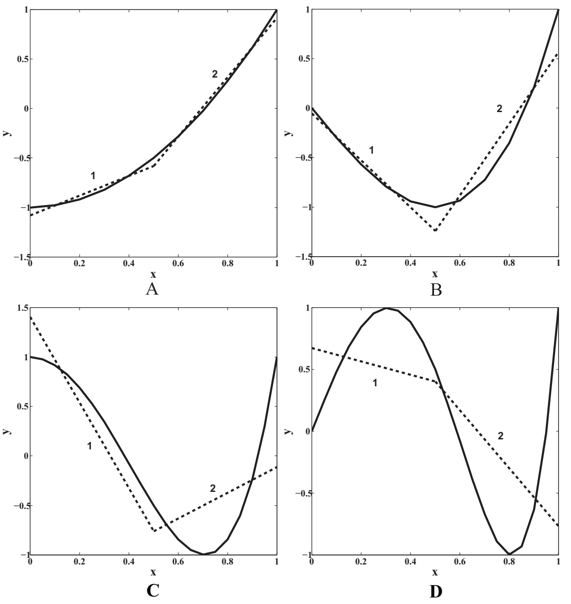


Fig. 2. PWA approximation of T2, T3, T4, T5 Chebyshev polynomials

$$T_0(x) = 1, \tag{11}$$

$$T_1(x) = x, \tag{12}$$

$$T_2(x) = 2x^2 - 1 \tag{13}$$

$$T_3(x) = 4x^3 - 3x \tag{14}$$

$$T_4(x) = 8x^4 - 8x^2 + 1. \tag{15}$$

The first few Chebyshev polynomials of the second kind are

$$U_0(x) = 1, \tag{16}$$

$$U_1(x) = 2x, \tag{17}$$

$$U_2(x) = 4x^2 - 1 \tag{18}$$

$$U_3(x) = 8x^3 - 4x \tag{19}$$

$$U_4(x) = 16x^4 - 12x^2 + 1. \tag{20}$$

2.2 *OAF NN topology*

It is possible to define Generalized Fourier series with orthogonal polynomilas by neural network with one hidden layer. In this work we use a Matlab function framework for orthogonal activation function based neural networks which is part of the toolbox. Aftr slight revision it is possible to use this methodology for modeling the fourier series. Example of the network for modeling function of two variables is depicted in Fig. 1.

If we consider general structure of the network in ARX fashion with na, nb, and nk parameters we get network output equation:

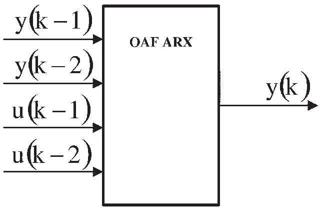


Fig. 3. OAF ARX model for na=2, nb=2, nk=0 or nk=1

$$\begin{aligned}
y &= w_1 \frac{1}{\pi}^p + \dots \\
&\frac{2}{\pi^p}(w_2 T_1(y(k|1)) + \dots + w_n T_{n-1}(y(k|1)))+ \\
&\vdots \\
&\frac{2}{\pi^p}(w_{i_1} T_1(y(k|na)) + \dots + w_{i_2} T_{n-1}(y(k|na)))+ \\
&\frac{2}{\pi^p}(w_{i_3} T_1(u(k|nk)) + \dots + w_{i_4} T_{n-1}(u(k|nk)))+ \\
&\vdots \\
&\frac{2}{\pi^p}(w_{i_5} T_1(u(k|i_7)) + \dots + w_{i_6} T_{n-1}(u(k|i_7))). \\
&i_1 = (na-1)(n-1)+2 \\
&i_2 = na(n-1)+1 \\
&i_3 = nb(n-1)+2 \\
&i_4 = (nb+1)(n-1)+1 \\
&i_5 = (nb+na-1)(n-1)+2; \\
&i_6 = (nb+na)(n-1)+1; \\
&i_7 = nk+nb-1 \\
&p = na+nb
\end{aligned} \tag{21}$$

where $y(k|na)$ denotes $y(k-na)$ and similarly $u(k|nk) \equiv u(k-nk)$. Every Chebyshev polynomial is aproximated by set of lines (Fig. 2)

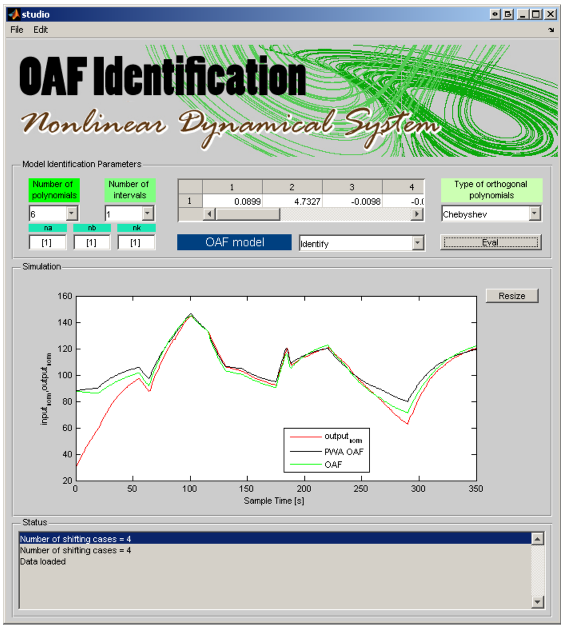
$$T(x) \approx a_i x + b_i \quad \text{for} \quad i = \{1, 2, \dots, n_{div}\} \tag{22}$$

Then output equation becomes difference equation.

A convenient feature of all Chebyshev polynomial is their symmetry. All polynomials of even order are symmetrical by vertical axis and all polynomial of odd order are symmetrical by origin. These properties allow decreasing number of linearization points to half while keeping precision. To get the lowest number of shift cases of generated PWA model we linearized the polynomials in the same points, Fig. 2. The term 'linearization point' denotes the interval division point where the PWA function breaks.

2.3 *Transformation to state space PWA form*

Accuracy of the approximation of nonlinear system is significantly increased when the function is linearized around multiple distinct linearization points. State space PWA structure describes behavior of nonlinear dynamical systems in multiple linearization points.



Number of polynomials: 6
Number of intervals: 1
na: [1]
nb: [1]
nk: [1]

	1	2	3	4
1	0.0899	4.7327	-0.0098	-0.(

Type of orthogonal polynomials: Chebyshev
Identify

Number of shifting cases = 4
Number of shifting cases = 4
Data loaded

Fig. 4. PWA OAF ID studio

$$\begin{aligned} x(k+1) &= A_i x(k) + B_i u(k) + f_i \\ y(k) &= C_i x(k) + D_i u(k) + g_i \end{aligned} \tag{23a}$$

$$IF \begin{bmatrix} x \\ u \end{bmatrix} \in \mathrm{D}_i, \quad i = 1, \ldots, n_L \tag{23b}$$

Every dynamic i is active in polyhedral partition (23b) which can be expressed by inequality

$$guardX_i x(k) + guardU_i u(k) \leq guardC_i \tag{24}$$

Difference equation (21) can be easily transformed to state space form. In Matlab difference equation can be expressed by discrete transfer function. It is possible to use transformation function **tf2ss**. But this policy doesn't lead to desired state space PWA form. Desired state space form has to keep all outputs of difference equation (21) in state vector. So we can correctly define guardline inequality (24).

Here we present transformation example for system with parameters **na=2, nb=2, nk=0 or nk=1**, Fig.3 . Difference equation:

$$\begin{aligned} y(k) = {} & c^{(i)} + c^{(i)}_{y_1} y(k-1) + c^{(i)}_{y_2} y(k-2) + \\ & c^{(i)}_{u_1} u(k-1) + c^{(i)}_{u-2} u(k-2) \end{aligned} \tag{25}$$

In PWA form guidelines are defined for $x1 = u(k-2), x2 = y(k-2), x3 = y(k-1)$ and $u = u(k-1)$ PWA state space model:

$$x(k+1) = A_i x(k) + B_i u(k) + f_i \tag{26a}$$

$$y(k) = C_i x(k) + D_i(k) + g_i \tag{26b}$$

$$A_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ c^{(i)}_{u_2} & c^{(i)}_{y2} & c^{(i)}_{y_1} \end{bmatrix} \tag{26c}$$

$$B_i = \begin{bmatrix} 1 \\ 0 \\ c^{(i)}_{u_1} \end{bmatrix} \tag{26d}$$

$$C_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{26e}$$

$$D_i = 0 \tag{26f}$$

$$f_i = \begin{bmatrix} 0 \\ 0 \\ c^{(i)} \end{bmatrix} \tag{26g}$$

$$g_i = 0; \tag{26h}$$

$$x \in <3 \times 1> \tag{26i}$$

3. PWA OAF IDENTIFICATION TOOLBOX

PWA identification problem has garnered great interest in the research community. In Matlab enviroment several toolboxes were developed for identification hybrid and nonlinear systems (Roll et al., 2004; Ferrari-Trecate, 2005; Julian et al., 1999). The main aim of the PWA OAF Identification Toolbox (PWA OAF IT) is to provide efficient tools for analysis, identification and simulation of PWA OAF model. In following section we present toolbox functionality on several identification examples.

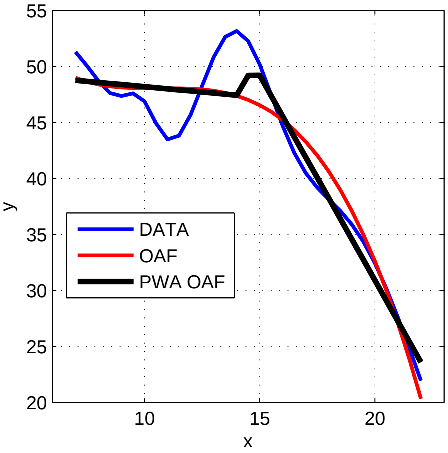
In PWA OAF IT the model is represented by the following fields of the model structure:

`model.na` - Number of past output terms
`model.nb` - Number of past input terms
`model.nk` - Delay from input to the output
`model.npoly` - Number of Chebys. polynomials
`model.ndiv` - Division of {0,1} interval
`model.Fi` - Connection matrix of network
`model.w` - Network parameters
`model.type` - Type of polynomials 'Chebys'
`model.const` - Constant in difference equation
`model.yconst` - Y-cons in difference equation
`model.uconst` - U-const in difference equation
`model.sysStruct` - PWA state space struct
`model.ynorm` - Normalized output data
`model.unorm` - Normalized input data
`model.u` - Input data
`model.y` - Output data
`model.ypar` - Normalization param. of output
`model.upar` - Normalization param. of input

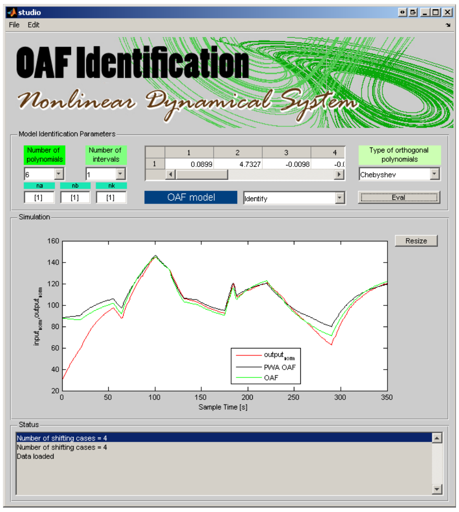
So far PWA OAF ID supports only MISO systems. In order to obtain identified model, call

```
>>model = pwaoafid(y,u,modelstruct,param)
```

Input arguments are in standard notation well



(a) 2-D function example



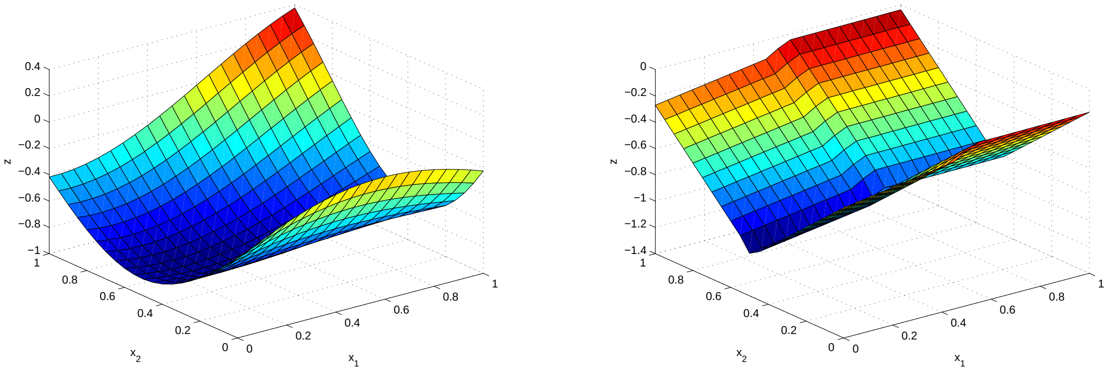
Number of polynomials: 6
Number of intervals: 1
na: [1] nb: [1] nk: [1]

	1	2	3	4
1	0.0899	4.7327	-0.0098	-0.(

Type of orthogonal polynomials: Chebyshev
Identify

Number of shifting cases = 4
Number of shifting cases = 4
Data loaded

(b) PWA-OAF Identification Studio



(a) 3-D function data

(b) PWA approximation of 3-D function

Fig. 5. 3D function

known from PWAID toolbox. For more information type

```
>>help oafpwaid
```

For using gui application Fig. 4, call

```
>> oafpwaid_studio
```

3.1 *Identification of 2-D function*

2-D function is defined by formula:

$$\begin{aligned} y = {} & a_1 e^{-((x-b_1)/c_1)^2} + a_2 e^{-((x-b_2)/c_2)^2} + \\ & a_3 e^{-((x-b_3)/c_3)^2} + a_4 e^{-((x-b_4)/c_4)^2} \end{aligned}$$

$$\begin{aligned} & a_1 = 53.4,\ b_1 = 5.165,\ c_1 = 8.815, \\ & a_2 = 31.25,\ b_2 = 18.69,\ c_2 = 5.109, \\ & a_3 = 20.2,\ b_3 = 13.89,\ c_3 = 2.381, \\ & a_4 = 4.316,\ b_4 = 9.864,\ c_4 = 0.992, \end{aligned} \tag{27}$$

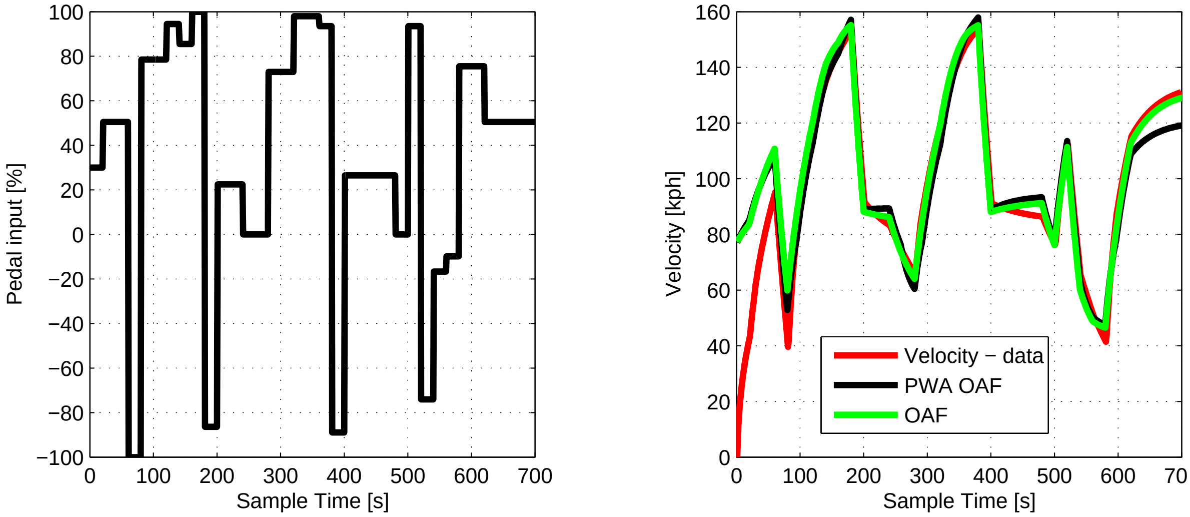
We have made sample data in interval $\{7, 22\}$ (Fig. 5). In our example we did approximation in one point, by two lines. Before parameter estimation it was necessary to normalize data into the interval $\{-1, 1\}$ where Chebyshev polynomials are orthogonal. We used the first four Chebyshev polynomials $T_0 \div T_3$. Mean square error for this approximation is mse $= 5.1947$. To choose a best position of linearization points is a state of art of many algorithms. Through fast network parameters computation it is possible to use even genetic approach to get better position of linearization point and number of chebyshev polynomials.

3.2 *Identification of 3-D function*

Consider a 3-D nonlinear function defined as

$$\begin{aligned} f(\bar{x}) = {} & -.2(\sin(x_1 + 4x_2)) - 2\cos(2x_1 + 3x_2) \\ & -3\sin(2x_1 - x_2) + 4\cos(x_1 - 2x_2) \\ & x_1 \in \{0, 1\}, \\ & x_2 \in \{0, 1\}, \end{aligned} \tag{28}$$

We used the first six Chebyshev polynomials, up to the fifth order $T_0 \div T_5$, linearized in 1 point, each polynomial by two lines. The total number of shifting cases for the resulting PWA function is n_u^{lp+1} where n_u is the number of neural network inputs and lp is the number of linearization points.



(a) Pedal time response

(b) Vehicle speed time response

Fig. 6. Vehicle identification data

For the 3-D function example (28) we get $2^2 = 4$ shifting cases. The result is plotted in Fig. 5b. For this approximation mse=0.0144.

3.3 *Modeling and control of nonlinear dynamic system*

In next example we will try to capture vehicle nonlinear dynamic from input output data for purpose of predictive control design of automatic cruise control. We used Simulink vehicle model with automatic transmission controller (Veh, 2006). Input for model is throttle and break torque signal. Output is vehicle velocity. From the character of input signals we can merge throttle and break torque signal to one input signal (Fig. 6a). Positive part of the input signal is proportional to accelerator pedal pressing and negative part of the input signal is proportional to breaking pedal pressing. Input-output data and identified system output are captured in Fig. 6. We used following identification parameters:

$$\begin{aligned} na &= 1 \\ nb &= 1 \\ nk &= 1 \\ npoly &= 4 \quad \text{polynomials:} T_0, T_1, T_2, T_3 \\ ndiv &= 1 \quad \text{approximation by two lines} \end{aligned} \tag{29}$$

These parameters leads to state space model with one state variable and one input. Acquired PWA state space model has four dynamics (four shifting cases) and it is possible to design an automatic cruise control for such system.
For control design we used MPT toolbox (Kvasnica et al., 2004). We designed explicit mpc controller with time varying reference tracking property. We choosed quadratic cost control problem:

$$\begin{aligned} \min_{u(0),\dots,u(N-1)} &= x(N)^T P_N x(N) + \\ &\sum_{k=1}^{N-1} u(k)^T R u(k) + x(k)^T Q x(k) \end{aligned} \tag{30a}$$

$$s.t.: \begin{cases} x(k+1|t) = f_{dyn}(x(k), u(k)) \\ u_{min} \le u(k) \le u_{max} \\ \Delta u_{min} \le u(k) - u(k-1) \le \Delta u_{max} \\ y_{min} \le g_{dyn}(x(k), u(k)) \le y_{max} \\ x(N) \in T_{set} \end{cases} \tag{30b}$$

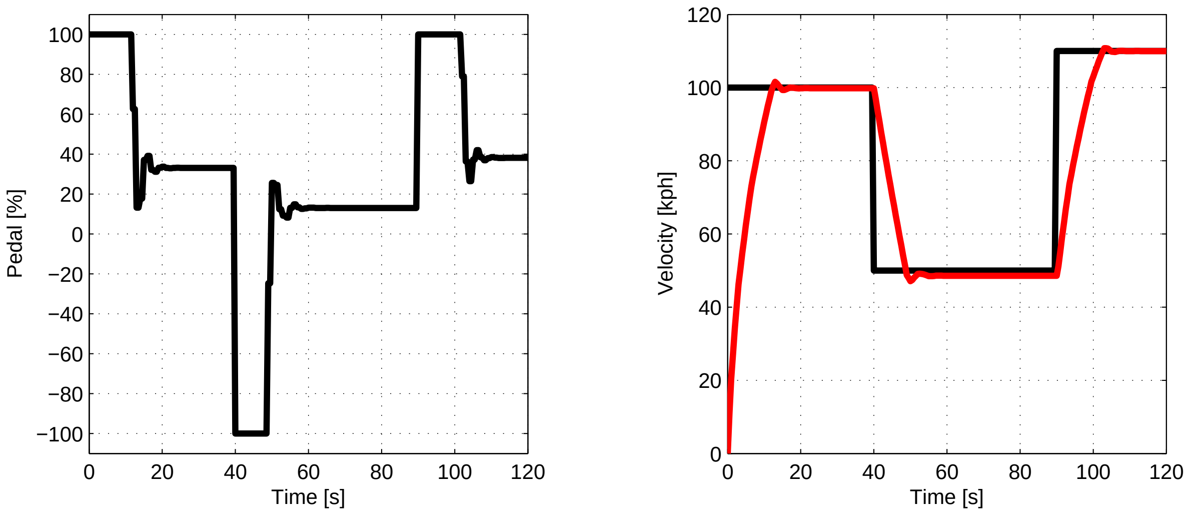
Parameters of control design:

```
       norm: 2
subopt_lev: 0
          N: 3
   tracking: 1
          Q: 100
          R: 1
         Qy: 700
```

Thanks to few PWA dynamics it is possible choose higher prediction horizon to refine control performance. Resulting control law is defined over 430 regions. It is possible to get satisfactory performance with control law defined over fewer number of regions. Designed control law was used in feedback control with nonlinear vehicle model Fig. 7b.

4. CONCLUSION

PWA OAF toolbox significantly improves identification and modeling of nonlinear systems. Transformation to PWA state space model allows to use existing control design tools. So far PWA OAF ID supports only MISO systems. Three studied cases were presented. It was shown that the proposed



(a) Control and reference input time response

(b) Controlled output time response

Fig. 7. Automatic cruise control

approach was effective in model precision and universal in various input configuration. Computation of network parameters is fast and it allows to execute identification for various parameters (order of used Chebyshev polynomials, number of linearization points) to get better performance or even to use genetic approach. Accuracy of the PWA OAF NN approximation depends on the number of linearization points, the highest order of used Chebyshev polynomials and absolute value of computed parameters of the neural network. More linearization points give better precision of the approximation but complexity of the PWA model increases. It is necessary to find suitable proportion between the number of linearization points and required precision.

ACKNOWLEDGMENTS

This paper was supported by Vega project No. 1/1105/11.

References

Modeling an automatic transmission controller, 2006. URL `http://www.mathworks.com/`.

A. Bemporad. An efficient technique for translating mixed logical dynamical systems into piecewise affine systems. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 2, pages 1970 – 1975 vol.2, 2002.

A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *Automatic Control, IEEE Transactions on*, 45(10):1864 – 1876, 2000.

Alberto Bemporad and Manfred Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

G. Ferrari-Trecate. Hybrid Identification Toolbox (HIT), 2005.

W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. On the equivalence of classes of hybrid dynamical models. In *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, 2001a.

W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001b.

P. Julian, A. Desages, and O. Agamennoni. High-level canonical piecewise linear representation using a simplicial partition. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 46(4):463 –480, April 1999.

S. Kozak and J. Stevek. Improved piecewise linear approximation of nonlinear functions in hybrid control. 2010.

M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. URL `http://control.ee.ethz.ch/ mpt/`.

Cornelius T. Leondes, editor. *Control and Dynamic Systems, Neural Network Systems Techniques and Applications, Volume 7*. Academic Press, 1997.

Jacob Roll, Alberto Bemporad, and Lennart Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37 – 50, 2004.

Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31:1691–1724, December 1995. ISSN 0005-1098.

J.L. Villa, M. Duque, A. Gauthier, and N. Rakoto-Ravalontsalama. Translating pwa systems into mld systems. In *Intelligent Control, 2004.*

Proceedings of the 2004 IEEE International Symposium on, pages 37 – 42, 2004.

MATLAB Toolbox for Automatic Approximation of Nonlinear Functions

Alexander Szücs [*,1], **Michal Kvasnica** [*], and **Miroslav Fikar** [*]

[*] *Institute of Automation, Information Engineering and Mathematics, Slovak University of Technology, 812 37 Bratislava, Slovakia*

Abstract: : Given a nonlinear dynamical system in analytic form, the paper proposes a novel method for approximating the system by a suitable hybrid model such that the approximation accuracy is maximized. Specifically, the problem of approximating generic nonlinear functions by piecewise affine (PWA) models is considered. We show that under mild assumptions, the task can be transformed into a series of one-dimensional approximations, for which we propose an efficient solution method based on solving simple nonlinear programs. Moreover, the paper discusses a software implementation of the proposed procedure in form of a MATLAB toolbox.

Keywords: hybrid systems, approximation, nonlinear optimization

1. INTRODUCTION

Mathematical models of physical plants play a vital role in many areas, such as in rigorous simulations, analysis, or control synthesis. Typically, high model accuracy is usually desired while keeping the model complexity on an acceptable level. Traditionally, nonlinear models were preferred from simulations, while most of available control techniques are based on a local approximation around a single operating point. The concept of hybrid systems (Branicky, 1995) can be viewed as a compromise solution between accuracy of the model and its complexity. Hybrid models feature a collection of local models accompanied with logic IF-THEN conditions which enforce switching of the local dynamics. When all local models are linear (or affine), such systems are referred to as *linear* hybrid systems. Although still nonlinear due to the presence of switches, the underlying piecewise linearity allows for somewhat easier control synthesis and analysis compared to using full nonlinear models. Several mathematical frameworks capable of capturing the relation between logic rules and linear dynamics can be used: Piecewise Affine (PWA) models (Sontag, 1981), Mixed Logical Dynamical (MLD) systems (Bemporad and Morari, 1999), Linear Complementarity systems (Heemels et al., 2000) and max-min-plus-scaling models (De Schutter and Van den Boom, 2001). Under mild assumptions, all these frameworks are equivalent to each other and it is possible to transform e.g. the MLD system into a PWA model and vice-versa (Heemels et al., 2001). For the purpose of this work we consider PWA models, which use the concept of multiple linearization to approximate a given nonlinear system with arbitrary accuracy.

The problem which we address in this paper is the following: given a nonlinear dynamical model $x^+ = f(x,u)$ and a fixed complexity of its PWA approximation $\widetilde{f}(x,u) \approx f(x,u)$, how should one design $\widetilde{f}$ which minimizes the approximation error $\int(f(x,u) - \widetilde{f}(x,u))^2$? The answer is non-trivial even putting optimality of the approximation aside. Traditionally, two distinct approaches for deriving PWA approximations are used. When the mathematical formulation of the original nonlinear system is known, one can design the approximation by hand. This is usually done by employing human knowledge and experience to devise several linearization points around which the original nonlinear model should be linearized. Needless to say, placement of such points has a crucial impact on the accuracy of the approximation. The HYSDEL (Hybrid Systems Description Language) tool (Torrisi and Bemporad, 2004; Kvasnica and Herceg, 2010) can be used to accelerate this line of development. Formally, HYSDEL transforms a linguistic description of a hybrid system into the corresponding MLD model, which can then be converted into the PWA form. The language allows to define IF-THEN switching rules which, based on whether some logic condition is satisfied or not, enforce certain continuous dynamics. Another option is to use hybrid identification techniques (Ferrari-Trecate et al., 2001; Roll et al., 2004; Ferrari-Trecate, 2005) to construct the PWA approximation from the input-output measurements. The crucial advantage is that the model of the original nonlinear system is not required to be fully available. The downside, however, is that the approximation is only accurate in the interval captured by the identification data. Moreover, the procedure is computationally expensive and suited mainly to low-dimensional problems.

In this work we propose to use an optimization-based approach to derive PWA approximations of nonlinear systems whose vector field is an a-priori known function of multiple variables. After formally stating the problem in Section 2, we show in Section 3 that an optimal PWA approximation of generic nonlinear functions in one variable can be formulated and solved as a nonlinear programming problem. Subsequently, the approach is extended to deriving PWA approximations of multivariable functions in Section 4. We show that, under a certain assumption, the

[1] Corresponding author, e-mail: alexander.szucs@stuba.sk

problem boils down to solving a series of one-dimensional approximations.

The algorithmic and software implementation of the approximation procedure are then discussed in Section 5. Specifically, we introduce a new software toolbox which packs the proposed approximation strategy in an easily accessible form. Specifically, the toolbox allows user to perform the approximation either directly from MATLAB's command line, or by using a custom graphical user interface. Short, yet illuminating examples are provided to illustrate capabilities of the toolbox.

2. PROBLEM STATEMENT

We consider generic dynamic systems in discrete-time

$$x^+ = f(x, u), \tag{1}$$

where the vector field $f(\cdot, \cdot)$ is assumed to be continuous in the state variables $x \in \mathbb{R}^{n_x}$ and in the inputs $u \in \mathbb{R}^{n_u}$. System states and inputs are assumed to be constrained to connected and closed domains $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$, respectively.

The objective is to approximate (1) by a different dynamic system $x^+ = \widetilde{f}(x, u)$ whose vector field $\widetilde{f}(x, u)$ is a PWA function which consists of a pre-specified number N of local linear dynamics:

$$\widetilde{f}(x,u) = \begin{cases} A_1 x + B_1 u + c_1 & \text{if } \left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \mathcal{R}_1 \\ \quad\vdots & \quad\vdots \\ A_N x + B_N u + c_N & \text{if } \left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \mathcal{R}_N. \end{cases} \tag{2}$$

Here, $A_i \in \mathbb{R}^{n_x \times n_x}$, $B_i \in \mathbb{R}^{n_x \times n_u}$, $c_i \in \mathbb{R}^{n_x}$, are the state-update matrices of the i-th local linear approximation, and $\mathcal{R}_i \subset \mathbb{R}^{n_x \times n_u}$ is the region of validity of the i-th local model satisfying $\mathcal{R}_i \neq \emptyset$, $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$, $\forall i \neq j$, and $\cup_i \mathcal{R}_i = \mathcal{X} \times \mathcal{U}$.

Formally, the problem which we aim at solving can be stated as follows:

Problem 2.1. Given a nonlinear vector field $f(x, u)$ of system (1), find the PWA approximation (2) of pre-specified complexity which minimizes the approximation error

$$e_{\text{aprx}} := \int (f(x,u) - \widetilde{f}(x,u))^2 \, \mathrm{d}x \mathrm{d}u, \tag{3}$$

where the integral is evaluated over the whole region of validity of (1), i.e. over $\mathcal{X} \times \mathcal{U}$.

In the sequel we show how to solve Problem 2.1 provided that the vector field $f(z)$, $z = [x,\ u]^T$ satisfies the following assumption.

Assumption 2.2. The function $f(z_1, \ldots, z_n)$ can be written as $\sum_{i=1}^{n} \alpha_i \Big(\prod_{j=p_i}^{q_i} f_j(z_j) \Big)$.

As an example, the function $z_1 e^{z_2}$ satisfies such an assumption, while the function $e^{z_1 z_2}$ does not. Although the assumption is somewhat restrictive, the gained advantage is that approximating any multivariable function $f(z_1, \ldots, z_n)$ boils down to solving a series of 1D problems, as evidenced in the following two sections.

Remark 2.3. Since the approximation procedure discussed in the sequel considers only the vector field in the right-hand-side of (1), continuous-time systems $\dot{x} = f(x, u)$ can be treated as well.

3. FUNCTIONS IN ONE VARIABLE

First, we consider the one-dimensional case, i.e. approximating a nonlinear function $f(z) : \mathbb{R} \mapsto \mathbb{R}$, with domain $\mathcal{Z} \subset \mathbb{R}$, by a PWA function $\widetilde{f}(z) = a_i z + c_i$ if $z \in \mathcal{R}_i$. Since $\mathcal{Z}$ is assumed to be connected and closed, it is a line segment $[\underline{z},\ \overline{z}]$. Regions $\mathcal{R}_i$ define the partition of such a line into N non-overlapping parts, i.e. $\mathcal{R}_1 = [\underline{z},\ r_1]$, $\mathcal{R}_2 = [r_1,\ r_2]$, …, $\mathcal{R}_{N-1} = [r_{N-2},\ r_{N-1}]$, $\mathcal{R}_N = [r_{N-1},\ \overline{z}]$ with $\cup_i \mathcal{R}_i = [\underline{z},\ \overline{z}]$. Solving Problem 2.1 then becomes to find the slopes a_i, offsets c_i and breakpoints r_i such that the approximation error is minimized, i.e.

$$\min_{a_i, c_i, r_i} \int_{\underline{z}}^{\overline{z}} (f(z) - \widetilde{f}(z))^2 \, \mathrm{d}z \tag{4a}$$

$$\text{s.t. } \widetilde{f}(z) = \begin{cases} a_1 z + c_1 & \text{if } z \in [\underline{z},\ r_1] \\ \quad\vdots & \quad\vdots \\ a_N z + c_N & \text{if } z \in [r_{N-1},\ \overline{z}] \end{cases} \tag{4b}$$

$$\underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \tag{4c}$$

$$a_i r_i + c_i = a_{i+1} r_i + c_{i+1},\ i = 1, \ldots, N-1, \tag{4d}$$

where (4d) enforces continuity of $\widetilde{f}(z)$ along the breakpoints r_i. The IF-THEN based nonlinear constraint (4b) can be eliminated by observing that, by definition, regions $\mathcal{R}_i$ are non-overlapping, and the integral in (4a) can hence be written as

$$\int_{\underline{z}}^{\overline{z}} \big(f(z) - \widetilde{f}(z)\big)^2 \mathrm{d}z = \sum_{i=1}^{N} \Big(\int_{r_{i-1}}^{r_i} \big(f(z) - (a_i z + c_i)\big)^2 \mathrm{d}z \Big), \tag{5}$$

with $r_0 = \underline{z}$ and $r_N = \overline{z}$. The NLP (4) can therefore be written as

$$\min_{a_i, c_i, r_i} \sum_{i=1}^{N} \Big(\int_{r_{i-1}}^{r_i} \big(f(z) - (a_i z + c_i)\big)^2 \, \mathrm{d}z \Big) \tag{6a}$$

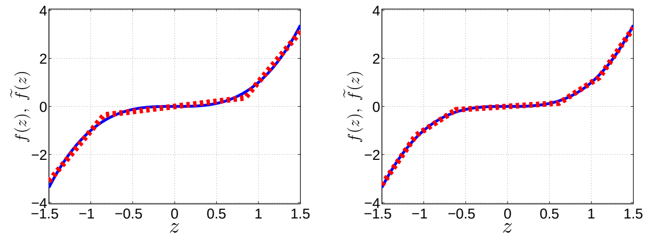
$$\text{s.t. } \underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \tag{6b}$$

$$a_i r_i + c_i = a_{i+1} r_i + c_{i+1},\ i = 1, \ldots, N-1. \tag{6c}$$

Remark 3.1. The number of approximation segments can be be reduced by normalizing the domain of f to an interval $[-1,\ 1]$.

For simple functions $f(z)$, the integral in (6a) can be expressed in an analytical form in unknowns a_i, c_i, r_i, along with the corresponding gradients. For more complex expressions, the integrals can be evaluated numerically, e.g. by using the trapezoidal rule. In either case, problem (6) can be solved to a local optimality e.g. by using the `fmincon` solver of MATLAB. Alternatively, one can use global optimization methods (Adjiman et al., 1996; Papamichail and Adjiman, 2004; Chachuat et al., 2006) that guarantee that an ϵ-neighborhood of the global optimum can be found.

Example 3.2. Consider the function $f(z) = z^3$ on domain $-1.5 \leq z \leq 1.5$. The analytic form of the integral (6a) is



(a) PWA approximation with 3 regions.

(b) PWA approximation with 5 regions.

Fig. 1. Graph of $f(z) = z^3$ (blue line) and the PWA approximations $\widetilde{f}(z)$ (red dashed lines).

$$\sum_{i=1}^{N} \Big(c_i^2(r_i + r_{i-1}) + a_i c_i(r_i^2 - r_i^2) + \frac{a_i^2}{3}(r_i^3 - r_{i-1}^3) - \\ -\frac{c_i}{2}(r_i^4 - r_{i-1}^4) - \frac{2a_i}{5}(r_i^5 - r_{i-1}^5) + \frac{1}{7}(r_i^7 - r_{i-1}^7)\Big),$$

with $r_0 = -1.5$ and $r_N = 1.5$. The PWA approximation of $f(z)$ with $N = 3$ regions was found by solving the NLP (6) using `fmincon`, which took 0.05 seconds on a 2.4 GHz CPU running MATLAB 2009b. The obtained PWA approximation is then given by

$$\widetilde{f}(z) = \begin{cases} 4.1797z + 3.1621 & \text{if } -1.5 \le z \le -0.8423 \\ 0.4257z & \text{if } -0.8423 \le z \le 0.8423 \\ 4.1797z - 3.1621 & \text{if } \;\; 0.8423 \le z \le 1.5 \end{cases}$$

The approximation accuracy can be increased by roughly a factor of 10 by approximating $f(z)$ by $N = 5$ regions, as can be seen from Figure 1.

4. MULTIVARIABLE FUNCTIONS

The task is to approximate a given multivariable function $f(z_1, \ldots, z_n) : \mathbb{R}^n \mapsto \mathbb{R}$ with domain $\mathcal{Z} \subset \mathbb{R}^n$ by a PWA function $\widetilde{f}(z_1, \ldots, z_n)$, defined over the same domain, such that the approximation error (3) is minimized.

Definition 4.1. (Williams (1993)). Function $f(z_1, \ldots, z_n)$ is called *separable* if it can be expressed as the sum of functions of a single variable, i.e. $f(z_1, \ldots, z_n) = f_1(z_1) + \cdots + f_n(z_n)$.

If $f(z_1, \ldots, z_n)$ is readily separable (e.g. when $f(z_1, z_2) = e^{z_1} + \sin(z_2)$), its optimal PWA approximation can be obtained by applying the 1D scenario of Section 3 to the individual components of the function, i.e. $\widetilde{f}(z_1, \ldots, z_n) = \widetilde{f}_1(z_1) + \cdots + \widetilde{f}_n(z_n)$. The total number of regions over which the PWA approximation $\widetilde{f}(\cdot)$ is defined is hence given by $\sum_{j=1}^{n} N_j$, where N_j is the pre-specified complexity of the j-th approximation $\widetilde{f}_j(z_j)$.

A surprisingly large number of non-separable functions can be converted into the separable form by applying a simple trick, elaborated in more details e.g. in Williams (1993). To introduce the procedure, consider a non-separable function $f(z_1, z_2) = z_1 z_2$ with domain $\mathcal{Z} := [\underline{z}_1, \ \overline{z}_1] \times [\underline{z}_2, \ \overline{z}_2]$. Define two new variables

$$y_1 = (z_1 + z_2), \quad y_2 = (z_1 - z_2). \tag{7}$$

Then it is easy to verify that $1/4(y_1^2 - y_2^2) = z_1 z_2$. The coordinate transformation therefore transforms the original function into a separable form, where both terms (y_1^2 and y_2^2) are now functions of a single variable. The procedure of Section 3 can thus be applied to compute PWA approximations of $f_{y_1}(y_1) := y_1^2$ and $f_{y_2}(y_2) := y_2^2$, where the function arguments relate to z_1 and z_2 via (7). Important to notice is that $f_{y_1}(\cdot)$ and $f_{y_2}(\cdot)$ have different domains, therefore their PWA approximations $\widetilde{f}_{y_1}(y_1) \approx y_1^2$ and $\widetilde{f}_{y_2}(y_2) \approx y_2^2$ will, in general, be different. Specifically, the domain of $f_{y_1}(\cdot)$ is $[\underline{y}_1, \ \overline{y}_1]$ with $\underline{y}_1 = \min\{z_1 + z_2 \mid \underline{z}_1 \le z_1 \le \overline{z}_1, \ \underline{z}_2 \le z_2 \le \overline{z}_2\}$ and $\overline{y}_1 = \max\{z_1 + z_2 \mid \underline{z}_1 \le z_1 \le \overline{z}_1, \ \underline{z}_2 \le z_2 \le \overline{z}_2\}$. Similarly, the domain of $f_{y_2}(\cdot)$ is $[\underline{y}_2, \ \overline{y}_2]$, whose boundaries can be computed by respectively minimizing and maximizing $z_1 - z_2$ subject to the constraint $[z_1, \ z_2]^T \in \mathcal{Z}$. The overall PWA approximation $\widetilde{f}(z_1, z_2) \approx z_1 z_2$ then becomes

$$\widetilde{f}(z_1, z_2) = 1/4(\widetilde{f}_{y_1}(z_1 + z_2) - \widetilde{f}_{y_2}(z_1 - z_2)). \tag{8}$$

The value of $\widetilde{f}(z_1, z_2)$ for any points z_1, z_2 is obtained by subtracting the value of the PWA function $\widetilde{f}_{y_2}(\cdot)$ evaluated at the point $z_1 - z_2$ from the function value of $\widetilde{f}_{y_1}(\cdot)$ evaluated at $z_1 + z_2$, followed by a linear scaling.

The procedure naturally extends to multivariable functions represented by the product of two nonlinear functions of a single variable, i.e. $f(z_1, z_2) = f_1(z_1) f_2(z_2)$. Here, the transformation (7) becomes

$$y_1 = f_1(z_1) + f_2(z_2), \quad y_2 = f_1(z_1) - f_2(z_2). \tag{9}$$

Therefore, $1/4(y_1^2 - y_2^2) = f(z_1, z_2)$ still holds. Let $f_{y_1}(y_1) := y_1^2$ and $f_{y_2}(y_2) := y_2^2$. The domain of $f_{y_1}(\cdot)$ is $[\underline{y}_1, \ \overline{y}_1]$ and $\operatorname{dom} f_{y_2}(\cdot) = [\underline{y}_2, \ \overline{y}_2]$ with

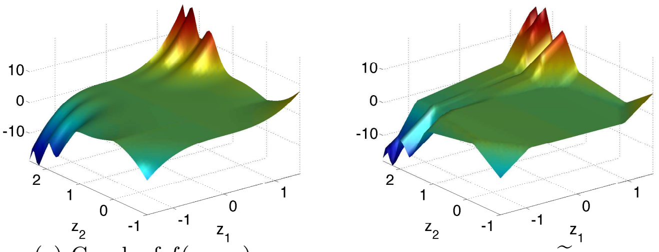
$$\underline{y}_1 = \min\{f_1(z_1) + f_2(z_2) \mid [z_1, \ z_2]^T \in \mathcal{Z}\}, \tag{10a}$$
$$\overline{y}_1 = \max\{f_1(z_1) + f_2(z_2) \mid [z_1, \ z_2]^T \in \mathcal{Z}\}, \tag{10b}$$
$$\underline{y}_2 = \min\{f_1(z_1) - f_2(z_2) \mid [z_1, \ z_2]^T \in \mathcal{Z}\}, \tag{10c}$$
$$\overline{y}_2 = \max\{f_1(z_1) - f_2(z_2) \mid [z_1, \ z_2]^T \in \mathcal{Z}\}, \tag{10d}$$

which can be computed by solving four NLP problems. Finally, since all expressions are now functions of a single variable, the PWA approximations $\widetilde{f}_1(z_1) \approx f_1(z_1)$, $\widetilde{f}_2(z_2) \approx f_2(z_2)$, $\widetilde{f}_{y_1}(y_1) \approx f_{y_1}(y_1)$, and $\widetilde{f}_{y_2}(y_2) \approx f_{y_2}(y_2)$ can be computed by solving the NLP (6). The overall optimal PWA approximation $\widetilde{f}(z_1, z_2) \approx f(z_1, z_2)$ then becomes

$$\widetilde{f}(z_1, z_2) = 1/4\Big(\widetilde{f}_{y_1}\big(\widetilde{f}_1(z_1) + \widetilde{f}_2(z_2)\big) - \widetilde{f}_{y_2}\big(\widetilde{f}_1(z_1) - \widetilde{f}_2(z_2)\big)\Big). \tag{11}$$

The evaluation procedure is similar as above. I.e., given the arguments z_1 and z_2, one first evaluates $\widetilde{z}_1 = \widetilde{f}_1(z_1)$ and $\widetilde{z}_2 = \widetilde{f}_2(z_2)$. Subsequently, one evaluates $\widetilde{y}_1 = \widetilde{f}_{y_1}(\cdot)$ with the argument $\widetilde{z}_1 + \widetilde{z}_2$, then $\widetilde{y}_2 = \widetilde{f}_{y_2}(\cdot)$ at the point $\widetilde{z}_1 - \widetilde{z}_2$. Finally, $\widetilde{f}(z_1, z_2) = 1/4(\widetilde{y}_1 - \widetilde{y}_2)$.

Example 4.2. Consider a non-separable function given by $f(z_1, z_2) = f_1(z_1) f_2(z_2)$ with $f_1(z_1) = z_1^3$, $f_2(z_2) = |z_2| + 0.5z_2^2 - \sin(z_2)^3$ on domain $[-1.5, \ 1.5] \times [-1, \ 2.5]$. Graph of the function is shown in Figure 2(a). In order to convert $f(z_1, z_2)$ into a separable form, we introduce variables y_1 and y_2 as per (9). The PWA approximation $\widetilde{f}(z_1, z_2) \approx f(z_1, z_2)$ is then given by (11). Here, $\widetilde{f}_1(z_1)$ was obtained by approximating $f_1(z_1)$ by a PWA

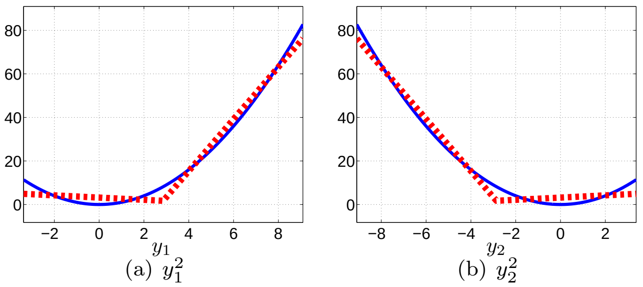


(a) Graph of $f(z_1, z_2)$.

(b) Approximation $\widetilde{f}(z_1, z_2)$

Fig. 2. Graph of $f(z_1, z_2)$ and its PWA approximation (11) in Example 4.2.

function with 3 regions as shown in Figure 1(a), while $\widetilde{f}_2(z_2) \approx f_2(z_2)$ was approximated by 7 regions. Subsequently, the domains $[\underline{y}_1,\ \overline{y}_1]$ and $[\underline{y}_2,\ \overline{y}_2]$ were computed via (10), which resulted into $\operatorname{dom} y_1 = [-3.374,\ 9.095]$ and $\operatorname{dom} y_2 = [-9.095,\ 3.374]$. Finally, the PWA approximations $\widetilde{f}_{y_1}(y_1) \approx y_1^2$ and $\widetilde{f}_{y_2}(y_2) \approx y_2^2$ were obtained by solving the NLP (6) with $N = 2$. Graphs of y_1^2, y_2^2 and their respective PWA approximations are presented in Figure 3. The overall approximation $\widetilde{f}(z_1, z_2)$ therefore consists of 14 regions. Despite a rather crude approximation of the square functions, the combined PWA function (11), shown in Figure 2(b), features only a minor average approximation error of 3% and a worst-case error of 15%. By increasing the number of linearizations for y_1^2 and y_2^2 from $N = 2$ to $N = 4$ (hence increasing the complexity of $\widetilde{f}(z_1, z_2)$ from 14 to 18 regions), the average and worst-case errors can be further reduced to 1% and 8%, respectively.



(a) y_1^2

(b) y_2^2

Fig. 3. Functions y_i^2 (blue) and their PWA approximation $\widetilde{f}_{y_i}(y_i)$ (red dashed lines) in Example 4.2.

Separation of multivariable functions with more than two terms can be performed in an inductive manner. Consider $f(z_1, z_2, z_3) = f_1(z_1)f_2(z_2)f_3(z_3)$. First, approximate the product $f_1(z_1)f_2(z_2)$ by a PWA function of the form of (11), which requires four PWA approximations

$$\widetilde{f}_1(\cdot) \approx f_1(\cdot),\ \widetilde{f}_2(\cdot) \approx f_2(\cdot),\ \widetilde{f}_{y_1}(\cdot) \approx y_1^2,\ \widetilde{f}_{y_2}(\cdot) \approx y_2^2,$$

with y_1 and y_2 as in (9). Let $f_a(z_1, z_2) := f_1(z_1)f_2(z_2)$. Then $f(z_1, z_2, z_3) = f_a(z_1, z_2)f_3(z_3)$, which can again be approximated as a product of two functions. Specifically, define

$$y_3 = f_a(\cdot) + f_3(z_3), \quad y_4 = f_a(\cdot) - f_3(z_3), \tag{12}$$

and hence $f_a(z_1, z_2)f_3(z_3) = 1/4(y_3^2 - y_4^2)$. The domains over which y_3^2 and y_4^2 need to be approximated are, respectively, $[\underline{y}_3,\ \overline{y}_3]$ and $[\underline{y}_4,\ \overline{y}_4]$ with

$$\underline{y}_3 = \min\{f_1(z_1)f_2(z_2) + f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13a}$$
$$\overline{y}_3 = \max\{f_1(z_1)f_2(z_2) + f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13b}$$
$$\underline{y}_4 = \min\{f_1(z_1)f_2(z_2) - f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13c}$$
$$\overline{y}_4 = \max\{f_1(z_1)f_2(z_2) - f_3(z_3) \mid z \in \mathcal{Z}\}, \tag{13d}$$

and $z = [z_1,\ z_2,\ z_3]^T$. Subsequently, three additional PWA approximations

$$\widetilde{f}_{y_3}(y_3) \approx y_3^2,\ \widetilde{f}_{y_4}(y_4) \approx y_4^2,\ \widetilde{f}_3(z_3) \approx f_3(z_3)$$

need to be computed over the corresponding domains. The aggregated optimal PWA approximation $\widetilde{f}(z_1, z_2, z_3) \approx f(z_1)f(z_2)f(z_3)$ consists of 7 individual approximations and is given by

$$\widetilde{f}(\cdot) = 1/4\Big(\underbrace{\widetilde{f}_{y_3}\big(\widehat{f}_a + \widetilde{f}_3(z_3)\big)}_{\widehat{y}_3} - \underbrace{\widetilde{f}_{y_4}\big(\widehat{f}_a - \widetilde{f}_4(z_3)\big)}_{\widehat{y}_4}\Big). \tag{14}$$

Here, $\widehat{f}_a$ is the function value of $\widetilde{f}_a(z_1, z_2) \approx f_1(z_1)f_2(z_2)$ at z_1 and z_2, where $\widetilde{f}_a(\cdot)$ is obtained from (11), i.e.:

$$\widehat{f}_a = 1/4\Big(\underbrace{\widetilde{f}_{y_1}\big(\widetilde{f}_1(z_1) + \widetilde{f}_2(z_2)\big)}_{\widehat{y}_1} - \underbrace{\widetilde{f}_{y_2}\big(\widetilde{f}_1(z_1) - \widetilde{f}_2(z_2)\big)}_{\widehat{y}_2}\Big). \tag{15}$$

The overall PWA approximation $\widetilde{f}(z_1, z_2, z_3)$ can then be evaluated, for any $z_1,\ z_2,\ z_3 \in \mathcal{Z}$, by computing the function values of the respective approximations in the following order:

Step 1: $\widehat{y}_1 = \widetilde{f}_{y_1}(\widetilde{f}_1(z_1) + \widetilde{f}_2(z_2))$,
Step 2: $\widehat{y}_2 = \widetilde{f}_{y_2}(\widetilde{f}_1(z_1) - \widetilde{f}_2(z_2)$,
Step 3: $\widehat{y}_3 = \widetilde{f}_{y_3}(1/4(\widehat{y}_1 - \widehat{y}_2) + \widetilde{f}_3(z_3))$,
Step 4: $\widehat{y}_4 = \widetilde{f}_{y_4}(1/4(\widehat{y}_1 - \widehat{y}_2) - \widetilde{f}_3(z_3))$,
Step 5: $\widetilde{f}(z_1, z_2, z_3) = 1/4(\widehat{y}_3 - \widehat{y}_4)$.

Such an inductive procedure can be repeated *ad-infinitum* to derive PWA approximations of any multivariable function which satisfies Assumption 2.2. In general, the PWA approximation will consists of $2p + n$ individual PWA functions, where n is the number of variables in $f(z_1, \ldots, z_n)$ and p is the number of products between individual subfunctions $f_j(z_j)$. As an example, for $f(\cdot) := \alpha_1 f_1(z_1)f_2(z_2)f_4(z_4) + \alpha_2 f_3(z_3)f_5(z_5)$ we have $p = 3$. We remark that inclusion of scalar multipliers α_j into the PWA description of the form (14)–(15) is straightforward and only requires linear scaling of the corresponding terms.

Remark 4.3. Since approximation of multivariable functions boils down to a series of 1D approximations which are then aggregated by a linear relation in (8), the overall approximation error is proportional to the sum of individual approximation errors.

Remark 4.4. Due to a linear nature of the aggregation in (8) and due to the fact that each single 1D approximation is continuous due to (6c), the overall multivariable approximation $\widetilde{f}$ is continuous as well.

5. SOFTWARE IMPLEMENTATION

Next, we discuss software implementation of the approximation procedure described above. The implementation is provided in a form of an open-source MATLAB toolbox, called AUTOPROX, which is freely available from `http://www.kirp.chtf.stuba.sk/∼sw/`. The toolbox provides two types of user interfaces. Input data can either be provided directly from the command line or, alternatively, entered using a graphical interface.

5.1 Command-Line Interface

The command-line interface is illustrated first by revisiting Example 3.2. To approximate the function $f(z) = z^3$, one proceeds as follows:

```
syms z
f = z^3
bounds = [-1.5, 1.5]
regions = 3
[aprx, data] = autoprox_1d(f, bounds, regions)
```

Here, AUTOPROX uses the Symbolic Toolbox to define symbolic representation of the function to be approximated on a given domain (represented by the `bounds` variable), with a given number of PWA segments (the `regions` variable). The first output argument (denotes as `aprx` here) is a function handle, which can be used e.g. to plot the approximation:

```
x = -1.5:0.001:1.5
plot(x, x.^3, x, aprx(x), '--')
```

which will generate a plot as seen in Figure 1(a). The second output (stored in the `data` variable) can be used to export the PWA approximation into the HYSDEL language:

```
 hysdel_1d(data, 'filename.hys')
```

The generated HYSDEL model can be subsequently compiled by the HYSDEL compiler, which will provide a mathematical model suitable e.g. for control synthesis.

Approximation of 2D functions can be performed in a similar manner. Let us again consider Example 4.2, i.e. the task is to approximate the function $f(z_1, z_2) = z_1^3(|z_2| + 0.5z_2^2 - \sin(z_2)^3)$ on domain $[-1.5,\ 1.5] \times [-1,\ 2.5]$. Again, the first step is to define the function using symbolic variables:

```
syms z1 z2
f1 = z1^3
f2 = abs(z2) + 0.5*z2^2 - sin(z2^3)
```

Next, the function domain and number of approximation segments need to be provided:

```
f1_bounds = [-1.5, 1.5]
f2_bounds = [-1, 2.5]
f1_regions = 3
f2_regions = 7
y1_regions = 2
y2_regions = 2
```

Finally, the approximation $\widetilde{f}(z_1, z_2)$ can be obtained by calling

```
[aprx,data] = autoprox_2d(f1,f2,f1_bounds,f2_bounds,...
 f1_regions, f2_regions,y1_regions, y2_regions)
```

Similarly as in the previous example, the `aprx` output is a function handle which can be used to directly evaluate the approximation at some given values of z_1 and z_2, e.g.

```
z1 = 0.5
z2 = -1
true_value = z1^3*(abs(z2) + 0.5*z2^2 - sin(z2^3))
aprx_value = aprx(z1, z2)
```

The second output (called `data`) again serves to generate the HYSDEL version of the approximation:

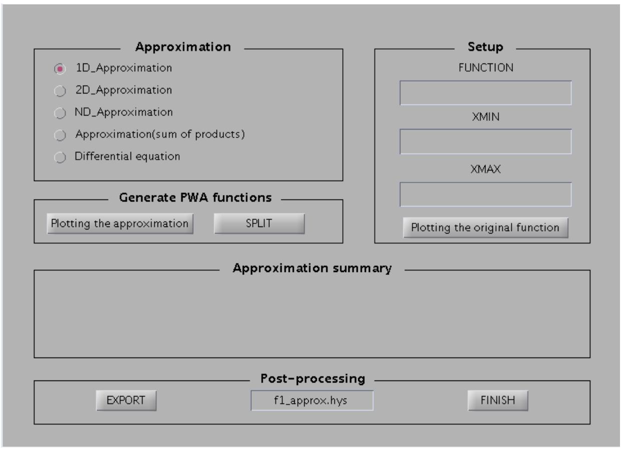
```
hysdel_2d(data, 'filename.hys')
```

Approximation of n-dimensional functions can be obtained by calling the `autoprox_nd` function. A detailed description of its calling syntax is omitted due to brevity, but is provided in the distribution package of AUTOPROX.

5.2 Graphical User Interface (GUI)

The GUI allows to perform the approximation in an easily accessible manner where all data can be entered conveniently without the need to remember the exact calling syntax of individual approximation functions.

The main window of the GUI is shown in Figure 4. The user starts by selecting the type of approximation using radio buttons. Then, he provides the symbolic representation of the function to approximate in the *FUNCTION* text box. The domain of the function, represented by its minimal and maximal bounds, has to be filled out next. After providing all necessary details, the user can select the number of approximation regions by a drop-down menu, as shown in Figure 5. Afterwards, the approximation is computed by clicking the *SPLIT* button. A concise statistical evaluation of the approximation will then appear in a corresponding section of the GUI. It informs the user about the approximation quality, represented by average and worst-case approximation errors. Finally, the approximation can be exported to a HYSDEL source by clicking the *EXPORT* button.

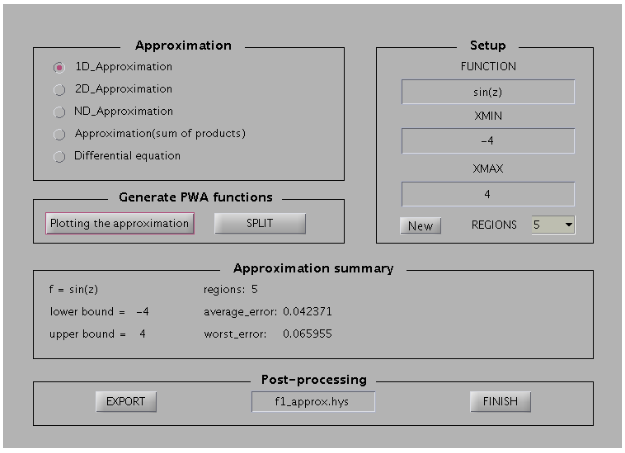

Approximation
● 1D_Approximation
○ 2D_Approximation
○ ND_Approximation
○ Approximation(sum of products)
○ Differential equation

Setup
FUNCTION
XMIN
XMAX

Post-processing
f1_approx.hys

Fig. 4. Basic GUI window.

It should be noted that the GUI is still subject to active development and substantial modifications are expected within next following months.



Approximation

● 1D_Approximation
○ 2D_Approximation
○ ND_Approximation
○ Approximation(sum of products)
○ Differential equation

Setup

FUNCTION: sin(z)
XMIN: -4
XMAX: 4
REGIONS: 5

Approximation summary

f = sin(z)	regions: 5
lower bound = -4	average_error: 0.042371
upper bound = 4	worst_error: 0.065955

Post-processing

f1_approx.hys

Fig. 5. GUI windows after performing approximation.

6. CONCLUSIONS

We have shown that a large class of dynamical systems with nonlinear vector fields can be approximated by PWA systems of fixed complexity in an optimal manner. The procedure boils down to solving a series of one-dimensional problems for which efficient solution methods exist. Derivation of the approximation can be easily automated and the HYSDEL variant of the hybrid approximation can be generated, hence allowing for subsequent control synthesis based on the hybrid model. A MATLAB toolbox which implements the proposed approximation strategy was described as well. The toolbox allows user to enter data either via command line, or by using a graphical user interface. An experimental version of the toolbox is available for free download at http://www.kirp.chtf.stuba.sk/~sw/.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07.

REFERENCES

Adjiman, C.S., Androulakis, I.P., Maranas, C.D., and Floudas, C.A. (1996). A global optimization method, αBB for process design. *Computers and Chemical Engineering*, 20, 419–424.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Branicky, M. (1995). *Studies in hybrid systems: modeling, analysis, and control.* Ph.D. thesis, LIDS-TH 2304, Massachusetts Institute of Technology, Cambridge, MA.

Chachuat, B., Singer, A.B., and Barton, P.I. (2006). Global methods for dynamic optimization and mixed-integer dynamic optimization. *Ind. Eng. Chem. Res.*, 45(25), 8373–8392.

De Schutter, B. and Van den Boom, T. (2001). On model predictive control for max-min-plus-scaling discrete event systems. *Automatica*, 37(7), 1049–1056.

Ferrari-Trecate, G. (2005). Hybrid Identification Toolbox (HIT). Available from http://www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/HIT_toolbox.html.

Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2001). Identification of Piecewise Affine and Hybrid Systems. In *Proc. on the American Control Conference*, 3521–3526. Arlington (VA), USA.

Heemels, W.P.M., De Schutter, B., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.

Heemels, W., Schumacher, J., and Weiland, S. (2000). Linear complementarity systems. *SIAM Journal on Applied Mathematics*, 60(4), 1234–1269.

Kvasnica, M. and Herceg, M. (2010). HYSDEL 3.0. Available from http://kirp.chtf.stuba.sk/~kvasnica/hysdel3/.

Papamichail, I. and Adjiman, C.S. (2004). Global optimization of dynamic systems. *Computers and Chemical Engineering*, 28, 403–415.

Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40, 37–50.

Sontag, E.D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2), 346–358.

Torrisi, F. and Bemporad, A. (2004). HYSDEL — A tool for generating computational hybrid models for analysis and synthesis problems. *IEEE Transactions on Control Systems Technology*, 12, 235–249.

Williams, H. (1993). *Model Building in Mathematical Programming.* John Wiley & Sons, Third Edition.

PIDTOOL 2.0 – Software for Identification and PID Controller Tuning

J. Oravec * M. Bakošová *

* *Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics, Radlinského 9, 812 37 Bratislava, Slovakia*
Tel: +421 2 59325364; e-mail: juraj.oravec@stuba.sk

Abstract: The main aim of this paper is to present a new version of software for PID controller tuning called *PIDTOOL 2.0*. The software represents a user friendly tool for simple step–response–based identification of a process model, fast PID controller tuning, and effective checking the quality of control. It has been developed in the MATLAB–Simulink programming environment using its graphic user interface and can be used as useful and visual software for teaching purposes. In *PIDTOOL 2.0*, user can easily change a language of the graphic user interface. Nowadays, there is a possibility to choose between English and Slovak.

Keywords: PID controller, controller tuning, identification, filtration, graphic user interface

1. INTRODUCTION

The aim of this paper is to present a new version of software for PID controller tuning called *PIDTOOL 2.0* (Oravec (2010), Bakošová and Oravec (2009)). It has been developed at the Institute of Information Engineering, Automation, and Mathematics of the FCFT STU in Bratislava (Čemanová (2007), Oravec (2010)) in the MATLAB – Simulink programming environment and uses its graphic user interface (GUI).

As PID controllers belong to the most used types of controllers in industry (Åström and Hägglund (1995)), *PIDTOOL 2.0* is oriented mainly on PID controller tuning. The software enables to tune PID controllers using various analytical and experimental methods, and new methods described in Xue and Chen (2008), Åström and Hägglund (1995), Bakošová and Fikar (2008), Mikleš and Fikar (2007), Ogunnaike and Ray (1994), Vítečková and Víteček (2006), Lošonský (2006), Weng et al. (1995), Shaw (2006), Bakošová et al. (2003), Kubík and Kotek (1983), and Šulc and Vítečková (2004) were added. If a transfer function of the controlled process is unknown, this software enables to identify the controlled process from its step response (Fikar and Mikleš (1999)). The identified step response can be either damped periodic or aperiodic. The step–response data can be set directly or loaded from the data file. If noisy step–response data have been loaded, designed software enables to run filtration. Properties of the tuned controller can be judged visually and analytically, as *PIDTOOL 2.0* displays simulated control response, time behavior of the manipulated variable and values of various integral performance indexes. So, it is easy to compare several closed–loop step responses generated using various PID controllers with different values of set-points, disturbances and constraints on manipulated variables.

The software *PIDTOOL 2.0* can be used especially for teaching purposes in laboratory exercises oriented on process control. It can be also useful for those who need to identify controlled process from its step–response data, to filter noisy data, to tune PID controllers or to compare various types of control algorithms with simple PID control.

2. PIDTOOL 2.0

PIDTOOL 2.0 solves two basic problems, identification and controller tuning (Fig. 1).

PIDTOOL 2.0 enables to identify a controlled process from its step response. The software distinguishes identification from an aperiodic or a damped periodic step response.

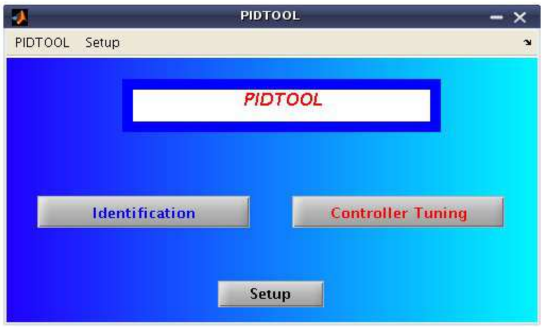


Fig. 1. Basic window of *PIDTOOL 2.0*

The Strejc method (Bakošová and Fikar (2008), Mikleš and Fikar (2007), Lošonský (2006)) is applied for identification from the aperiodic step response and the method described in Mikleš and Fikar (2007) is used for identification from the damped periodic step response. The result of identification is a controlled process model described by the transfer function (1) for aperiodic or (2) for damped periodic step response

$$G(s) = \frac{K}{(Ts+1)^n} e^{-Ds} \qquad (1)$$

$$G(s) = \frac{K}{T^2s^2 + 2\xi Ts + 1} e^{-Ds} \qquad (2)$$

where n is the order of the system, K is the gain, T is the time constant, ξ is the damping coefficient and D is the time delay.

The identification can be simply started using button *Identification* located in the basic window (Fig. 1). The next window (Fig. 2) offers three identification possibilities.

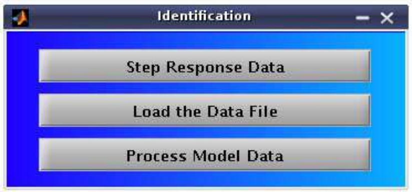


Fig. 2. Basic window of identification

The first button *Step Response Data* enables to identify the controlled process directly from data obtained from the measured and recorded step response. The second button *Load the Data File* (Fig. 2) opens the new window (Fig. 3), where user can comfortably find out a required data file containing recorded step–response data. The considered structure of the data file is as follows, the first column vector represents a time and the second column vector represents associated measured values of output variable. *PIDTOOL 2.0* enables to load the data file which includes also the third column vector of values of manipulated variable (Fig. 4). If several step responses are included in the loaded data file, they are automatically recognized and the nominal step response is evaluated (Fig. 5). It enables to reach the nominal transfer function of identified non-linear controlled process. This possibility makes the identification from step response using this software even user–friendlier.

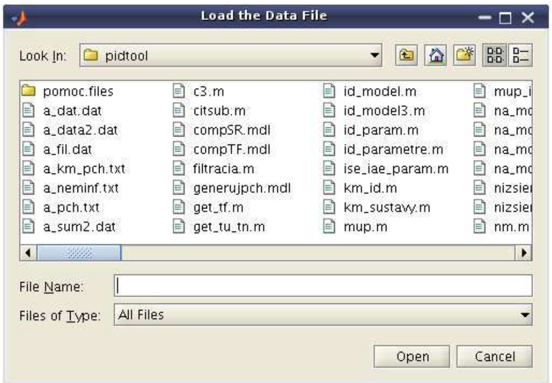


Fig. 3. Window for loading the datafile

In the next window, there is a choice of data processing (Fig. 6). To obtain the aperiodic model of controlled process, user can directly use the button *Identification*.

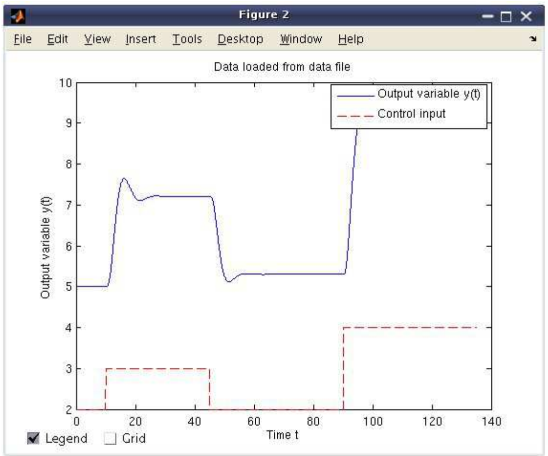


Fig. 4. Loaded data of step response

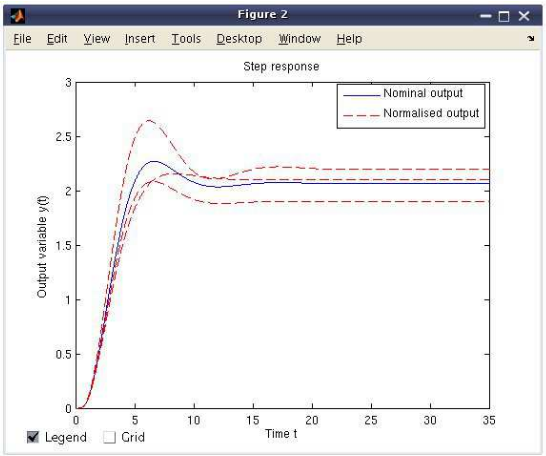


Fig. 5. Nominal and normalized step responses

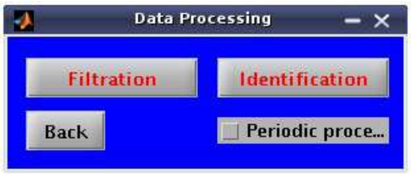


Fig. 6. Choice of processing of loaded data

When the damped periodic model is required, user can simply activate the checkbox *Periodic process* and then use the button *Identification.* If the controlled process has been identified using the Strejc method, the tangent to the step response is also depicted and its equation is given (Fig. 7). In the new window, the parameters of model (1) or (2) of identified process are shown (Fig. 8, Fig. 9).

PIDTOOL 2.0 enables to use button *Identification Tuning* (Fig. 10) to receive the transfer function, which generates the step response, that covers the original one more precisely. In the new window (Fig. 11) the step response of identified transfer function can be simply modified by changing the slope of its tangent (Lošonský (2006)). The new parameters of identified transfer function are directly shown. It helps to check whether the identified transfer function has still required properties, e.g. the order n. If the loaded data are noisy, the user can use the filtration before identification, simply using the button *Filtration* (Fig. 6). Then the new window for filtration is opened

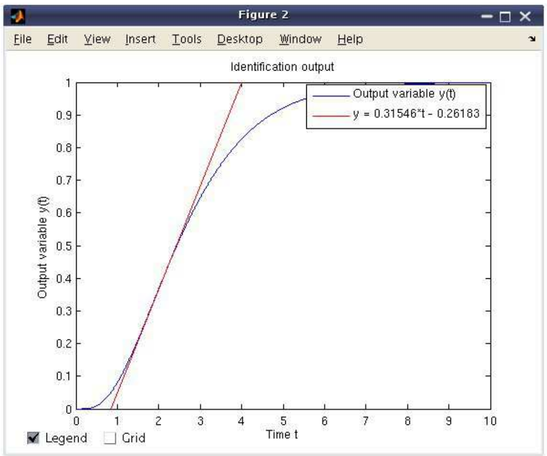
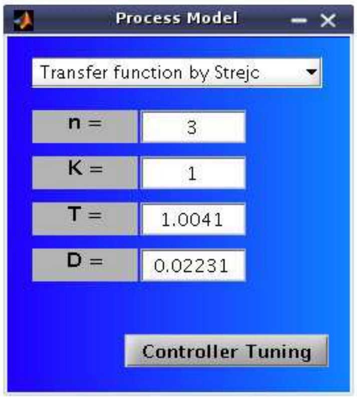


Fig. 7. Step response of identified process



Transfer function by Strejc

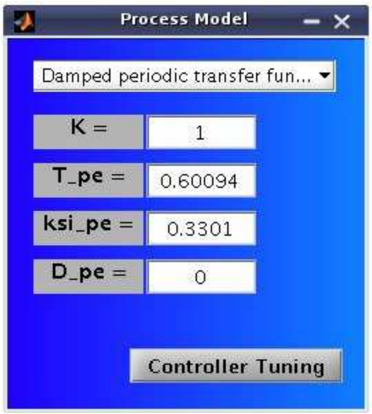
n = 3

K = 1

T = 1.0041

D = 0.02231

Fig. 8. Identified parameters of aperiodic system



Damped periodic transfer fun...

K = 1

T_pe = 0.60094

ksi_pe = 0.3301

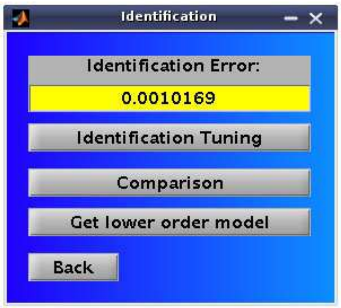
D_pe = 0

Fig. 9. Identified parameters of damped periodic system

(Fig. 12). By using the button *Save* (Fig. 12), user can simply store reached filtered data into the new data file for later usage. After filtration, identification can be started.

The third button *Process Model Data* (Fig. 2) enables to identify process model with required properties of transfer function. Using this button, the new window shown in Fig. 13 will be opened. In this window, the parameters of the model described by the transfer function (3) can be simply set.

$$G(s) = \frac{Num(s)}{Den(s)} e^{-Ds} \qquad (3)$$



Identification Error:

0.0010169

Fig. 10. Additional options of identification

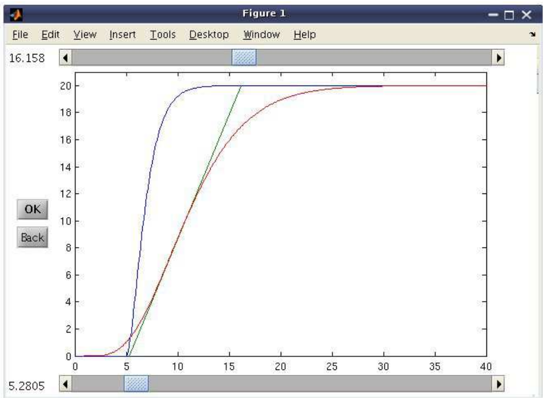
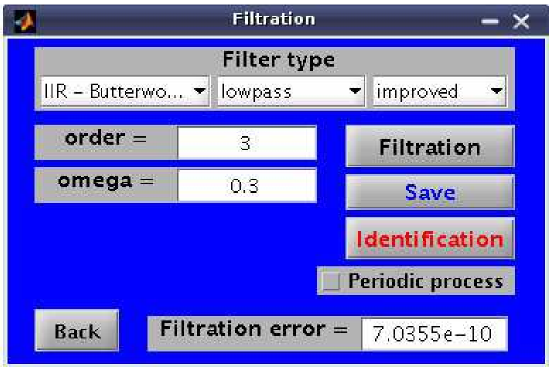


Fig. 11. Window for identification tuning



Filter type

IIR – Butterwo... lowpass improved

order = 3

omega = 0.3

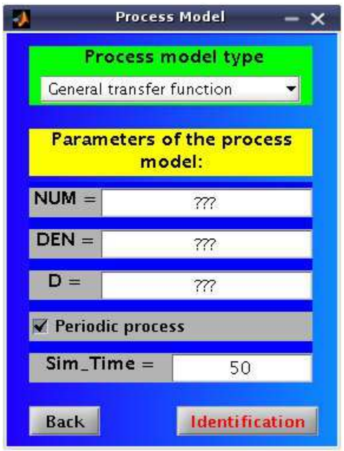
□ Periodic process

Filtration error = 7.0355e-10

Fig. 12. Window of filtration

If the checkbox *Periodic process* has been activated (Fig. 13), given model (3) will be approximated by the transfer function (2), otherwise by the transfer function (1). The approximation is started using the button *Identification* (Fig. 13). This approximation of the given model can be useful in the case when chosen PID controller tuning method requires controlled model described by the transfer function (1) or (2).

The software *PIDTOOL 2.0* is oriented mainly on PID controller tuning. PID controllers can be designed for controlled process models with either damped periodic or aperiodic step responses described by the transfer functions (1) or (2). To run direct controller tuning, user can use the button *Controller Tuning* located in the basic window (Fig. 1), or use this button after identification (Fig. 9, Fig. 8). Then, a window is opened where the user can choose a required type of PID controller and a type of a tuning method (Fig. 14).



Process model type: General transfer function

Parameters of the process model:

NUM = ???

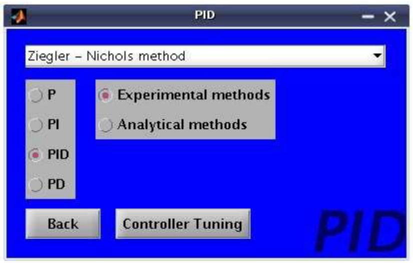
DEN = ???

D = ???

☑ Periodic process

Sim_Time = 50

Fig. 13. Process model data

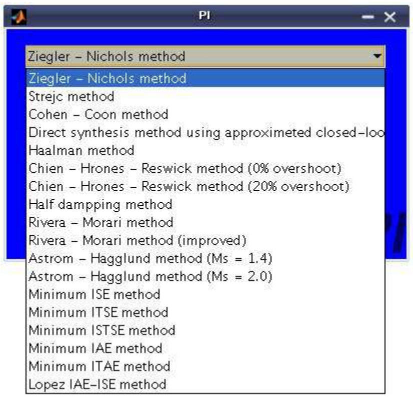


Ziegler – Nichols method

○ P
○ PI
● PID
○ PD

● Experimental methods
○ Analytical methods

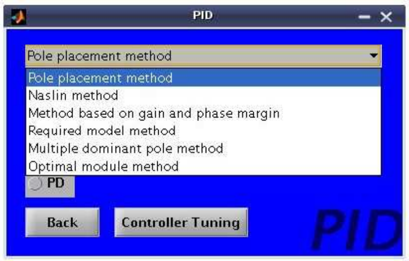
Fig. 14. Window for PID controller tuning



Ziegler – Nichols method

- Ziegler – Nichols method
- Strejc method
- Cohen – Coon method
- Direct synthesis method using approximeted closed–loo
- Haalman method
- Chien – Hrones – Reswick method (0% overshoot)
- Chien – Hrones – Reswick method (20% overshoot)
- Half dampping method
- Rivera – Morari method
- Rivera – Morari method (improved)
- Astrom – Hagglund method (Ms = 1.4)
- Astrom – Hagglund method (Ms = 2.0)
- Minimum ISE method
- Minimum ITSE method
- Minimum ISTSE method
- Minimum IAE method
- Minimum ITAE method
- Lopez IAE–ISE method

Fig. 15. List of experimental methods used for PI controller tuning

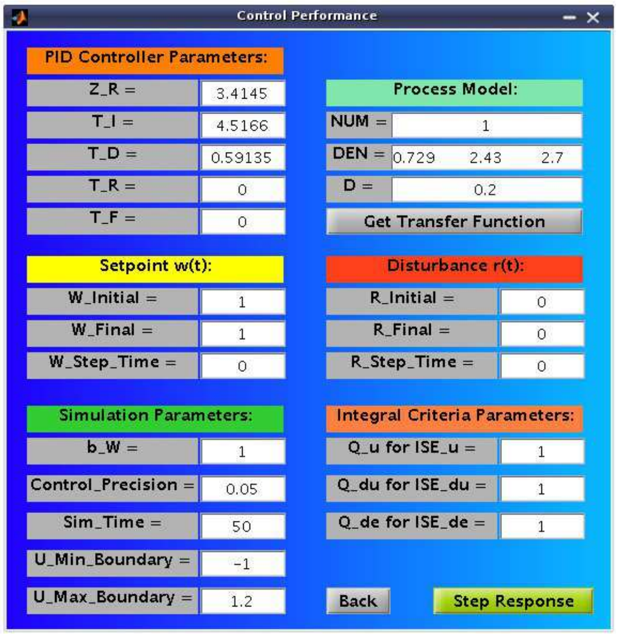


Pole placement method

- Pole placement method
- Naslin method
- Method based on gain and phase margin
- Required model method
- Multiple dominant pole method
- Optimal module method

Fig. 16. List of analytical methods used for PID controller tuning

It is possible to choose a P, PI, PID or PD controller. For simpler handling, the methods for controller tuning are divided into two main groups: analytical and experimental methods. Various types of analytical (Fig. 16) and experimental methods (Fig. 15) can be used for controller tuning. Calculated parameters of the tuned controller are shown in the new window (Fig. 17), where Z_R is the gain, T_I is the reset time and T_D is the derivative time of the controller.



PID Controller Parameters:

Z_R = 3.4145
T_I = 4.5166
T_D = 0.59135
T_R = 0
T_F = 0

Process Model:

NUM = 1
DEN = 0.729 2.43 2.7
D = 0.2

Setpoint w(t):

W_Initial = 1
W_Final = 1
W_Step_Time = 0

Disturbance r(t):

R_Initial = 0
R_Final = 0
R_Step_Time = 0

Simulation Parameters:

b_W = 1
Control_Precision = 0.05
Sim_Time = 50
U_Min_Boundary = -1
U_Max_Boundary = 1.2

Integral Criteria Parameters:

Q_u for ISE_u = 1
Q_du for ISE_du = 1
Q_de for ISE_de = 1

Fig. 17. Window for evaluating the quality of control

In the top right part of the opened window (Fig. 17), the parameters of the transfer function of the controlled process are also shown. These parameters can be modified and the tuned controller can be so tested in the presence of model uncertainty. The properties of the closed loop with the tuned controller can be judged by simulation of control. The standard control law (4) is supposed as a default control law. Setting the parameters b_W, T_R and T_F enables to use improved form of control law. The proportional part of control law is then modified by the value of the parameter b_W. Setting this parameter enables to change the weight of set–point in the control error evaluation (5). The non zero value of the parameter T_R modifies the integral part of control law (6). Setting this parameter enables to prevent integral windup. The parameter T_F modifies the derivative part of control law (7). The non zero value of this parameter represens a filter of derivative part of control law to obtain the proper transfer function of the derivative part of the controller.

$$u(t) = Z_R e(t) + \frac{Z_R}{T_I}\int\limits_0^t e(t)dt + Z_R T_D \frac{de(t)}{dt} \tag{4}$$

$$u_P(t) = Z_R\left(b_W w(t) - y(t)\right) \tag{5}$$

$$u_I(t) = \frac{Z_R}{T_I}\int\limits_0^t \left(e(t) - \frac{T_I}{Z_R}T_R\left(u(t) - u_{SAT}(t)\right)\right)dt \tag{6}$$

$$u_D(t) = \frac{Z_R T_D s}{1 + \frac{T_D}{T_F}s} \tag{7}$$

Using the button *Step Response* (Fig. 17) runs quality evaluation of control. The set–point tracking and the disturbance rejection can be simulated in the presence of boundaries on the control input. For simpler handling, the parameters have preset default values. In the case, the manipulated variable is constrained, user can compare the closed-loop step responses and the time behaviors of the manipulated variable before and after the saturation. After simulation, the closed–loop step responses (Fig. 18) and the time behaviors of the manipulated variable (Fig. 19) are shown. In the case the legend overlaps the displayed graph (Fig. 18), it is possible to deactivate the checkbox *Legend* and to hide the shown legend. Using the checkbox *Grid* leads to displaying the grid of the shown graph. Both of these possibilities are included in all displayed graphs, generated by this software. The quality of control (Fig. 20) can by also judged by calculating several integral performance indexes (Mikleš and Fikar (2007), Bakošová et al. (2003), Xue and Chen (2008), Oravec (2010), Čemanová (2007)).

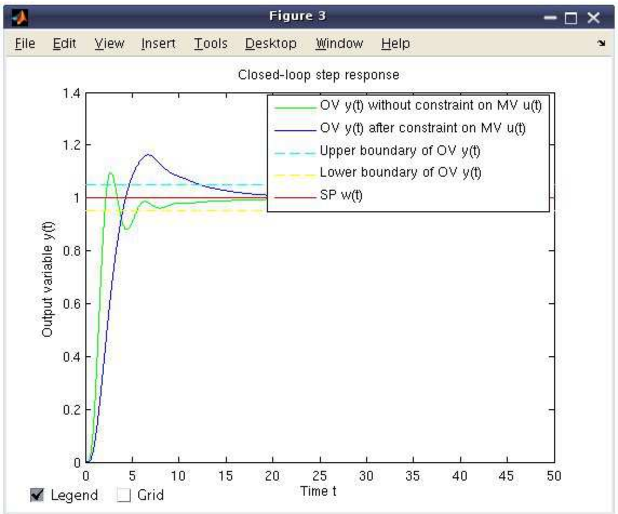


Fig. 18. Closed–loop step response

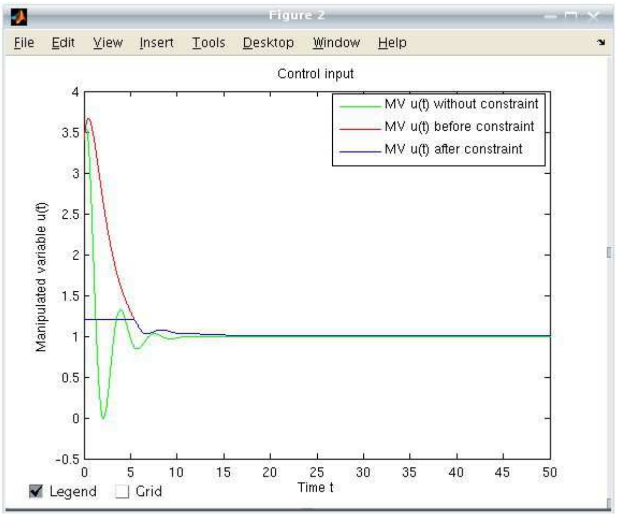
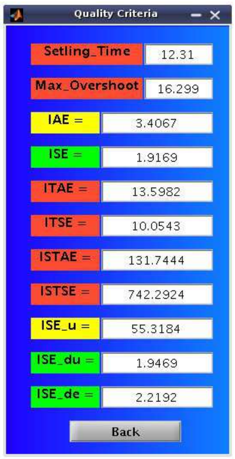


Fig. 19. Control input

Using the possibilities of window shown in Fig. 17, it is easy to compare several step responses and values of performance indexes reached with different values of set–points, disturbances and constraints on the manipulated variables. The values of all the parameters, which are necessary for simulations are stored by *PIDTOOL 2.0*. Calculated values of performance indexes are also stored. These stored data can be simply shown by using *Setup/Show results* located in basic window (Fig. 21). The stored data are transformed into *html*-file. The stored data are shown in simple summary table with date and time of simulation. This new ability can be helpful in the case, when many simulations at different conditions have been evaluated and user wants to compare obtained results to make decision, which controller is the most suitable for control.


Quality Criteria

Setling_Time	12.31
Max_Overshoot	16.299
IAE =	3.4067
ISE =	1.9169
ITAE =	13.5982
ITSE =	10.0543
ISTAE =	131.7444
ISTSE =	742.2924
ISE_u =	55.3184
ISE_du =	1.9469
ISE_de =	2.2192

Fig. 20. Values of performance indexes

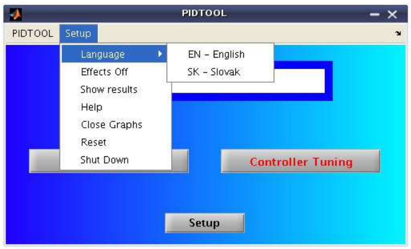


Fig. 21. Setup menu of *PIDTOOL 2.0*

It is easy to change a language of graphic user interface. Actually, there is a possibility to choose between English and Slovak language (Fig. 21). Also other languages can be simply added into the software *PIDTOOL 2.0*.

3. CONCLUSION

The software *PIDTOOL 2.0* has been developed in the MATLAB – Simulink programming environment using its Graphical User Interface and offers to the user a comfortable and visual environment for fast identification, simple PID controller tuning and effective evaluating the quality of control in various conditions. The values necessary for evaluating the quality of control and calculated quality criteria are stored and can be displayed in a simply summary table. Using this new ability helps to decide, which controller is the most suitable for control. This software has been tested by students at the FCFT in two courses; in the course Process Dynamics and Control that is taught in the first year of the Master study and the course Integrated Control in the Process Industry that is taught in the third year of the Bachelor study. Using the software is limited only for a teaching purposes. To obtain the software contact the authors.

There exist various software applications, which enables to design the PID controller, for example see (Lošonský (2006)) or (Schlegel and Čech (2004)). The comparison of the available applications will be subject of our next work.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency VEGA of the Slovak Republic under the grants 1/0537/10.

REFERENCES

Bakošová, M. and Fikar, M. (2008). *Process Control (in Slovak).* STU Press, Bratislava.

Bakošová, M. and Oravec, J. (2009). Software for PID controller tuning. *Proceedings of the 17th Interantional Conference on Porcess Control'09, FCFT SUT in Bratislava*, 524–527.

Bakošová, M., Ľ. Čirka, and Fikar, M. (2003). *Automatic control fundaments (in Slovak).* Vydavateľstvo STU, 1st ed. edition.

Fikar, M. and Mikleš, J. (1999). *System identification (in Slovak).* STU Press, Bratislava.

Kubík, S. and Kotek, Z. (1983). *Automatic control theory (in Czech).* SNTL, 1st ed. edition.

Lošonský, J. (2006). *System for support of control loops (in Slovak).* Master's thesis, Trnava: MTF SUT in Bratislava.

Mikleš, J. and Fikar, M. (2007). *Process Modelling, Identification, and Control.* Springer Verlag, Berlin Heidelberg.

Ogunnaike, B. and Ray, W.H. (1994). *Process dynamics, modeling and control.* Oxford University Press, New York.

Oravec, J. (2010). *Design of a Program System for Controller Tuning (in Slovak).* Master's thesis, FCFT SUT in Bratislava.

Åström, K. and Hägglund, T. (1995). *PID controllers: theory, design and tuning.* Triangle Park, 2nd ed. edition.

Schlegel, M. and Čech, M. (2004). Internet PID controller design : www.pidlab.com. *IBCE 04*, 1–6.

Shaw, J.A. (2006). *The PID Control Algorithm [online].* http://www.docstoc.com/docs/20312583, 2nd ed., version 2.2 edition.

Čemanová, L. (2007). *Design of GUI for controller tuning (in Slovak).* Master's thesis, FCFT SUT in Bratislava.

Vítečková, A. and Víteček, M. (2006). *Automatic control fundamentals (in Czech).* TU Ostrava.

Šulc, B. and Vítečková, M. (2004). *Theory and praxis of control design (in Czech).* ČVUT Press, Praha.

Weng, K.H., Chang, C.H., and Lisheng, S.C. (1995). Tuning of PID Controllers Based on Gain and Phase Margin Specifications. *Pergamon, Automatica*, 31(3), 497–502.

Xue, D. and Chen, Y.Q. (2008). *Analysis and Design with MATLAB (Advanced in Design and Control).* Society for Industrial Mathematics, 1st ed. edition.

SEGMENTATION OF COLOUR REGIONS FROM PRINTING SHEETS

Miroslav Fribert

University of Pardubice, Studentská 95, 53210 Pardubice

(Tel. +420 732 171 286, miroslav.fribert@upce.cz)

Abstract: The print characteristics of optical density, dot area and ink trapping are suitable to control of printing process. To evaluate the print characteristics on the basis of scanned micro-samples of printing sheets, it is necessary to carry out the segmentation of colour regions, printed with process inks. The accuracy of the evaluation of these print characteristics strongly depends on the sufficient accurate segmentation of colour regions involved in the print sample. This paper describes the thresholding method of segmentation, combined with edge detection between colour regions.

Keywords: Colour print, control of print, print characteristics, region segmentation

1. Introduction

The control of optimal colouring of printing sheets in the inking unit is the standard function, which is implemented in the printing press. Utilization of this function in the printing process needs the measuring of printing parameters optical density, dot area, ink trapping and colour differences. The method described in this paper enables measuring of theses parameters from segmented colour regions in halftone areas of printing sheets. The application of this method in printing process control increases the quality of the print.

Colour publications are printed with using four process inks – yellow (Y), magenta (M), cyan (C) and black (K) by various technologies of industrial press, such as digital screening - Green (1995). The results of such printing method are halftone dots of various sizes printed with these four process inks (fig. 1).

We can measure and evaluate the basic printing characteristics, which determine quality of the print, with various methods. How to obtain them from the halftone colour samples is described in this work.

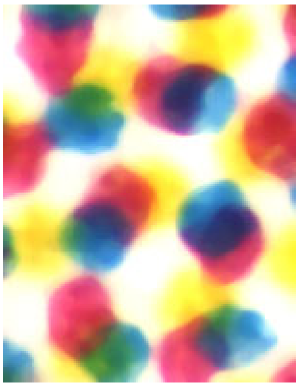


Fig. 1 Three-colour print sample with digital screening (scanned with zoom about 100)

The basic print characteristics used to examine of print quality, measured in the solid and halftoned areas by densitometry are - Mortimer (1991):

Optical density of the ink layer (determines quality of ink printing on the paper)

$$D = \log \frac{L_W}{L_R} \qquad (1)$$

where L_W is the intensity of light reflected from the white region of the paper, and where L_R is the intensity of light reflected from the process ink region printed on the paper.

Geometrical Dot Area in percents

$$\Phi_G = \frac{S_{PD}}{S_T} \qquad (2)$$

where S_{PD} is the surface of the printed dot, and where S_T is total surface of the halftone cell.

Ink trapping (determines quality of ink printing to ink layer printed first)

$$T = \frac{D_{12} - D_1}{D_2} \qquad (3)$$

where D_{12} is the density of the overprint, where D_1 is the density of the first printed colour, and where D_2 is the density of the second printed colour.

These parameters can be evaluated also from the regions of halftone dots of colour samples scanned with a CCD camera. The usual practice is to perform the segmentation of an individual process colour and their overprint regions from the colour sample, and then evaluate the print characteristics by the method of the image analysis. In this case the print characteristics are: optical density inside the segmented regions, ink trapping evaluated from these densities and area of segmented region – Fribert (2002).

Optical densities inside halftone dots of cyan, magenta and yellow, evaluated by image analyses method are:

$$D_c = \log\frac{L_w N_p}{\sum_{i,j} r_{i,j}} \qquad D_m = \log\frac{L_w N_p}{\sum_{i,j} g_{i,j}} \qquad D_y = \log\frac{L_w N_p}{\sum_{i,j} b_{i,j}} \tag{4}$$

where $L_w = \frac{\sum_{i,j}(r_{i,j} + g_{i,j} + b_{i,j})}{3N_w}$ is light intensity reflected from the white reference sample, N_p is number of pixels inside segmented colour region, N_w is number of pixels inside white reference sample and where $r_{i,j}$, $g_{i,j}$, $b_{i,j}$ are colour components of light reflected from the segmented colour region or from white reference sample.

The crucial point of successful evaluation of print characteristics is the accurate segmentation of the colour regions in the print sample. There are many methods, how to improve accuracy results of segmentation – Sonka et al. (1993), Russ (1999). The combination of thresholding grey level image, gained from separation process, and edge detection applied on original colour image, can be used for this purpose.

2. Separation of colour regions

This proposed segmentation method is based in the first step on the process of colour separation of the print sample image into grey level image. After separation has the region of separated colour minimal brightness. The separation process is based on the evaluation of colour differences between colours included in the print sample.

$$D_{i,j} = \sqrt{(R_{i,j} - R_r)^2 + (G_{i,j} - G_r)^2 + (B_{i,j} - B_r)^2} \tag{5}$$

where D_{ij} is colour difference in the pixel i,j, $R_{i,j}$, $G_{i,j}$, $B_{i,j}$ are components of the pixel i, j involved in the sample, and R_r, G_r, B_r are the reference values of the colour, that is to be separated from the sample.

There was proposed equation for the gray level brightness *BR* of the separated colour – Fribert (2005). The function *MAX* is applied for all possible values *i, j* in the image:

$$BR_{i,j} = 255 \cdot \frac{\sqrt{(R_{i,j} - R_r)^2 + (G_{i,j} - G_r)^2 + (B_{i,j} - B_r)^2}}{MAX\sqrt{(R_{i,j} - R_r)^2 + (G_{i,j} - G_r)^2 + (B_{i,j} - B_r)^2}} \tag{6}$$

On the fig. 2 are presented images of blue and yellow separations created by application of eq. 6. The motives of this colours has the minimal brightness in comparison with other colours.

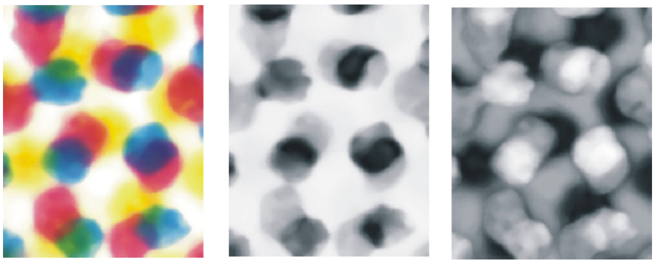


Fig. 2 Separations of blue and yellow regions

The segmentation of specific colour regions is based on using the monochromatic images obtained after described separation process.

The simplest segmentation method in the image of the grey-level separation is the thresholding. As was proven by many acts of measuring and evaluating, the optimal threshold always exists for images gained by the described method of the separation - Böhm (2001), Fribert (2002). The usual determination of threshold value from the brightness histogram of the separation is impossible in most cases, because of the inexpressive maximum or not-sharp valley in this histogram (fig. 3). Then the proposed method of the combination of the thresholding and edge detection is suitable for this purpose.

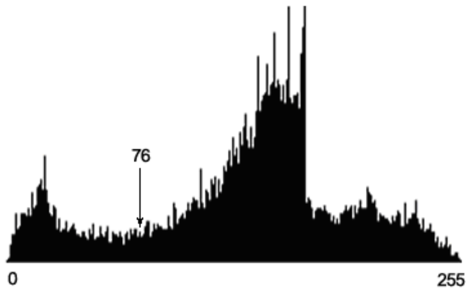


Fig. 3 Histogram of yellow separation and optimal threshold, the value 76 was gained by proposed method.

This method is based on the assumption that the border of the particular colour region lies in the middle of the edge of the colour changeover from the first to the second colour. This border can be obtained by the edge detection process. We can calculate significant edges of the specific colour from the original colour image by using of any standard edge detectors. The coordinates of the edge pixels determine the various threshold values in the image of separation. The average of these threshold values is the optimal threshold for the segmentation of the specific colour region.

3. Edge detection of colour regions

The edge detection method used in this work is based on the colour differences between the specific colour region and its neighbourhood. The statistical parameter variance of the colour value Hue in this neighbourhood indicates the location and size of the edge. For the correct segmentation it is necessary to determine edges only on the borders of colour

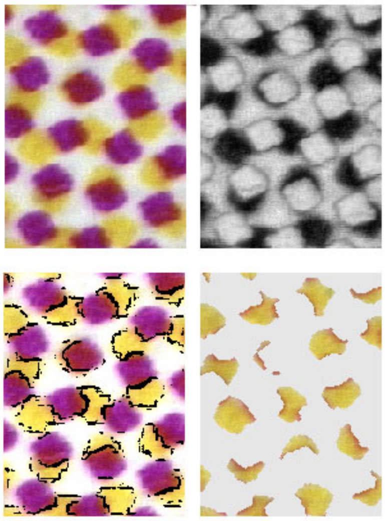


Fig. 4 Original image, its yellow separation, yellow edges and yellow regions segmented with optimal threshhold

regions, that are to be segmented. On the fig. 4 are presented the original colour image, the image of its yellow separation, original image with significant edges of yellow regions and the resulting image of the yellow region segmentation.

The pixel coordinates gained from the image of yellow edges were used as the pointers to the image of yellow separation to evaluate optimal threshold. As a result of segmentation process is the segmentation mask and after masking of the original image with this mask we get the resulting segmentation of yellow regions.

On the fig. 5 are presented images of cyan-magenta sample and the separations of cyan, magenta and blue regions. The corresponding optimal threshold values for cyan, magenta and blue separations, calculated with help of the edge detection process, were 78, 88 and 74.

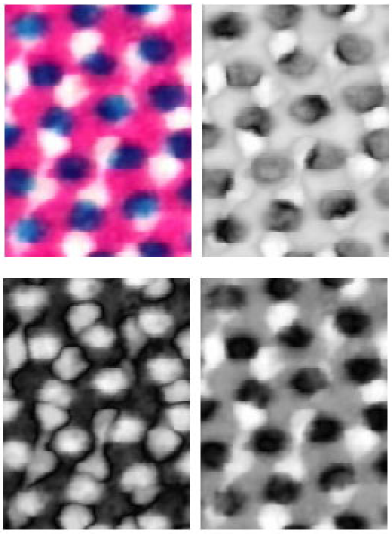


Fig. 5 Separations of colour regions included in the cyan-magenta sample

4. Accuracy of segmentation

The effect of the segmentation error can be demonstrated on evaluation of colour regions areas from the two-colour samples. In these two-colour samples were segmented all colours including white background. The sum of the colour regions areas would have to be 100 %. The difference from the 100 % determines the segmentation error.

On the fig. 6 are presented the images of segmentations gained from previous cyan-magenta sample. These segmentation images were processed by thresholding of separation images with optimal threshholds and with subsequent masking of the original image. The sum of all segmented colour region areas, including white, was 97.11% segmentation error is then in this case 2.89 %.

The proposed method was check on three tested sheets with two-colour halftone motives Cyan-Magenta, Cyan-Yellow and Magenta-Yellow with various values of the parameter dot area. There were scanned and processed by the proposed method 5 image samples from each sheet. The sheets were printed on the press Heidelberg Quickmaster QM-42 in the department of graphics arts of the University of Pardubice.

By verifying of the proposed method the achieved maximal error was less then 5% in the case of the offset print samples. In the offset print, in comparison e.g. with flexography print, the edges of halftone dots are blurred.

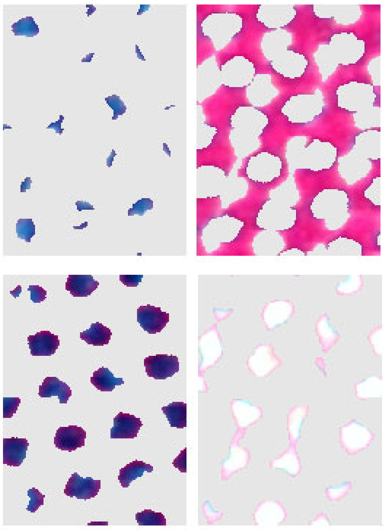


Fig. 6 Segmentations of cyan, magenta blue and white regions

4. Conclusions

There were processed many image halftone samples from printing sheets in the department of graphic arts. The segmentation has been performed usually with the standard thresholding. The accuracy of segmentation was always under 10% in the case of the offset print. The proposed method is approximately twice more accurately.

The improving the segmentation accuracy has the positive effect on the successful evaluation of all important print characteristics, especially of the dot area parameter. The dot area parameter enables the measuring of the dot gain value (spreading of halftone dots), which causes the main distortion in the colour reproduction - colour shift.

The contemporary production printing needs measure, evaluate and control the print characteristics with sufficient accuracy. The method described in this work can benefit to this purpose, but it can be used also in the other areas of image processing.

References

Mortimer, A. (1991). Colour Reproduction in the Printing Industry. Pira International, Surrey.

Sonka, M., Hlavac, V., Boyle, R. (1993). Image Processing and Machine Vision. Chapmann & Hall, London.

Sangwine, S. J., Horne (1998). The colour Image Processing Handbook. Chapman & Hall, London .

Green, P. (1995). Understanding Digital Colour. Graphic Arts Technical Foundation.

Russ, J. C. (1999). The Image Processing Handbook. CRC Press, IEEE Press.

Fribert, M. (2002) Calculation of the dot area with image analysis method. Sci. pap. Univ. Pardubice Ser. A 8, 159-165.

Böhm, J. (2001) Měření síťových tónových hodnot dvoubarevných vzorků tisku. Bakalářská práce, Univerzita Pardubice .

Fribert, M. (2003) Separation Model of Colour Regions in a Halftone Print. Computers & Graphics 27, 801-806.

Fribert, M. (2005). Segmentation of colour regions from two colour halftone prints. Poster proceedings WSCG 2005, Plzeň, Czech republic.

Disturbance Decoupling of Discrete-time Nonlinear Systems by Static Measurement Feedback

A. Kaldmäe Ü. Kotta

Institute of Cybernetics at TUT, Akadeemia tee 21, 12618 Tallinn, Estonia
fax : +372 620 4151 and e-mail : arvo@cc.ioc.ee, kotta@cc.ioc.ee

Abstract: This paper addresses the disturbance decoupling problem (DDP) for nonlinear systems extending the results for continuous-time systems into the discrete-time case. Sufficient conditions are given for the solvability of the problem. The notion of the rank of a one-form is used to find the static measurement feedback, that solves the DDP whenever possible. Moreover, necessary and sufficient conditions are given for single-input single-output systems as well as for multi-input single-output systems under the additional assumption.

Keywords: Nonlinear systems, discrete-time systems, disturbance decoupling, static measurement feedback.

1. INTRODUCTION

The disturbance decoupling problem (DDP) for discrete-time nonlinear control system by state feedback has been addressed in many papers; see Aranda-Bricaire and Kotta (2004 2001); Fliegner and Nijmeijer (1994); Grizzle (1985); Kotta and Nijmeijer (1991); Monaco and Normand-Cyrot (1984). Most papers extend the known results for continuous-time systems (see for example Nijmeijer and van der Schaft (1990); Conte et al. (2007); Isidori (1995)) into the discrete-time domain and in all these papers the control system is described by smooth or analytic difference equations. However, there are no papers that address the DDP for discrete-time nonlinear control systems using the output feedback except that of by Shumsky and Zhirabok (2010) (see also Kotta and Mullari (2010)) and Kotta et al. (2011). In Shumsky and Zhirabok (2010) the controlled output is a vector function of the measured output, having possibly less components than the measured output itself. Therefore, the above solution may be considered only as a partial solution. The paper by Kotta et al. (2011) provides a full algorithmic solution for the problem using the dynamic feedback. In both papers the novel algebraic approach, called the algebra of functions, is applied, see Zhirabok and Shumsky (2008).

As for continuous-time nonlinear control systems there exist also only a few papers addressing the problem, see Pothin et al. (2002); Isidori et al. (1981); Xia and Moog (1999); Andiarti and Moog (1996). The paper by Pothin et al. (2002) studies the problem using a static measurement feedback, and in Isidori et al. (1981) the feedback considered is restricted to pure dynamic measurement feedback, whereas the other two papers focus on the dynamic measurement feedback.

The goal of this paper is to extend the results of Pothin et al. (2002) for discrete-time nonlinear control systems.

2. PROBLEM STATEMENT

Consider a discrete-time nonlinear control system

$$\begin{aligned} x(t+1) &= f(x(t), u(t), w(t)) \\ y(t) &= h(x(t)) \\ z(t) &= k(x(t)), \end{aligned} \tag{1}$$

where the state $x(t) \in \mathbb{R}^n$, the control input $u(t) \in \mathbb{R}^m$, the disturbance input $w(t) \in \mathbb{R}^\nu$, the output to be controlled $y(t) \in \mathbb{R}$ and the measured output $z(t) \in \mathbb{R}^\mu$. Assume that f, h and k are meromorphic functions of their arguments.

Let $\mathcal{K}^*$ denote the inversive field of meromorphic functions in variables $x(t)$, $u(t)$, $w(t)$ and a finite number of their (independent) forward- and backward shifts. Note that not all the variables are independent because of the relationships defined by (1) and in the computations the dependent variables have to be expressed via the independent ones. For example, $x(t+1)$ has to be replaced by $f(x(t), u(t), w(t))$. See Aranda-Bricaire et al. (1996) for the details how to construct $\mathcal{K}^*$.

Define the vector spaces $\mathcal{X} = \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}x(t)\}$, $\mathcal{Z} = \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}z(t)\}$, $\mathcal{U} = \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}u(t+k), k \geq 0\}$, $\mathcal{W} = \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}w(t+k), k \geq 0\}$ and $\mathcal{E} = \mathcal{X} + \mathcal{U} + \mathcal{W}$.

Definition 1. (Aranda-Bricaire et al. (1996)) The relative degree r of the output $y(t)$ is defined by

$$r := \min\{k \in \mathbb{N} | \mathrm{d}y(t+k) \notin \mathcal{X}\}.$$

If such an integer does not exist, then define $r := \infty$.

The static measurement feedback of the form $u(t) = F(z(t), v(t))$ is called regular if F is invertible with respect to $v(t)$, i.e. if there exists an inverse function $\alpha := F^{-1}$ such that $v(t) = \alpha(z(t), u(t))$.

Problem Statement. Given a nonlinear system of the form (1), the goal is to find, if possible, a regular static measurement feedback of the form

$$u(t) = \alpha^{-1}(z(t), v(t)),$$

such that controlled output $y(t)$ of the closed loop system satisfies the following conditions:

(i) $\mathrm{d}y(t+k) \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}v(t), \ldots, \mathrm{d}v(t+k-r)\}, \forall k \geq r$
(ii) $\mathrm{d}y(t+r) \notin \mathcal{X}$.

Condition (i) represents the independence of the output of the closed-loop system from the disturbance whereas the condition (ii) represents the output controllability of the closed-loop system.

Analogously to the continuous-time case, see Pothin et al. (2002), define the subspace $\Omega \subset \mathcal{X}$ by

$$\Omega := \{\omega(t) \in \mathcal{X} | \forall k \in \mathbb{N} : \omega(t+k) \\ \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}y(t+r), \ldots, \mathrm{d}y(t+r+k-1)\}\}.$$

Lemma 1. The subspace Ω may be computed as the limit of the following algorithm:

$$\begin{aligned} \Omega^0 &= \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t)\} \\ \Omega^{k+1} &= \{\omega(t) \in \Omega^k | \omega(t+1) \in \Omega^k \\ &\quad + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r)\}\}, \qquad k \geq 0. \end{aligned} \tag{2}$$

Proof: We show below, that sequence Ω^k converges and in the limit we get Ω. Consider a subspace Ω^k. By (2), $\Omega^{k+1} \subset \Omega^k$ or $\Omega^{k+1} = \Omega^k$. Since the subspace Ω^k is finite-dimensional vector space, at certain step k^*+1, $\Omega^{k^*} = \Omega^{k^*+1}$. Thus the sequence (2) converges and the limit is Ω^{k^*}. We show now that $\Omega = \Omega^{k^*}$. Suppose $\omega(t) \in \Omega^{k^*}$. Then, by (2)

$$\omega(t+1) \in \Omega^{k^*-1} + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r)\}$$

and so $\omega(t+1) = \tilde{\omega}(t) + \xi \mathrm{d}y(t+r)$ for some $\tilde{\omega}(t) \in \Omega^{k^*-1}$ and function $\xi \in \mathcal{K}^*$. Since $\tilde{\omega}(t) \in \Omega^{k^*-1}$, by (2)

$$\tilde{\omega}(t+1) \in \Omega^{k^*-2} + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r)\}$$

and so forward shift of $\omega(t+1)$ is

$$\omega(t+2) \in \Omega^{k^*-2} + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r), \mathrm{d}y(t+r+1)\}.$$

Continuing in the same way, we get

$$\omega(t+k^*) \in \Omega^0 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r), \ldots, \mathrm{d}y(t+r+k^*-1)\},$$

which means that $\omega(t) \in \Omega$. We showed that if $\omega(t) \in \Omega^{k^*}$, then $\omega(t) \in \Omega$, i.e. $\Omega^{k^*} \subset \Omega$.

Now suppose that $\omega(t) \in \Omega$. Then by definition of Ω,

$$\omega(t+k^*) \in \mathcal{X} + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r), \ldots, \mathrm{d}y(t+r+k^*-1)\}.$$

Because $\Omega^0 = \mathcal{X}$,

$$\omega(t+k^*) = \tilde{\omega}(t) + \xi_1 \mathrm{d}y(t+r) + \ldots + \xi_{k^*} \mathrm{d}y(t+r+k^*-1),$$

where $\tilde{\omega}(t) \in \Omega^0$ and $\xi_1, \ldots, \xi_{k^*} \in \mathcal{K}^*$. Backward shift $\tilde{\omega}(t-1) \in \Omega^1$, because $\tilde{\omega}(t-1) \in \Omega^0$ and $\tilde{\omega}(t) \in \Omega^0 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r)\}$. Note that $\mathrm{d}y(t+r-1) \in \Omega^{k^*}$, because $\mathrm{d}y(t+r) \in \Omega^l + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r)\}$ for every $l \geq 0$. Thus backward shift of $\omega(t+k^*)$ is

$$\omega(t+k^*-1) \in \Omega^1 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r), \ldots, \mathrm{d}y(t+k^*-2)\}.$$

Continuing in the same way, we get

$$\omega(t+1) \in \Omega^{k^*-1} + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+r)\}.$$

Thus $\omega(t) \in \Omega^{k^*}$ and we are shown that $\Omega \subset \Omega^{k^*}$. Above we showed that $\Omega^{k^*} \subset \Omega$, so $\Omega = \Omega^{k^*}$. □

We will show next how Ω changes under the regular static measurement feedback $u(t) = \alpha(z(t), v(t))$. Denote by $\overline{\mathcal{K}^*}$ the field of meromorphic functions in variables $x(t)$, $v(t)$, $w(t)$ and a finite number of their independent forward- and backward shifts and define the vector spaces $\overline{\mathcal{X}} = \mathrm{span}_{\overline{\mathcal{K}^*}}\{\mathrm{d}x(t)\}$, $\overline{\mathcal{U}} = \mathrm{span}_{\overline{\mathcal{K}^*}}\{\mathrm{d}u(t+k), k \geq 0\}$, $\overline{\mathcal{W}} = \mathrm{span}_{\overline{\mathcal{K}^*}}\{\mathrm{d}w(t+k), k \geq 0\}$, $\overline{\mathcal{E}} = \overline{\mathcal{X}} + \overline{\mathcal{U}} + \overline{\mathcal{W}}$. Analogously to Xia and Moog (1999) one can prove that there exists an isomorphism $\Phi : \mathcal{E} \to \overline{\mathcal{E}}$ such that if Ω_{cl} is the subspace for the closed loop system, then $\Omega_{cl} = \Phi(\Omega)$.

Let $\omega(t) \in \Theta$ be a one-form. In general, $\omega(t)$ is a linear combination of all n basis elements of Θ, i.e. $\{\theta_1, \ldots, \theta_n\}$. However, it is often possible to find a linearly independent subset of the set $\{\theta_1, \ldots, \theta_n\}$ with less than n elements in terms of which $\omega(t)$ can be expressed.

Definition 2. (Choquet-Bruhat et al. (1996)) Let γ be the minimal number of linearly independent one-forms necessary to express a one-form $\omega(t)$. Then $\omega(t)$ is said to be of rank γ.

Note that $1 \leq \gamma \leq n$. For example, if the rank γ of a one-form $\omega(t)$ is 1, then $\omega(t) = \xi \mathrm{d}\alpha$ and thus $\omega(t) \wedge \mathrm{d}\omega(t) = 0$. In the general case, if the rank γ is k, then $\omega(t) \wedge (\mathrm{d}\omega(t))^{(k)} = 0$, where $(\mathrm{d}\omega(t))^{(k)} = \mathrm{d}\omega(t) \wedge \ldots \wedge \mathrm{d}\omega(t)$ is k-fold wedge product.

We prove the following lemma for MISO systems, providing the alternative formulation of the disturbance decoupling.

Lemma 2. Under the assumption that the relative degree r of the output $y(t)$ is finite, the system (1) is disturbance decoupled iff

$$\mathrm{d}y(t+r) \in \Omega + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}u(t)\}. \tag{3}$$

Proof: *Necessity.* Assume, that system (1) is disturbance decoupled, i.e.

$$\mathrm{d}y(t+k) \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}u(t), \ldots, \mathrm{d}u(t+k-r)\} \tag{4}$$

for $k \geq r$ and

$$\mathrm{d}y(t+r) \notin \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t)\}. \tag{5}$$

In particular, $\mathrm{d}y(t+r) \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}u(t)\}$. Rewrite the latter as

$$\mathrm{d}y(t+r) \in \mathcal{X} + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}u(t)\}. \tag{6}$$

Thus there exists a one-form $\omega_0(t) \in \mathcal{X}$ and a function $\xi \in \mathcal{K}^*$ such that $\mathrm{d}y(t+r) = \omega_0(t) + \xi \mathrm{d}u(t)$. We are going to show, that $\omega_0(t) \in \Omega$. Assume contrarily, that $\omega_0(t) \notin \Omega$. The forward shift of $\mathrm{d}y(t+r) \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}u(t)\}$ is

$$\mathrm{d}y(t+r+1) \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}w(t), \mathrm{d}u(t), \mathrm{d}u(t+1)\},$$

which yields a contradiction with (4). Thus, $\omega_0 \in \Omega$ and we can rewrite (6) as $\mathrm{d}y(t+r) \in \Omega + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}u(t)\}$.

Sufficiency. Assume that for system (1) the condition (3) is fulfilled. We must show that system (1) satisfies conditions (4) and (5). Because r is the relative degree of $y(t)$, (5) is satisfied. Because of (3),

$$\mathrm{d}y(t+r) = \omega_0(t) + \xi \mathrm{d}u(t),$$

where $\omega_0(t) \in \Omega$ and $\xi \in \mathcal{K}^*$. Since $\omega_0(t) \in \Omega$

$$\omega_0(t+l) \in \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}y(t+r), \ldots, \mathrm{d}y(t+r+l-1)\}$$

for all $l \geq 0$. Thus

$$\begin{aligned}\mathrm{d}y(t+r+l) \in \operatorname{span}_{\mathcal{K}^*}\{&\mathrm{d}x(t), \mathrm{d}y(t+r), \ldots \\ &\ldots, \mathrm{d}y(t+r+l-1), \mathrm{d}u(t+l)\}\end{aligned}$$

for all $l \geq 0$. Hence
$\mathrm{d}y(t+r+l-1) \in \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}y(t+r), \ldots, \mathrm{d}y(t+r+l-2)\}$
and

$$\begin{aligned}\mathrm{d}y(t+r+l) \in \operatorname{span}_{\mathcal{K}^*}\{&\mathrm{d}x(t), \mathrm{d}y(t+r), \ldots \\ &\ldots, \mathrm{d}y(t+r+l-2), \mathrm{d}u(t+l-1), \mathrm{d}u(t+l)\}.\end{aligned}$$

Continuing the same way, we get

$$\mathrm{d}y(t+r+l) \in \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}x(t), \mathrm{d}u(t), \ldots, \mathrm{d}u(t+l)\}.$$

Changing l by $l = k - r$, we get (4) and thus sufficiency is fulfilled. □

We are going to use the subspace Ω and the concept of the rank of a one-form to give a sufficient condition for the disturbance decoupling problem.

3. MAIN RESULTS

Theorem 3. The disturbance decoupling problem for system (1) is solvable by static measurement feedback if:

(i) $\mathrm{d}y(t+r) \in \Omega + \mathcal{Z} + \mathcal{U}$,
(ii) there exists a one-form $\omega(t) \in \mathcal{Z} + \mathcal{U}$ such that $\mathrm{d}y(t+r) - \omega(t) \in \Omega$ and rank $\omega(t) = \gamma \leq m$,
(iii) for any basis $\{\mathrm{d}\alpha_1(z(t), u(t)), \ldots, \mathrm{d}\alpha_\gamma(z(t), u(t))\}$ of $\omega(t)$,

$$\operatorname{rank}\Big[\frac{\partial \alpha(z(t), u(t))}{\partial u(t)}\Big] = \gamma, \qquad (7)$$

where $\alpha := [\alpha_1, \ldots, \alpha_\gamma]^T$.

Proof: Assume that condition (i) is fulfilled. Under the condition (ii) there exists a one-form $\omega(t)$ such that $\mathrm{d}y(t+r) - \omega(t) \in \Omega$ where

$$\omega(t) = \beta_1 \mathrm{d}\alpha_1(z(t), u(t)) + \ldots + \beta_\gamma \mathrm{d}\alpha_\gamma(z(t), u(t)).$$

When condition (iii) is satisfied, then γ one-forms $\mathrm{d}\alpha_i(z(t), u(t))$, $i = 1, \ldots, \gamma$, are independent with respect to the variable $u(t)$. Define for $i = 1, \ldots, \gamma$

$$v_i(t) = \alpha_i(z(t), u(t)). \qquad (8)$$

If $\gamma < m$, then by renumbering the inputs $u(t)$, if necessary, complete (8) with

$$v_i(t) = u_i(t), \qquad i = \gamma + 1, \ldots, m \qquad (9)$$

to get an invertible map. Define a static measurement feedback $u(t) = \alpha(z(t), u(t))$ as the solution of (8) and (9). Note that this yields

$$\mathrm{d}y(t+r) \in \Omega \oplus \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}v(t)\}$$

and thus by Lemma 2, system (1) is disturbance decoupled. □

In case of SISO systems when $m = 1$, (7) and (ii) of Theorem 3 yield

$$\operatorname{rank}\Big[\frac{\partial \alpha(z(t), u(t))}{\partial u(t)}\Big] = \gamma = 1.$$

Thus, condition (iii) of Theorem 3 is satisfied if and only if $\gamma = 1$. For SISO systems one can conclude from Theorem 3 a necessary and sufficient condition.

Corollary 4. For SISO nonlinear control systems the DDP is solvable by a regular static measurement feedback iff:

(i) $\mathrm{d}y(t+r) \in \Omega + \mathcal{Z} + \mathcal{U}$
(ii) There exists a one-form $\omega(t) \in \mathcal{Z} + \mathcal{U}$ such that $\mathrm{d}y(t+r) - \omega(t) \in \Omega$ and rank $\omega(t) = 1$.

Proof: *Necessity.* Assume that system (1) is decoupled by the regular static measurement feedback

$$u(t) = \alpha(z(t), v(t)), \qquad v(t) = \alpha^{-1}(z(t), u(t)). \qquad (10)$$

Then by Lemma 2

$$\mathrm{d}y(t+r) \in \Omega + \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}v(t)\}. \qquad (11)$$

Combining (11) with (10) implies condition (i). Since $\omega(t) = \xi \mathrm{d}(F^{-1}(z(t), u(t)))$, $\omega(t) \wedge \mathrm{d}\omega(t) = 0$ and rank $\omega(t) = 1$. Thus condition (ii) is also fulfilled.

Sufficiency. Assume that (i) holds. Then

$$\mathrm{d}y(t+r) \in \Omega \oplus \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}z(t), \mathrm{d}u(t)\}.$$

Since by (ii) the rank of the one-form $\omega(t)$ is 1, define $\omega(t) := \lambda \mathrm{d}v(t)$ and so

$$\mathrm{d}y(t+r) \in \Omega \oplus \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}v(t)\}$$

meaning that the system is decoupled. □

In general there is no necessary and sufficient condition for MISO systems, but under an additional assumptions $\Omega \cap \mathcal{Z} = \emptyset$ and $\mathrm{d}y(t+r) \in \Omega \oplus \mathcal{Z} + \mathcal{U}$ one can find a necessary and sufficient condition for MISO systems.

Theorem 5. Assume that $\Omega \cap \mathcal{Z} = \emptyset$ and $\mathrm{d}y(t+r) \in \Omega \oplus \mathcal{Z} + \mathcal{U}$. The DDP is solvable by regular static measurement feedback iff

(i) There exists a one-form $\omega(t) \in \mathcal{Z} + \mathcal{U}$ such that $\mathrm{d}y(t+r) - \omega(t) \in \Omega$ and $\gamma := \operatorname{rank} \omega(t) \leq m$.
(ii) For any basis $\{\mathrm{d}\alpha_1(z(t), u(t)), \ldots, \mathrm{d}\alpha_\gamma(z(t), u(t))\}$ of $\omega(t)$,

$$\operatorname{rank}\Big[\frac{\partial \alpha(z(t), u(t))}{\partial u(t)}\Big] = \gamma.$$

Proof: *Necessity.* Assume that system (1) is disturbance decoupled by the regular static measurement feedback $v(t) = \alpha(z(t), u(t))$. By Lemma 2, $\mathrm{d}y_{cl}(t+r) \in \Omega_{cl} + \mathcal{V}$, where $\mathcal{V} = \operatorname{span}_{\overline{\mathcal{K}^*}}\{\mathrm{d}v_1(t), \ldots, \mathrm{d}v_m(t)\}$ and $y_{cl}(t)$ is the output of the closed-loop system. Because of isomorphism $\Phi : \mathcal{E} \to \overline{\mathcal{E}}$ described above and feedback $\alpha(z(t), u(t))$, one can write

$$\mathrm{d}y(t+r) \in \Omega + \operatorname{span}_{\mathcal{K}^*}\{\mathrm{d}\alpha(z(t), u(t))\}.$$

Thus, there exist a one-form $\tilde{\omega}(t) \in \Omega$ and $\xi \in \mathcal{K}^*$ such that

$$\mathrm{d}y(t+r) = \tilde{\omega}(t) + \xi \mathrm{d}\alpha(z(t), u(t)).$$

Assumption $\mathrm{d}y(t+r) \in \Omega \oplus \mathcal{Z} + \mathcal{U}$ implies that $\tilde{\omega}(t) \in \Omega + \mathcal{Z}$. Rewrite $\tilde{\omega}(t) = \tilde{\omega}_0(t) + \tilde{\omega}_z(t)$ for some $\tilde{\omega}_0(t) \in \Omega$ and $\tilde{\omega}_z(t) \in \mathcal{Z}$. As in the proof of Lemma 2, one can show that $\tilde{\omega}_z(t) \in \Omega$. Due to the assumption $\Omega \cap \mathcal{Z} = 0$, we have $\tilde{\omega}_z(t) = 0$. Then define $\omega(t) = \xi \mathrm{d}\alpha(z(t), u(t))$ and the necessity of condition (i) is fulfilled.

Because the rank of a one-form $\omega(t)$ is γ,

$$\omega(t) = \beta_1 \mathrm{d}\alpha_1(z(t), u(t)) + \ldots + \beta_\gamma \mathrm{d}\alpha_\gamma(z(t), u(t))$$

where $\beta_i \in \mathcal{K}^*$, $i = 1, \ldots, \gamma$. Suppose, contrarily to the claim of Theorem 5 that (ii) is not fulfilled. Then there exist a one-form

$$\xi_1 \mathrm{d}\alpha_1(z(t), u(t)) + \ldots + \xi_\gamma \mathrm{d}\alpha_\gamma(z(t), u(t)) \in \mathcal{Z}.$$

Assume without loss of generality that $\xi_1 \neq 0$, then $\omega(t)$ can be decomposed into

$$\omega(t) = \tilde{\omega}_z(t) + \eta_2 \mathrm{d}\alpha_2(z(t), u(t)) + \ldots + \eta_\gamma \mathrm{d}\alpha_\gamma(z(t), u(t))$$

in which

$$\tilde{\omega}_z(t) = \frac{\beta_1}{\xi_1}(\xi_1 \mathrm{d}\alpha_1(z(t), u(t)) + \ldots + \xi_\gamma \mathrm{d}\alpha_\gamma(z(t), u(t))) \in \mathcal{Z}$$

and

$$\eta_i = \beta_i - \frac{\beta_1}{\xi_1}\xi_i,$$

for $i = 2, \ldots, \gamma$. As shown before, if $\tilde{\omega}_z(t) \in \mathcal{Z}$ then necessarily $\tilde{\omega}_z(t) \in \Omega$ and since $\Omega \cap \mathcal{Z} = 0$, this yields a contradiction. Thus condition (ii) has to be fulfilled.

Sufficiency. Because all of the conditions of Theorem 3 are satisfied, then sufficiency is fulfilled. □

4. EXAMPLES

The first example illustrates Theorem 3.

Example 1. Consider the system

$$\begin{aligned} x_1(t+1) &= x_2(t) + x_3(t)u_1(t)x_4(t) + u_2(t)x_4(t) \\ x_2(t+1) &= x_2(t) + x_3(t)u_1(t)x_4(t) + u_2(t)x_4(t) + x_3^2(t) \\ x_3(t+1) &= \cos x_1(t) \\ x_4(t+1) &= w(t) \\ y(t) &= x_1(t) \\ z(t) &= x_4(t). \end{aligned} \tag{12}$$

Note that the relative degree of the output $y(t)$ is 1, because

$$\begin{aligned} \mathrm{d}y(t+1) &= \mathrm{d}x_2(t) + u_1(t)x_3(t)\mathrm{d}x_4(t) + x_3(t)x_4(t)\mathrm{d}u_1(t) \\ &\quad + u_1(t)x_4(t)\mathrm{d}x_3(t) + u_2(t)\mathrm{d}x_4(t) + x_4(t)\mathrm{d}u_2(t). \end{aligned}$$

Next we find the vector space Ω using the algorithm, defined by (2). First,

$$\Omega^0 = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x_1(t), \mathrm{d}x_2(t), \mathrm{d}x_3(t), \mathrm{d}x_4(t)\}.$$

Because $\mathrm{d}y(t+1) = \mathrm{d}x_1(t+1)$,

$$\begin{aligned} \mathrm{d}x_1(t+1) &= \mathrm{d}y(t+1) \in \Omega^0 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+1)\} \\ \mathrm{d}x_2(t+1) &= \mathrm{d}y(t+1) + 2x_3(t)\mathrm{d}x_3(t) \\ &\in \Omega^0 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+1)\} \\ \mathrm{d}x_3(t+1) &= -\sin x_1(t)\mathrm{d}x_1(t) \in \Omega^0 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+1)\} \\ \mathrm{d}x_4(t+1) &= \mathrm{d}w(t) \notin \Omega^0 + \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}y(t+1)\}. \end{aligned}$$

Thus, $\Omega^1 = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x_1(t), \mathrm{d}x_2(t), \mathrm{d}x_3(t)\}$. In the next step we get $\Omega^1 = \Omega^2 = \Omega$. Since $\mathrm{d}z(t) = \mathrm{d}x_4(t)$, the condition (i) of Theorem 3 is satisfied, i.e. $\mathrm{d}y(t+1) \in \Omega + \mathcal{Z} + \mathcal{U}$. Next step is to choose $\omega(t)$ such that $\omega(t) \in \mathcal{Z} + \mathcal{U}$ and $\mathrm{d}y(t+1) - \omega(t) \in \Omega$. One can take $\omega(t) := u_2(t)\mathrm{d}x_4(t) + x_4(t)\mathrm{d}u_2(t) + u_1(t)x_3(t)\mathrm{d}x_4(t) + x_3(t)x_4(t)\mathrm{d}u_1(t)$ which can be rewritten as

$$\omega(t) = \mathrm{d}(u_2(t)z(t)) + x_3(t)\mathrm{d}(u_1(t)z(t)).$$

From above, the rank of $\omega(t)$ is $2 = m$. Thus condition (ii) of Theorem 3 is satisfied. Condition (iii) is easily verified and the disturbance decoupling feedback may be found as the solution of the system of equations $v_1(t) = u_2(t)z(t)$ and $v_2(t) = u_1(t)z(t)$ with respect to $u_1(t)$ and $u_2(t)$.

In the second example the rank of a one-form $\omega(t)$ is strictly less than the number of inputs, $\gamma < m$.

Example 2. Consider the system

$$\begin{aligned} x_1(t+1) &= x_2(t) + x_4(t)u_1(t)u_2(t) \\ x_2(t+1) &= x_2(t) + x_4(t)u_1(t)u_2(t) + x_3^2(t) \\ x_3(t+1) &= \cos x_1(t) \\ x_4(t+1) &= w(t) \\ y(t) &= x_1(t) \\ z(t) &= x_4(t). \end{aligned} \tag{13}$$

The relative degree of output $y(t)$ is 1 and

$$\begin{aligned} \mathrm{d}y(t+1) &= \mathrm{d}x_2(t) + u_1(t)u_2(t)\mathrm{d}x_4(t) \\ &\quad + u_2(t)x_4(t)\mathrm{d}u_1(t) + u_1(t)x_4(t)\mathrm{d}u_2(t). \end{aligned}$$

Like in Example 1, one can find the subspace $\Omega = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x_1(t), \mathrm{d}x_2(t), \mathrm{d}x_3(t)\}$ and thus the condition (i) of Theorem 3 is satisfied. Since now one can choose $\omega(t)$ as $\omega(t) = \mathrm{d}(u_1(t)u_2(t)z(t))$, $\gamma :=$rank $\omega(t) = 1$ and condition (ii) of Theorem 3 is fulfilled. Note that (iii) is satisfied and the regular static measurement feedback can be found from $v_1(t) = u_1(t)u_2(t)z(t)$ and $v_2(t) = u_2(t)$.

The following example shows that for the MISO case the condition (i) of Theorem 3 is not necessary.

Example 3. Consider the system

$$\begin{aligned} x_1(t+1) &= x_2(t) + u_1(t)x_3(t)x_4(t) + u_2(t)x_4(t) \\ x_2(t+1) &= x_2(t) + u_1(t)x_3(t)x_4(t) + u_2(t)x_4(t) + x_3^2(t) \\ x_3(t+1) &= u_1(t)x_4(t) \\ x_4(t+1) &= w(t) \\ y(t) &= x_1(t) \\ z(t) &= x_4(t) \end{aligned} \tag{14}$$

Condition (i) of Theorem 3 is not satisfied, because $\Omega = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x_1(t)\}$, but

$$\begin{aligned} \mathrm{d}y(t+1) &= \mathrm{d}x_2(t) + u_1(t)x_4(t)\mathrm{d}x_3(t) + u_1(t)x_3(t)\mathrm{d}x_4(t) \\ &\quad + x_3(t)x_4(t)\mathrm{d}u_1(t) + u_2(t)\mathrm{d}x_4(t) + x_4(t)\mathrm{d}u_2(t). \end{aligned}$$

Still, one can choose $\omega(t) = x_3(t)\mathrm{d}(u_1(t)z(t)) + \mathrm{d}(u_2(t)z(t))$ and find the static measurement feedback from $v_1(t) = u_1(t)z(t)$ and $v_2(t) = u_2(t)z(t)$, which solves the DDP.

Example 4. Consider the system

$$\begin{aligned} x_1(t+1) &= e^{x_2(t)x_3^2(t)} \\ x_2(t+1) &= \cos x_2(t) \\ x_3(t+1) &= u_1(t)\sin x_4(t) \\ x_4(t+1) &= u_2(t)w(t) \\ y(t) &= x_1(t) \\ z(t) &= x_4(t). \end{aligned} \tag{15}$$

Note that the relative degree of the output $y(t)$ is 2. Sequence (2) for this system converges and the subspace $\Omega = \Omega^2 = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x_2(t)\}$. Because

$$\mathrm{d}y(t+2) = \mathrm{d}(e^{u_1^2(t)\cos x_2(t)\sin^2 x_4(t)}),$$

the first condition of Theorem 3 is satisfied. One can choose $\omega(t)$ to be

$$
\begin{aligned}
\omega(t) = 2e^{u_1^2(t)\cos x_2(t)\sin^2 z(t)}&(\cos x_2(t)\sin^2 z(t)u_1(t)\mathrm{d}u_1(t)\\
&+\cos x_2(t)\cos z(t)\sin z(t)u_1^2(t)\mathrm{d}z(t)).
\end{aligned}
$$

Because $\omega(t)\wedge \mathrm{d}\omega(t) = 0$, the rank of $\omega(t)$ is 1 and thus the second condition is satisfied. Next, one can find $\alpha(z(t), u(t)) = \ln(u_1(t)\sin z(t))$; so rank$\left[\frac{\partial\alpha(z(t),u(t))}{\partial u(t)}\right] = 1$ and condition (iii) is also satisfied. The feedback that solves the DDP is

$$
\begin{aligned}
u_1(t) &= e^{v_1(t)}\csc z(t)\\
u_2(t) &= v_2(t).
\end{aligned}
$$

and the closed-loop system

$$
\begin{aligned}
x_1(t+1) &= e^{x_2(t)x_3(t)^2}\\
x_2(t+1) &= \cos(x_2(t))\\
x_3(t+1) &= e^{v_1(t)}\\
x_4(t+1) &= v_2(t)w(t)\\
y(t) &= x_1(t)\\
z(t) &= x_4(t)
\end{aligned}
$$

is disturbance decoupled.

Example 5. Consider the system

$$
\begin{aligned}
x_1(t+1) &= x_4(t)w(t)\ln(x_2(t)u_2(t))\\
x_2(t+1) &= x_1(t)x_2(t)\\
x_3(t+1) &= e^{u_1(t)x_4(t)}\\
x_4(t+1) &= u_2(t)w(t)\\
y(t) &= x_1(t)\\
z(t) &= x_4(t).
\end{aligned}
\tag{16}
$$

The relative degree of output $y(t)$ is 1 and the subspace $\Omega = \mathrm{span}_{\mathcal{K}^*}\{\mathrm{d}x_1(t), \mathrm{d}x_2(t)\}$. Since

$$
\begin{aligned}
\mathrm{d}y(t+1) = \frac{w(t)z(t)}{x_2(t)}\mathrm{d}x_2(t) &+ \frac{w(t)z(t)}{u_2(t)}\mathrm{d}u_2(t)\\
&+ w(t)\ln(u_2(t)x_2(t))\mathrm{d}z(t)\\
&+ z(t)\ln(u_2(t)x_2(t))\mathrm{d}w(t),
\end{aligned}
$$

there does not exist a one-form $\omega(t) \in \mathcal{Z} + \mathcal{U}$ such that $\mathrm{d}y(t+1) - \omega(t) \in \Omega$. Thus the second condition of Theorem 3 is not satisfied and the DDP is not solvable by the static measurement feedback.

5. CONCLUSION

In this paper the notion of the rank of a one-form and the subspace Ω of differential one-forms was used to solve the DDP for nonlinear discrete-time control systems by static measurement feedback. Sufficient conditions for solvability of the DDP were found. Necessary and sufficient conditions were derived from the above conditions for SISO systems and for MISO systems under the additional assumption. The sufficient condition also provided a procedure to find the static measurement feedback to solve the DDP. Because these conditions are very restrictive, further research is necessary. Next step is to extend the results by Xia and Moog (1999) addressing the dynamic measurement feedback in the framework of differential forms for discrete-time systems. Those results can then be compared with those by Kotta et al. (2011), that are obtained using the tools of algebra of functions. Additionally to above theoretical problems the functions in Mathematica will be developed for solving the DDP and integrated into the symbolic software package NLControl, developed in the Institute of Cybernetics at Tallinn University of Technology.

ACKNOWLEDGMENTS

The work has been supported by the ESF grant nr. 8787. A. Kaldmäe is a master student of Tallinn University and this work is part of his thesis.

REFERENCES

Andiarti, R. and Moog, C. (1996). Output feedback disturbance decoupling in nonlinear systems. *IEEE Trans. Autom. Control*, 41, 1683–1689.

Aranda-Bricaire, E. and Kotta, Ü. (2001). Generalized controlled invariance for discrete-time nonlinear systems with application to the dynamic disturbance problem. *IEEE Trans. Autom. Control*, 46, 165–171.

Aranda-Bricaire, E. and Kotta, Ü. (2004). A geometric solution to the dynamic disturbance decoupling for discrete-time nonlinear systems. *Kybernetika*, 49, 197–206.

Aranda-Bricaire, E., Kotta, U., and Moog, C.H. (1996). Linearization of discrete-time systems. *SIAM J. Control and Optimization*, (6), 1999–2023.

Choquet-Bruhat, Y., DeWitt-Morette, C., and Dillard-Bleick, M. (1996). *Analysis, Manifolds and Physics.* Elsevier.

Conte, G., Moog, C., and Perdon, A. (2007). *Algebraic Methods for Nonlinear Control Systems. Theory and Applications.* Springer.

Fliegner, T. and Nijmeijer, H. (1994). Dynamic disturbance decoupling of nonlinear discrete-time systems. In *Proc. of the 33rd IEEE Conf. on Decision and Control*, volume 2, 1790–1791.

Grizzle, J. (1985). Controlled invariance for discrete-time nonlinear systems with an application to the disturbance decoupling problem. *IEEE Trans. Autom. Control*, 30, 868–873.

Isidori, A. (1995). *Nonlinear control systems.* Springer, London.

Isidori, A., Krener, A., Gori-Giorgi, C., and Monaco, S. (1981). Nonlinear decoupling via feedback: A differential gemetric approach. *IEEE Trans. Autom. Control*, 26, 331–345.

Kotta, U. and Mullari, T. (2010). Discussion on: "Unified approach to the problem of full decoupling via output feedback". *European Journal of Control*, 16(4), 326–328.

Kotta, Ü. and Nijmeijer, H. (1991). Dynamic disturbance decoupling for nonlinear discrete-time systems (in russian). *Proc. Acad of Sciences of USSR, Technical Cybernetics*, 52–59.

Kotta, U., Shumsky, A., and Zhirabok, A. (2011). Output feedback disturbance decoupling in discrete-time nonlinear systems. *Submitted for publication.*

Monaco, S. and Normand-Cyrot, D. (1984). Invariant distributions for discrete-time nonlinear systems. *Systems and Control Letters*, 5, 191–196.

Nijmeijer, H. and van der Schaft, A. (1990). *Nonlinear dynamical control systems*. Springer, New York.

Pothin, R., Moog, C., and Xia, X. (2002). Disturbance decoupling of nonlinear miso systems by static measurement feedback. *Kybernetika*, 38, 601–608.

Shumsky, A. and Zhirabok, A. (2010). Unified approach to the problem of full decoupling via output feedback. *European Journal of Control*, 16(4), 313–325.

Xia, X. and Moog, C. (1999). Disturbance decoupling by measurement feedback for siso nonlinear systems. *IEEE Trans. Autom. Control*, 44, 1425–1429.

Zhirabok, A. and Shumsky, A. (2008). *The algebraic methods for analysis of nonlinear dynamic systems (In Russian)*. Dalnauka, Vladivostok.

Transfer matrix and its Jacobson form for nonlinear systems on time scales: *Mathematica* implementation

J. Belikov [*] **Ü. Kotta** [*] **A. Leibak** [**]

[*] *Institute of Cybernetics, Tallinn University of Technology, Akadeemia tee 21, 12618, Tallinn, Estonia*
fax : +372 620 4151 and e-mails : {jbelikov, kotta}@cc.ioc.ee
[**] *Department of Mathematics, Tallinn University of Technology, Ehitajate tee 5, 19086, Tallinn, Estonia*
fax : +372 620 3051 and e-mail : alar@staff.ttu.ee

Abstract: This paper suggests a detailed algorithm for computation of the Jacobson form of the polynomial matrix associated with the transfer matrix describing the multi-input multi-output nonlinear control system, defined on homogeneous time scale. The algorithm relies on the theory of skew polynomial rings.

Keywords: nonlinear control system, input-output models, time scales, symbolic computations.

1. INTRODUCTION

In the Institute of Cybernetics at Tallinn University of Technology symbolic software package NLControl has been developed over the years within *Mathematica* environment, for the detailed information see Kotta and Tõnso (2003), Tõnso et al. (2009). The package is based on different algebraic methods, in particular on the approach based on the differential one-forms, see Conte et al. (2007), and on the theory of skew polynomial ring. It allows to solve various modelling, analysis and synthesis problems not only for continuous and discrete-time nonlinear control systems, but also for those defined on homogeneous time scales, see Casagrande et al. (2010). Note that the key idea of a time scale calculus is unification of the theories of differential and difference equations, see Bohner and Peterson (2001). Both continuous and discrete-time (in terms of the difference operator) cases are merged in time scale formalism into a general framework which provides not only unification but also an extension. The main concept of the time scale calculus is the so-called delta-derivative that is a generalization of both time-derivative and the difference operator (but accommodates more possibilities, e.g. q-difference operator).

In the linear control theory the transfer matrix (TM) approach has been very popular. Recently the concept of the TM has been extended for the continuous-time nonlinear control systems Halás (2008) and later in Halás and Kotta (2007a) for discrete-time systems and for control system defined in terms of the pseudo-linear operator, see Halás and Kotta (2007b). Note that the latter includes also the systems, defined on homogeneous time scales, since in that case the delta-derivative may be understood as the special case of the pseudo-linear operator. However, the pseudo-linear approach cannot handle the systems defined on non-homogeneous time scales, since the time scale formalism unifies both continuous- and discrete-time cases, it would be interesting to study whether TM-based transparent control methods can be extended to nonlinear systems defined on time scale. In TM-based control design, a special form of the matrix, the Jacobson-Teichmüller [1] form, plays a key role. The first step in transformation of the TM into the required form is to transform the polynomial matrix, associated with it, into the Jacobson form.

Note that in the case of nonlinear control systems, the polynomials belong into the non-commutative polynomial ring that is the principal ideal domain (p.i.d.). The basic algorithm to transform a polynomial matrix into this form was given in Cohn (1985) for an arbitrary ring being the p.i.d. There exist a number of implementations of this algorithm such as Blinkov et al. (2003), Chyzak et al. (2007) and its fraction-free version Levandovskyy and Schindelar (2010). However, except Insua and Ladra (2006), not available for public use, all of them have been implemented either in Maple, e.g. Blinkov et al. (2003), Chyzak et al. (2007) or in Singular:Plural Levandovskyy and Schindelar (2010). Moreover, it is not documented whether and how these packages are applicable for nonlinear control systems, in particular for those defined on homogeneous time scale.

The main contribution of the paper is the specification the algorithm given in Cohn (1985) into the form necessary to handle the nonlinear control system defined on homogeneous time scale and description of the experience from its implementation in *Mathematica*, within the package NLControl. All the steps of the algorithm are clear, strictly defined and easily convertible into any programming code. It should be mentioned that some preliminary results for the discrete-time case were presented in Belikov et al. (2010).

2. CALCULUS ON TIME SCALE

For a general introduction to the calculus on time scales, see Bohner and Peterson (2001). Here we give only those notions and facts that we need in our paper.

[1] Note that in the linear control theory this form is called the Smith-McMillan form, see Ito et al. (2003).

A *time scale* $\mathbb{T}$ is an arbitrary nonempty closed subset of the set $\mathbb{R}$ of real numbers. The standard cases include $\mathbb{T} = \mathbb{R}$, $\mathbb{T} = \mathbb{Z}$, $\mathbb{T} = h\mathbb{Z}$ for $h > 0$, but also $\mathbb{T} = q^{\mathbb{Z}} := \{q^k : k \in \mathbb{Z}\} \cup \{0\}, q > 1$ is a time scale.

The following operators on $\mathbb{T}$ are often used:

- the *forward jump operator* $\sigma : \mathbb{T} \to \mathbb{T}$, defined by $\sigma(t) := \inf\{\tau \in \mathbb{T} : \tau > t\}$ and $\sigma(\sup \mathbb{T}) = \sup \mathbb{T}$, if $\sup \mathbb{T} \in \mathbb{T}$,
- the *backward jump operator* $\rho : \mathbb{T} \to \mathbb{T}$, defined by $\rho(t) := \inf\{\tau \in \mathbb{T} : \tau < t\}$ and $\rho(\inf \mathbb{T}) = \inf \mathbb{T}$, if $\inf \mathbb{T} \in \mathbb{T}$,
- the *graininess* function $\mu : \mathbb{T} \to [0, \infty)$, defined by $\mu(t) := \sigma(t) - t$.

If $\mu \equiv \mathrm{const}$ then a time scale $\mathbb{T}$ is called *homogeneous*. In this paper we assume that the time scale $\mathbb{T}$ is homogeneous.

Example 1:

- If $\mathbb{T} = \mathbb{R}$, then for any $t \in \mathbb{R}$, $\sigma(t) = t = \rho(t)$, and the graininess function $\mu(t) \equiv 0$.
- If $\mathbb{T} = h\mathbb{Z}$, for $h > 0$, then for every $t \in h\mathbb{Z}$, $\sigma(t) = t + h$, $\rho(t) = t - h$, and $\mu(t) = h$.
- If $\mathbb{T} = \overline{q^{\mathbb{Z}}}$, for $q > 1$, then for every $t \in \mathbb{T}$, $\sigma(t) = qt$, $\rho(t) = \frac{t}{q}$, and $\mu(t) = (q - 1)t$.

So, the first two cases are homogeneous time scales whereas the third is not.

Definition 1. The *delta derivative* of a function $f : \mathbb{T} \to \mathbb{R}$ at t is the number $f^{\Delta}(t)$ such that for each $\varepsilon > 0$ there exists a neighborhood $U(\varepsilon)$ of t, $U(\varepsilon) \subset \mathbb{T}$ such that for all $\tau \in U(\varepsilon)$, $|f(\sigma(t)) - f(\tau) - f^{\Delta}(t)(\sigma(t) - \tau)| \leq \varepsilon|\sigma(t) - \tau|$.

The typical special cases of the delta operator are summarized in the following remark.

Remark 1. (i) If $\mathbb{T} = \mathbb{R}$, then $f : \mathbb{R} \to \mathbb{R}$ is delta differentiable at $t \in \mathbb{R}$ if and only if $f^{\Delta}(t) = \lim_{s \to t} \frac{f(t) - f(s)}{t - s} = f'(t)$, i.e. iff f is differentiable in the ordinary sense at t.

(ii) If $\mathbb{T} = T\mathbb{Z}$, where $T > 0$, then $f : T\mathbb{Z} \to \mathbb{R}$ is always delta differentiable at every $t \in T\mathbb{Z}$ with $f^{\Delta}(t) = \frac{f(\sigma(t)) - f(t)}{\mu(t)} = \frac{f(t+T) - f(t)}{T}$ meaning the usual forward difference operator.

Proposition 1. Let $f : \mathbb{T} \to \mathbb{R}$, $g : \mathbb{T} \to \mathbb{R}$ be two delta differentiable functions defined on $\mathbb{T}$ and let $t \in \mathbb{T}$. Then the delta derivative satisfies the following properties

(i) $f^{\sigma} = f + \mu f^{\Delta}$,
(ii) $(\alpha f + \beta g)^{\Delta} = \alpha f^{\Delta} + \beta g^{\Delta}$, for any constants α and β,
(iii) $(fg)^{\Delta} = f^{\sigma} g^{\Delta} + f^{\Delta} g$,
(iv) if $gg^{\sigma} \neq 0$, then $(f/g)^{\Delta} = (f^{\Delta} g - f g^{\Delta})/(gg^{\sigma})$.

For a function $f : \mathbb{T} \to \mathbb{R}$ we define second delta derivative $f^{[2]} := f^{\Delta\Delta}$ provided that f^{Δ} is delta differentiable on $\mathbb{T}$. Similarly we define higher order derivatives $f^{[n]}$.

Denote $\sigma^n := \underbrace{\sigma \circ \cdots \circ \sigma}_{n-\text{times}}$ and $f^{\sigma^n} := f \circ \sigma^n$

Proposition 2. (Kotta et al. (2009)). Let f and f^{Δ} be delta differentiable functions on homogeneous time scale $\mathbb{T}$. Then

(i) $f^{\Delta\sigma} = f^{\sigma\Delta}$,
(ii) $f^{\sigma^n} = \sum_{k=0}^{n} C_n^k \mu^k f^{[k]}$.

At the end of this section we introduce some notation that will be useful in the following sections. Let f be a function admitting the delta-derivatives up to the c-th order. Let a and b be integers such that $0 \leq a < b \leq c$. We set $f^{[0]} = f$. Let $f^{[a \dots b]}$ denote the set $\{f^{[a]}, \dots, f^{[b]}\}$.

3. PRELIMINARIES

Consider a multi-input multi-output nonlinear control system described by a set of higher order input-output delta-differential equations on the homogeneous time scale $\mathbb{T}$ relating the inputs u_j, $j = 1 \dots, m$, the outputs y_i, $i = 1, \dots, p$ and the finite number of their delta derivatives:

$$y_i^{[n_i]} = \Phi_i\Big(y_1^{[0 \dots n_{i1}-1]}, \dots, y_p^{[0 \dots n_{ip}-1]}, u_1^{[0 \dots s_{i1}]}, \dots, u_m^{[0 \dots s_{pm}]}\Big), \quad (1)$$

where the functions Φ_i are real analytic functions of their arguments, and functions $y_i : \mathbb{T} \to \mathbb{R}$, $i = 1, \dots, p$ and $u_j : \mathbb{T} \to \mathbb{R}$, $j = 1, \dots, m$ are delta differentiable at least up to order n_i and $s_j := \max_{1 \leq i \leq p}(s_{ij})$, respectively.

3.1 Algebraic framework

Below we briefly recall the algebraic formalism for nonlinear control systems defined on homogeneous time scales, described in Bartosiewicz et al. (2007), Kotta et al. (2009), Kotta et al. (2011). Let $\mathscr{K}$ denote the field of meromorphic functions in a finite number of (independent) variables

$$\mathscr{C} = \{y_1^{[0 \dots n_1-1]}, \dots, y_p^{[0 \dots n_p-1]}, u_j^{[k]}, j = 1, \dots, m, k \geq 0\}.$$

Note that under the mild assumption on submersivity of system (1) (see below) the jump operator $\sigma : \mathscr{K} \to \mathscr{K}$ and the delta derivative $\Delta : \mathscr{K} \to \mathscr{K}$ may be extended to the field $\mathscr{K}$ as follows, see Kotta et al. (2011)

$$\sigma(F)\left(y_1^{[0 \dots n_1-1]}, \dots, y_p^{[0 \dots n_p-1]}, u_1^{[0 \dots s_1+1]}, \dots, u_m^{[0 \dots s_m+1]}\right)$$
$$:= F\left(y_1^{[0 \dots n_1-1]\sigma}, \dots, y_p^{[0 \dots n_p-1]\sigma}, u_1^{[0 \dots s_1]\sigma}, \dots, u_m^{[0 \dots s_m]\sigma}\right),$$

where

$$y_i^{[0 \dots n_i-1]\sigma} = y_i^{[0 \dots n_i-1]} + \mu \cdot \Big[y_i^{[1 \dots n_i-1]}, \Phi_i\Big(y_1^{[0 \dots n_{i1}]}, \dots, y_i^{[0 \dots n_i-1]}, \dots, y_p^{[0 \dots n_{ip}]}, u_1^{[0 \dots s_{p1}]}, \dots, u_m^{[0 \dots s_{pm}]}\Big)\Big],$$

$i = 1, \dots, p$, $u_j^{[0 \dots s_j]\sigma} = u_j^{[0 \dots s_j]} + \mu u_j^{[1 \dots s_j+1]}$, $j = 1, \dots, m$ and [2]

$$\Delta(F)\Big(y_i, \dots, y_p^{[n_i-1]}, u_j, \dots, u_j^{[k+1]}\Big) :=$$
$$\int_0^1 \{\mathrm{grad}\ F\Big(y_i + h\mu y_i^{\Delta}, \dots, y_i^{[n_i-1]} + h\mu\Phi_i\Big(y_1^{[0 \dots n_{i1}-1]}, \dots, y_p^{[0 .. n_{ip}-1]}, u_1^{[0 \dots s_{i1}]}, \dots, u_m^{[0 \dots s_{im}]}\Big),$$
$$u_j + h\mu u_j^{\Delta}, \dots, u_j^{[k]} + h\mu u_j^{[k+1]}\Big) \cdot$$
$$\begin{bmatrix} \Big(y_1^{[1 \dots n_1-1]}, \dots, y_p^{[1 \dots n_p-1]}\Big)^T, \\ \Phi_i\Big(y_1^{[0 \dots n_{i1}-1]}, \dots, y_p^{[0 \dots n_{ip}-1]}, \\ u_1^{[0 \dots s_{i1}]}, \dots, u_m^{[0 \dots s_{im}]}\Big), \\ \Big(u_1^{[1 \dots s_1+1]}, \dots, u_m^{[1 \dots s_m+1]}\Big)^T \end{bmatrix} \} \mathrm{d}h.$$

Notice that we will use $\sigma(F)$ and F^{σ} to denote the action of σ on F. Similarly, both $\Delta(F)$ and F^{Δ} will be used interchangeably.

[2] Proposition 3.3 from Bartosiewicz et al. (2007) shows how $\Delta(F)$ may be calculated not using integral explicitly.

In case σ is not injective, there may exist non-zero functions ϕ such that $\sigma(\phi) = 0$ meaning that the operator σ is not well-defined on the field $\mathscr{K}$. For σ to be an injective endomorphism on $\mathscr{K}$, the system (1) has to be submersive which can be guaranteed by the condition of the following theorem.

Theorem 1. (Kotta et al. (2011)). The nonlinear control system, defined on homogeneous time scale via the higher order i/o equations (1), is submersive if and only if the following condition

$$\operatorname{rank}_{\mathscr{K}} \begin{pmatrix} 1+\alpha_{11} & \dots & \alpha_{1p} & \beta_{11} & \dots & \beta_{1m} \\ \alpha_{p1} & \dots & 1+\alpha_{pp} & \beta_{p1} & \dots & \beta_{pm} \end{pmatrix} = p \qquad (2)$$

holds, where

$$\alpha_{ij} := \sum_{k=0}^{n_j-1} (-1)^{n_j-k-1} \mu^{n_j-k} \frac{\partial \Phi_i}{\partial y_j^{[k]}},$$

$i, j = 1, \dots, p$ and

$$\beta_{lk} := \sum_{j=0}^{s} (-1)^{s-j+1} \mu^{s-j+2} \frac{\partial \Phi_l}{\partial u_k^{[j]}},$$

$l = 1, \dots, p, k = 1, \dots, m.$

The operator Δ satisfies a generalization of Leibnitz rule

$$(FG)^{\Delta} = F^{\sigma} G^{\Delta} + F^{\Delta} G, \qquad (3)$$

for $F, G \in \mathscr{K}$. The derivation satisfying rule (3) is called a "σ-*derivation*", see Cohn (1985). Therefore, $\mathscr{K}$ is a differential field equipped with a σ-derivation Δ. In general, the field $\mathscr{K}$ is not inversive, i.e. not every element of $\mathscr{K}$ has a pre-image. Nevertheless, since Δ is injective, up to an isomorphism there exists an inversive σ-differential overfield $\mathscr{K}^*$, called the *inversive closure* of $\mathscr{K}$, such that Δ can be extended to $\mathscr{K}^*$ and this extension is automorphism of $\mathscr{K}^*$, see Cohn (1985). In Bartosiewicz et al. (2007) the details of construction of $\mathscr{K}^*$ for nonlinear control systems defined on time scales can be found. Below assume that $\mathscr{K}^*$ is given and use the same symbol $\mathscr{K}$ to denote the σ-differential field and its inversive closure.

Over the σ-differential field $\mathscr{K}$ one can define the vector space

$$\begin{aligned} \mathscr{E} := \operatorname{span}_{\mathscr{K}}\{ & \mathrm{d}y_i, \mathrm{d}y_i^{\Delta}, \dots, \mathrm{d}y_i^{[n_i-1]}, i = 1, \dots, p, \\ & \mathrm{d}u_j^{[k]}, j = 1, \dots, m, k \geq 0\}. \end{aligned}$$

The elements of $\mathscr{E}$ are called one-forms. For $F \in \mathscr{K}$ we define the operator $\mathrm{d} : \mathscr{K} \to \mathscr{E}$ as follows

$$\mathrm{d}F := \sum_{i=1}^{p} \sum_{l=0}^{n_i-1} \frac{\partial F}{\partial y_i^{[l]}} \mathrm{d}y_i^{[l]} + \sum_{j=1}^{m} \sum_{\ell=0}^{k} \frac{\partial F}{\partial u_j^{[\ell]}} \mathrm{d}u_j^{[\ell]}.$$

$\mathrm{d}F$ is said to be the (*total*) *differential* of the function F and is a one-form.

Let $\omega = \sum_j \alpha_j \mathrm{d}\varphi_j$ be a one-form with $\alpha_j \in \mathscr{K}$ and $\varphi_j \in \mathscr{C}$. Then, the operators $\sigma : \mathscr{K} \to \mathscr{K}$ and $\Delta : \mathscr{K} \to \mathscr{K}$ induce the operators $\sigma : \mathscr{E} \to \mathscr{E}$ and $\Delta : \mathscr{E} \to \mathscr{E}$ by

$$\sigma(\omega) := \sum_i \sigma(\alpha_i) \mathrm{d}\left[\sigma(\zeta_i)\right], \qquad (4)$$

$$\Delta(\omega) := \sum_i \left\{ \Delta(\alpha_i) \mathrm{d}\varphi_i + \sigma(\alpha_i) \mathrm{d}\left[\Delta(\varphi_i)\right] \right\}, \qquad (5)$$

Since $\sigma(\alpha_i) = \alpha_i + \mu\Delta(\alpha_i)$, (5) may be alternatively written as

$$\Delta(\omega) = \sum_i \left\{ \Delta(\alpha_i) \mathrm{d}\varphi_i + (\alpha_i + \mu\Delta(\alpha_i))\, \mathrm{d}\left[\Delta(\varphi_i)\right] \right\}.$$

It has been proved that $\Delta(\mathrm{d}F) = \mathrm{d}[F^{\Delta}]$, $\sigma(\mathrm{d}F) = \mathrm{d}[F^{\sigma}]$ and $\Delta\sigma = \sigma\Delta$, see Bartosiewicz et al. (2007).

3.2 *Polynomial framework*

Next, we recall the polynomial formalism which allows to represent the nonlinear i/o equations (1) via two polynomial matrices. Consider the differential field $\mathscr{K}$ with the σ-derivation Δ with σ being an automorphism of $\mathscr{K}$. A *left* differential polynomial is an element which can be uniquely written in the form $a(\partial) = \sum_{i=0}^{n} a_i \partial^{n-i}$, $a_i \in \mathscr{K}$, where ∂ is a formal variable and $a(\partial) \not\equiv 0$ if and only if at least one of the coefficients a_i, $i = 0, \dots, n$ is nonzero. If $a_0 \not\equiv 0$, then the positive integer n is called the degree of the left polynomial $a(\partial)$, denoted by $\deg a(\partial)$. Besides that we set $\deg 0 = -\infty$. The addition of the left differential polynomials is defined in the standard way. However, for $a \in \mathscr{K}$ the multiplication is defined by

$$\partial \cdot a := a^{\sigma} \partial + a^{\Delta}. \qquad (6)$$

The ring of differential polynomials will be denoted by $\mathscr{K}[\partial; \sigma, \Delta]$. Since σ is an automorphism, the ring of the left differential polynomials is a skew polynomial ring, that is proved to satisfy the left Ore condition, see Farb and Dennis (1993). By left Ore condition for all nonzero $a, b \in \mathscr{K}[\partial; \sigma, \Delta]$ there exist nonzero $a_1, b_1 \in \mathscr{K}[\partial; \sigma, \Delta]$ such that $a_1 b = b_1 a$, that is, a and b have a common left multiple (clm). The ring $\mathscr{K}[\partial; \sigma, \Delta]$ can, therefore, be embedded into its quotient field (field of fractions) by defining its left quotients as $\frac{a}{b} = b^{-1} \cdot a$, see Ore (1933). Denote the resulting quotient field by $\mathscr{K}(\partial; \sigma, \Delta)$. Moreover, we write $\mathscr{K}(\partial; \sigma, \Delta)^{p \times m}$ for the set of $p \times m$ rational matrices with entries in $\mathscr{K}(\partial; \sigma, \Delta)$, and $\mathscr{K}[\partial; \sigma, \Delta]^{p \times m}$ for the set of $p \times m$ polynomial matrices with entries in $\mathscr{K}[\partial; \sigma, \Delta]$.

Let $\sigma^n := \underbrace{\sigma \circ \cdots \circ \sigma}_{n-\text{times}}$ and denote $\sigma^n(a)$ by a^{σ^n} for $a \in \mathscr{K}$.

Lemma 1. (Kotta et al. (2009)). Let $a \in \mathscr{K}$. Then $\partial^n \cdot a \in \mathscr{K}[\partial; \sigma, \Delta]$, for $n \geq 0$, and $\partial^n \cdot a = \sum_{i=0}^{n} C_n^i \left(a^{[n-i]}\right)^{\sigma^i} \partial^i$.

In order to describe the i/o equation (1) via two polynomial matrices, we define

$$\partial^k \mathrm{d}y_{\nu} := \mathrm{d}y_{\nu}^{(k)}, \quad \partial^l \mathrm{d}u_{\upsilon} := \mathrm{d}u_{\upsilon}^{(l)} \qquad (7)$$

for $\nu = 1, \dots, p$, $\upsilon = 1, \dots, m$ and $k, l \geq 0$ in the vector space $\mathscr{E}$. Since an arbitrary one-form $\omega \in \mathscr{E}$ has the form $\omega = \sum_{\nu=1}^{p} \sum_{i=0}^{n-1} a_{\nu i} \mathrm{d}y_{\nu}^{(i)} + \sum_{\upsilon=1}^{m} \sum_{j=0}^{k} b_{\upsilon j} \mathrm{d}u_{\upsilon}^{(j)}$, where $a_{\nu i}, b_{\upsilon j} \in \mathscr{K}$, so ω can be expressed in terms of the left differential polynomials as $\omega = \sum_{\nu=1}^{p} \left(\sum_{i=0}^{n-1} a_{\nu i} \partial^i\right) \mathrm{d}y_{\nu} + \sum_{\upsilon=1}^{m} \left(\sum_{j=0}^{k} b_{\upsilon j} \partial^j\right) \mathrm{d}u_{\upsilon}$. A left differential polynomial can be considered as an operator acting on vectors $y = [y_1, \dots, y_p]^T$ and $u = [u_1, \dots, u_m]^T$ from $\mathscr{E}$: $\left(\sum_{i=0}^{k} a_i \partial^i\right)(\alpha \mathrm{d}\zeta) := \sum_{i=0}^{k} a_i (\partial^i \cdot \alpha) \mathrm{d}\zeta$, with $a_i, \alpha \in \mathscr{K}$ and $\mathrm{d}\zeta \in \{\mathrm{d}y, \mathrm{d}u\}$, where by Lemma 1, $\partial^i \cdot \alpha = \sum_{k=0}^{i} C_i^k \left(a^{[n-i]}\right)^{\sigma^i} \partial^k$. It is easy to note that $\partial(\omega) = \Delta(\omega)$, for $\omega \in \mathscr{E}$.

Now, by differentiating equation (1) and using (7) we get

$$P(\partial) \mathrm{d}y = Q(\partial) \mathrm{d}u, \qquad (8)$$

where $P(\partial) \in \mathscr{K}[\partial; \sigma, \Delta]^{p \times p}$ and $Q(\partial) \in \mathscr{K}[\partial; \sigma, \Delta]^{p \times m}$.

We assume that the Dieudonné determinant of the matrix $P(\partial)$ in (8) is nonzero, see Artin (1957) for details. The latter means that the following definition of the transfer matrix is well-defined.

Definition 2. An element of the form $H(\partial) := P^{-1}(\partial)Q(\partial) \in \mathscr{K}(\partial;\sigma,\Delta)^{p\times m}$, such that $\mathrm{d}y = H(\partial)\mathrm{d}u$, is said to be a transfer matrix of nonlinear system[3] (1).

Note that though every control system can be described by the transfer matrix, the converse is not always true. The reason is that the one-form corresponding to the transfer function may not be integrable, see Halás and Kotta (2007a) for details.

3.3 Polynomial matrices

Here we recall some basic properties of matrices with skew-polynomial entries. Suppose the matrix $P(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$.

Definition 3. An elementary column (row) operation on a polynomial matrix $P(\partial)$ is one of the following four operations

(i) interchanging two columns (rows);
(ii) multiplying any column (row) by invertible element $k \in \mathscr{K}[\partial;\sigma,\Delta]$ from the right (left);
(iii) adding a right (left) multiple of one column (row) to another;
(iv) replacement of the first elements of any two columns (rows) by their greatest common left (right) divisor (gcl(r)d) and zero, respectively.

All these operations correspond to multiplication of the matrix $P(\partial)$ by an elementary matrix $E^s_R(\partial)$ or $E^s_L(\partial)$ from the right or left, respectively Cohn (1985), where $s \in \{(i)-(iv)\}$. Operations (i)-(iii) may be represented by the product of the matrices of the form $E_{ij}(\partial) = I_\nu + 1_{ij}k$ with I_ν identity matrix and 1_{ij} the matrix made of a single 1 at the intersection of row i and column j, $1 \le i, j \le \nu$, and zeros elsewhere, with $k \in \mathscr{K}[\partial;\sigma,\Delta]$, and with $\nu = m$ for actions with columns and $\nu = p$ for actions with rows, see Lévine (2009). The elementary matrices corresponding to the operations from Definition 3 can be obtained

(i) by swapping columns (rows) i and j of the identity matrix;
(ii) by multiplying all elements of the corresponding column (row) of identity matrix by $k \in \mathscr{K}[\partial;\sigma,\Delta]$;
(iii) from identity matrix with element $k \in \mathscr{K}[\partial;\sigma,\Delta]$ in entry (i,j).
(iv) The procedure for constructing this matrix is described in the algorithm presented in Section 4.

Definition 4. A matrix $U(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{q\times q}$ is called unimodular if it has an inverse $U^{-1}(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{q\times q}$.

Every right or left unimodular matrix $U_R(\partial)$ or $U_L(\partial)$ may be obtained as a product of the corresponding elementary matrices from Definition 3.

In order to find the gcld, one may use the *left Euclidean division algorithm*, see Bronstein and Petkovšek (1996). For given two polynomials $p_1(\partial)$ and $p_2(\partial)$ with $\deg(p_1(\partial)) > \deg(p_2(\partial))$ there exist unique polynomials $\gamma_1(\partial)$ and $p_3(\partial)$ such that

$$p_1(\partial) = p_2(\partial)\gamma_1(\partial) + p_3(\partial), \quad \deg(p_3(\partial)) < \deg(p_2(\partial)).$$

Using the left Euclidean division algorithm, after $k-1$ steps, one obtains $p_i(\partial) = p_{i+1}(\partial)\gamma_i(\partial) + p_{i+2}(\partial)$ for $i = 2,\dots,k-2$ and $p_{k-1}(\partial) = p_k(\partial)\gamma_{k-1}(\partial)$. Hence the gcld of $p_1(\partial)$ and $p_2(\partial)$ is $p_k(\partial)$. Moreover, eliminating polynomials $p_{k-1}(\partial),\dots,p_3(\partial)$ we get the *Bézout identity*, i.e. there exist polynomials $u(\partial), v(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ such that $p_1(\partial)u(\partial) + p_2(\partial)v(\partial) = p_k(\partial)$. Note that the *right Euclidean division algorithm* can be defined in a similar manner.

4. JACOBSON FORM

For $P(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$ one can find elementary row and column operations corresponding to multiplication by unimodular matrices $U_L^{p\times p}(\partial)$ and $U_R^{m\times m}(\partial)$, respectively, such that

$$U_L(\partial)P(\partial)U_R(\partial) = \Lambda(\partial), \tag{9}$$

where $\Lambda(\partial) = \mathrm{diag}\,\{\lambda_1(\partial),\dots,\lambda_r(\partial),0,\dots,0\}$ and $\lambda_i(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ are unique monic polynomials obeying a property that $\lambda_{i+1}(\partial)$ is divisible by $\lambda_i(\partial)$, $\lambda_i(\partial) \parallel \lambda_{i+1}(\partial)$, i.e. there exist $\alpha_i(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ such that $\lambda_{i+1}(\partial) = \lambda_i(\partial)\cdot\alpha_i(\partial)$ for all $i = 1,\dots,r-1$. The matrix $\Lambda(\partial)$ is called the *Jacobson form* of $P(\partial)$, and $\lambda_i(\partial)$ are called the invariant polynomials of $P(\partial)$, see Nakayama (1938).

Suppose $H(\partial) \in \mathscr{K}(\partial;\sigma,\Delta)^{p\times m}$ is a transfer matrix whose entries are assumed to be in the irreducible form, i.e. without common left factors in the corresponding numerators and denominators, and write it in a standard form

$$H(\partial) = [q(\partial)]^{-1}P(\partial), \tag{10}$$

where the matrix $P(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$ is a polynomial matrix and $q(\partial)$ is the monic least common left multiple (lclm) of the denominators of all entries of $H(\partial)$. Then, $P(\partial) = q(\partial)H(\partial)$ is a polynomial matrix, that can be transformed into the Jacobson form as above.

4.1 The main Algorithm

The algorithm, presented below, allows to transform the matrix $P(\partial)$ into the Jacobson form. Consider the matrix

$$P(\partial) = \begin{pmatrix} p_{11}(\partial) & \cdots & p_{1m}(\partial) \\ \vdots & \ddots & \vdots \\ p_{p1}(\partial) & \cdots & p_{pm}(\partial) \end{pmatrix}$$

in the ring $\mathscr{K}[\partial;\sigma,\Delta]^{p\times m}$.

Step 1. $k := 1$.

Step 2. Find $p_{ij}(\partial) \neq 0$ for $i = k,\dots,p$ and $j = k,\dots,m$ with the lowest degree and, using operation (i) from Definition 3, put it on the position (k,k).

Step 3. Using elementary column (item (a)) and row (item (b)) operation (iv) from Definition 3,

(a) replace the elements $p_{kk}(\partial)$ and $p_{kj}(\partial)$ for $j = k+1,\dots,m$ with their gcld and zero, respectively. This operation can be implemented by solving the following equations

$$p_{kk}(\partial)a_{kk}(\partial) + p_{kj}(\partial)c_{jk}(\partial) = e_{kk}(\partial), \tag{11}$$

$$p_{kk}(\partial)b_{kj}(\partial) + p_{kj}(\partial)d_{jj}(\partial) = 0 \tag{12}$$

with respect to $a_{kk}(\partial), b_{kj}(\partial), c_{jk}(\partial)$ and $d_{jj}(\partial)$, and multiplying $P(\partial)$ from the right by the elementary matrix $E^4_{Rkj}(\partial)$, which can be constructed as follows. Create $m\times m$ identity matrix and put the elements $a_{kk}(\partial)$, $b_{kj}(\partial)$, $c_{jk}(\partial)$ and $d_{jj}(\partial)$ on the positions (k,k), (k,j), (j,k) and (j,j), respectively. Making $(m-k)$ replacements

[3] Notice that there exists an algorithm which allows to obtain the transfer matrix from a nonlinear system described by state-space equations, for additional information see Halás (2008).

specified above, we transform the matrix $P(\partial)$ into the new matrix with $p_{kk}(\partial) = e_{kk}(\partial)$, $p_{k,k+1}(\partial) = \cdots = p_{km}(\partial) = 0$ and some new elements $p_{il}(\partial)$ for $i = k+1,\ldots,p$ and $l = k,\ldots,m$ obtained after multiplication $P(\partial)$ by the respective matrix $U_{Rk}(\partial) = E^4_{Rk,k+1}(\partial) \cdot \ldots \cdot E^4_{Rkm}(\partial)$.

(b) replace the elements $p_{kk}(\partial)$ and $p_{ik}(\partial)$ for $i = k+1,\ldots,p$ with their gcrd and zero, respectively. The previous operation can be implemented by solving the following equations

$$a_{kk}(\partial)p_{kk}(\partial) + b_{ki}(\partial)p_{ik}(\partial) = e_{kk}(\partial), \tag{13}$$

$$c_{ik}(\partial)p_{kk}(\partial) + d_{ii}(\partial)p_{ik}(\partial) = 0 \tag{14}$$

with respect to $a_{kk}(\partial), b_{ki}(\partial), c_{ik}(\partial)$ and $d_{ii}(\partial)$, and multiplying $P(\partial)$ from the left by the elementary matrix $E^4_{Lik}(\partial)$, which can be constructed as follows. Create $p \times p$ identity matrix and put elements $a_{kk}(\partial)$, $b_{ki}(\partial)$, $c_{ik}(\partial)$ and $d_{ii}(\partial)$ on the positions (k,k), (k,i), (i,k) and (i,i), respectively. Making $(p-k)$ replacements specified above, we transform the matrix $P(\partial)$ into the new matrix with $p_{kk}(\partial) = e_{kk}(\partial)$, $p_{k+1,k}(\partial) = \cdots = p_{pk}(\partial) = 0$ and some new elements $p_{lj}(\partial)$ for $j = k+1,\ldots,m$ and $l = k,\ldots,p$ obtained after multiplication $P(\partial)$ by the respective matrix $U_{Lk}(\partial) = E^4_{Lk+1,k}(\partial) \cdot \ldots \cdot E^4_{Lpk}(\partial)$.

However, in the course of doing this, nonzero entries may reappear in the k-th row of the matrix $P(\partial)$, and one has then to repeat Step 3. Note that at each iteration the number of divisors of the element $p_{kk}(\partial)$ reduces, and therefore, in a finite number of steps the k-th row and column become zero. The latter means that after a finite number of consecutive steps one will obtain the matrix with $p_{kk}(\partial) = e_{kk}(\partial)$ and other entries in the k-th row and column equal to zero.

Step 4. If $k \neq \min(p,m) - 1$, then $k := k+1$ and go to **Step 2**, otherwise go to **Step 5**.

Step 5. If $p = m$, then go to **Step 6**, otherwise depending whether $m > p$ or $m < p$, one has to execute additional $(m-k)$ or $(p-k)$ operations over the last column(s) or row(s) described in **Steps 3(a)** or **3(b)**, respectively.

Step 6. Consider the elements of the main diagonal $p_{ii}(\partial), \ldots, p_{kk}(\partial)$. Here, the following two cases are possible:

(a) If the divisibility property holds for all pairs $p_{ii}(\partial) \parallel p_{jj}(\partial)$ for $1 \leq i < j \leq k$, then go to **Step 7**.

(b) If the divisibility property does not hold for some pair of elements $p_{ii}(\partial)$ and $p_{jj}(\partial)$ with $1 \leq i < j \leq k$, i.e. $p_{ii}(\partial) \nparallel p_{jj}(\partial)$, then, using row operation (iii) from Definition 3, the matrix $P(\partial)$ has to be transformed into a new matrix with element $p_{jj}(\partial)$ on the position (i,j) by adding the j-th row to the i-th row. After that, execute again all **Steps 2-5** with modified matrix $P(\partial)$ and $k = i$. The main idea of this transformation and the subsequent executing of the steps 2-5 consists in replacing the element $p_{ii}(\partial)$ by $\mathrm{gcld}(p_{ii}(\partial), p_{jj}(\partial))$ or $\mathrm{gcrd}(p_{ii}(\partial), p_{jj}(\partial))$, respectively, obeying a division property $p_{ii}(\partial) \parallel p_{jj}(\partial)$.

Step 7. End of the Algorithm.

Remark 2. Equations (11) and (13) are Bézout identities and can be solved using the left and right Euclidean division algorithm, respectively. Besides, equations (12) and (14) are the right and left Ore conditions, respectively. For example, for (14) it means that there exist $c_{ik}(\partial), d_{ii}(\partial) \in \mathscr{K}[\partial;\sigma,\Delta]$ such that $c_{ik}(\partial)p_{kk}(\partial) = -d_{ii}(\partial)p_{ik}(\partial)$ holds.

We have implemented the algorithm for computing Jacobson form in *Mathematica* package NLControl. However, it should be mentioned that even for the very simple examples calculations become extremely complex. Note that in our calculations, we have to simplify the obtained expressions using the relations, defined by the system equations (1) as well as those, obtained from (1) by taking the delta derivatives. If not done, the computations may lead to erroneous result.

Example. Consider the system described by the input-output equations

$$\begin{aligned} y_1^{[2]} &= u_1(1 + y_1^{\Delta}) + u_1^{\Delta}(y_1 + \mu y_1^{\Delta}) - u_2 \\ y_2^{\Delta} &= u_1 y_2 - u_2 \end{aligned} \tag{15}$$

First, we compute, according to Definition 2 and using the property (i) from Proposition 1, the transfer matrix of the system (15)

$$H(\partial) = \begin{pmatrix} \dfrac{y_1^{\sigma}\partial + y_1^{\Delta} + 1}{\partial^2 - u_1^{\sigma}\partial - u_1^{\Delta}} & \dfrac{1}{-\partial^2 + u_1^{\sigma}\partial + u_1^{\Delta}} \\ \dfrac{y_2}{\partial - u_1} & \dfrac{1}{-\partial + u_1} \end{pmatrix}.$$

Since the lclm of all the denominators in $H(\partial)$ equals to $\partial^2 - u_1^{\sigma}\partial - u_1^{\Delta}$, multiplying numerators of the elements $h_{21}(\partial), h_{22}(\partial)$ by ∂ from the left, decomposition (10) for this example takes the form

$$H(\partial) = \left(\partial^2 - u_1^{\sigma}\partial - u_1^{\Delta})\right)^{-1} \cdot \begin{pmatrix} y_1^{\sigma}\partial + y_1^{\Delta} + 1 & -1 \\ y_2^{\sigma}\partial + y_2^{\Delta} & -\partial \end{pmatrix}.$$

Obviously, the element $p_{12} = -1$ is that of the lowest possible degree of $P(\partial)$ and, after permuting the rows and columns, i.e. multiplying $P(\partial)$ by the corresponding elementary matrix [4]

$$E^1_{R12} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

from the right, we obtain

$$P(\partial) = \begin{pmatrix} -1 & y_1^{\sigma}\partial + y_1^{\Delta} + 1 \\ -\partial & y_2^{\sigma}\partial + y_2^{\Delta} \end{pmatrix}. \tag{16}$$

Next, one can easily check that $e_{11} := \mathrm{gcrd}(p_{11}, p_{12}) = 1$. After solving equations (11) and (12), corresponding to this example, i.e. the equations

$$\begin{aligned} (-1) \cdot a_{11}(\partial) + (y_1^{\sigma}\partial + y_1^{\Delta} + 1) \cdot c_{21}(\partial) &= 1 \\ (-1) \cdot b_{12}(\partial) + (y_1^{\sigma}\partial + y_1^{\Delta} + 1) \cdot d_{22}(\partial) &= 0, \end{aligned}$$

we obtain $a_{11}(\partial) = -1$, $b_{12}(\partial) = y_1^{\sigma}\partial + y_1^{\Delta} + 1$, $c_{21}(\partial) = 0$, $d_{22}(\partial) = 1$. According to Step 3(a), we construct the matrix

$$E^4_{R12}(\partial) = \begin{pmatrix} -1 & y_1^{\sigma}\partial + y_1^{\Delta} + 1 \\ 0 & 1 \end{pmatrix}$$

and multiply (16) by it from the right to get

$$\begin{pmatrix} 1 & 0 \\ \partial & -{y_1^{\sigma}}^2\partial^2 - (2y_1^{\Delta\sigma} - y_2^{\sigma} + 1)\partial - y_1^{[2]} + y_2^{\Delta} \end{pmatrix}. \tag{17}$$

Again, one can check that $e_{11} := \mathrm{gcld}(p_{11}, p_{21}) = 1$. Therefore, solving equations (13) and (14), i.e. the equations

$$\begin{aligned} a_{11}(\partial) \cdot 1 + b_{12}(\partial) \cdot \partial &= 1 \\ c_{21}(\partial) \cdot 1 + d_{22}(\partial) \cdot \partial &= 0, \end{aligned}$$

[4] In order not to mislead the reader, note that not all the operations listed in Definition 3 have been used in this example, but only those that correspond to the cases $s = 1$ and $s = 4$.

we obtain $a_{11}(\partial) = 1$, $b_{12}(\partial) = 0$, $c_{21}(\partial) = -\partial$, $d_{22}(\partial) = 1$. According to Step 3(b), we construct the matrix

$$E^4_{L21}(\partial) = \begin{pmatrix} 1 & 0 \\ -\partial & 1 \end{pmatrix}.$$

and multiply (17) by it from the left to obtain

$$\Lambda(\partial) = \begin{pmatrix} 1 & 0 \\ 0 & -y_1^{\sigma^2}\partial^2 - (2y_1^{\Delta\sigma} - y_2^{\sigma} + 1)\partial - y_1^{[2]} + y_2^{\Delta} \end{pmatrix}.$$

Due to the fact that the number of rows of $P(\partial)$ equals to the number of its columns, one can directly go to Step 6. Obviously, the division property $\lambda_1(\partial) \parallel \lambda_2(\partial)$ holds. Finally, decomposition (9) of $P(\partial)$ is

$$\Lambda(\partial) = E^4_{L21}(\partial)P(\partial)E^1_{R12}E^4_{R12}(\partial).$$

5. CONCLUSION

In this paper we have suggested a detailed algorithm for computation of the Jacobson form of the polynomial matrix associated with the transfer matrix describing the multi-input multi-output nonlinear control system, defined on homogeneous time scale, using the theory of skew polynomials. In addition, we adapted the algorithm given in Cohn (1985) for the case of the nonlinear control systems defined on homogeneous time scale. Notice that, using previous experience with *Mathematica* program, we implemented our results in NLControl package. However, the algorithm is presented in a form that can be easily implemented by any programming language.

ACKNOWLEDGMENTS

This work was partially supported by the Governmental funding project no. SF0140018s08 and Estonian Science Foundation Grants no. 8365 and 8787.

REFERENCES

E. Artin. *Geometric algebra*. Interscience publishers, New York, London, 1957.

Z. Bartosiewicz, Ü. Kotta, E. Pawłuszewicz, and M. Wyrwas. Algebraic formalism of differential one-forms for nonlinear control systems on time scales. *Proc. Estonian Acad. of Sci. Phys. Math.*, 56(3):264–282, 2007.

J. Belikov, Ü. Kotta, and A. Leibak. Transformation of the transfer matrix of the nonlinear system into the jacobson form. In *International Congress on Computer Applications and Computational Science*, pages 495–498, Singapore, December 2010.

Y.A. Blinkov, C.F. Cid, V.P. Gerdt, W. Plesken, and D. Robertz. The maple package „janet": 2. linear partial differential equations. In *Proc. of the 6th International Workshop on Computer Algebra in Scientific Computing*, pages 41–54, Passau, Germany, 2003.

M. Bohner and A. Peterson. *Dynamic Equations on Time Scales*. Birkhäuser, Boston, USA, 2001.

M. Bronstein and M. Petkovšek. An introduction to pseudo-linear algebra. *Theoretical Computer Science*, 157(1):3–33, 1996.

D. Casagrande, Ü. Kotta, M. Tõnso, and M. Wyrwas. Mathematica application for nonlinear control systems on time scales. In *International Congress on Computer Applications and Computational Science*, pages 621–624, Singapore, December 2010.

F. Chyzak, A. Quadrat, and D. Robertz. Oremodules: A symbolic package for the study of multidimensional linear systems. In *Applications of Time Delay Systems*, Lecture Notes in Control and Information Sciences, pages 233–264. Springer Berlin / Heidelberg, 2007.

P.M. Cohn. *Free rings and their relations*. Academic Press, London, UK, 1985.

G. Conte, C.H. Moog, and A.M. Perdon. *Algebraic Mehtods for Nonlinear Control Systems*. Springer-Verlag, London, UK, 2007.

B. Farb and R.K. Dennis. *Noncommutative algebra*. Springer-Verlag, New York, USA, 1993.

M. Halás. An algebraic framework generalizing the concept of transfer functions to nonlinear systems. *Automatica*, 44(5): 1181–1190, 2008.

M. Halás and Ü. Kotta. Transfer functions of discrete-time nonlinear control systems. *Estonian Acad. Sci. Phys. Math.*, 56(4):322–335, 2007a.

M. Halás and Ü. Kotta. Pseudo-linear algebra: a powerful tool in unification of the study of nonlinear control systems. In *7th IFAC Symposium on Nonlinear Control Systems,*, pages 684–689, Pretoria, South Africa, 2007b.

M.A. Insua and M. Ladra. Smith normal form can be computed using gröbner bases. In *International Congress of Mathematicians*, Madrid, Spain, August 2006.

N. Ito, W. Schmale, and H.K. Wimmer. Computation of minimal state space realizations in jacobson normal form. In *Contemporary Mathematics*, pages 221–232. American Mathematical Society Boston, MA, USA, 2003.

Ü. Kotta and M. Tõnso. Linear algebraic tools for discrete-time nonlinear control systems with mathematica. In *Nonlinear and Adaptive Control*, Lecture Notes in Control and Information Sciences, pages 195–205. Springer Berlin / Heidelberg, 2003.

Ü. Kotta, Z. Bartosiewicz, E. Pawłuszewicz, and M. Wyrwas. Irreducibility, reduction and transfer equivalence of nonlinear input-output equations on homogeneous time scales. *Systems and Control Letters*, 58(9):646–651, 2009.

Ü. Kotta, B. Rehak, and M. Wyrwas. On submersivity assumption for nonlinear control systems on homogeneous time scales. *Proc. Estonian Acad. of Sci. Phys. Math.*, 2011. Accepted for publication.

V. Levandovskyy and K. Schindelar. Computing diagonal form and jacobson normal form of a matrix using gröbner bases. http://arxiv.org/abs/1003.3785, March 2010.

J. Lévine. *Analysis and Control of Nonlinear Systems*. Springer-Verlag, Berlin, Germany, 2009.

T. Nakayama. A note on the elementary divisor theory in noncommutative domains. *Bull. Amer. Math. Soc.*, 44(10):719–723, 1938.

O. Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34:480–508, 1933.

M. Tõnso, H. Rennik, and Ü. Kotta. Webmathematica-based tools for discrete-time nonlinear control systems. *Proc. Estonian Acad. of Sci. Phys. Math.*, 58(4):224–240, 2009.

Relationship between two polynomial realization methods

Ü. Kotta M. Tõnso

*Institute of Cybernetics at TUT, Akadeemia tee 21,
12618 Tallinn, Estonia. fax : +372 620 4151
e-mails : {kotta, maris}@cc.ioc.ee*

Abstract: The aim of the paper is to show that two different polynomial realization methods, one of them based on adjoint polynomials and the other on the polynomial quotients, are equivalent. It is proved that both methods provide exactly the same set of basis vectors of the subspace determining the differentials for the state coordinates.

Keywords: nonlinear control system, state space realization, non-commutative polynomials, adjoint polynomials

1. INTRODUCTION

Identification of a nonlinear system provides a mathematical model of the system in the form of input-output (i/o) differential equation. At the same time, most of the control system theory is developed for the systems represented by the state equations. Thus, it is necessary to bring the obtained i/o equation into the state-space form. This task is called realization of the system. For nonlinear systems realization is a sophisticated problem; note that every input-output equation does not necessarily admit a state-space representation. There are several (equivalent) necessary and sufficient realizability conditions available in the literature. Among them the most known are algebraic conditions formulated in terms of integrability of the subspaces of differential one-forms, see Conte et al. (2007) and those in terms of Lie brackets by Delaleau and Respondek (1995). Recently, the theory of the noncommutative polynomial rings was applied to the realization problem. The polynomial approach is built upon the approach of differential one-forms and is most efficient from the computational point of view. The aim of the present paper is to find relations between the basis one-forms of the subspaces used to find differentials of the state coordinates in Halás and Kotta (2009) and Tõnso and Kotta (2009).

2. PROBLEM STATEMENT AND ALGEBRAIC FRAMEWORK

Consider a single-output nonlinear system, described by a higher order i/o differential equation, relating the input $u = [u_1, \ldots, u_m]^T$, the output y and a finite number of their time derivatives

$$y^{(n)} = \phi(y, \dot{y}, \ldots, y^{(n-1)}, u, \dot{u}, \ldots, u^{(n-1)}). \tag{1}$$

In (1) $u \in U \subset \mathbb{R}^m$, $y \in Y \subset \mathbb{R}$, $t \geqslant 0$ and ϕ is a real analytic function.

The realization problem is defined as follows. Given a nonlinear system, described by the i/o equation of the form (1), find, if possible, the state coordinates $x \in \mathbb{R}^n$, $x = \psi(y, \ldots, y^{(n-1)}, u, \ldots, u^{(n-1)})$ such that in these coordinates the system takes the classical state space form

$$\begin{aligned} \dot{x} &= f(x, u) \\ y &= h(x), \end{aligned} \tag{2}$$

and the sequences $\{u,\, y,\, t \geqslant 0\}$, generated by (2) (for different initial states), coincide with the sequences $\{u,\, y,\, t \geqslant 0\}$, satisfying equations (1). Then (2) is called *a realization of (1).* A system (1) is said to be *realizable* if for it exists a realization of the form (2).

Below we briefly recall the algebraic formalism, described in Conte et al. (2007). Let $\mathcal{K}$ denote the field of meromorphic functions in a finite number of the independent system variables $\{y, \ldots, y^{(n-1)}, u^{(k)},\ k \geq 0\}$ and $s : \mathcal{K} \to \mathcal{K}$ denote the time derivative operator $\mathrm{d}/\mathrm{d}t$. Then the pair $(\mathcal{K}, s)$ is a differential field Kolchin (1973). Over the field $\mathcal{K}$ one can define a differential vector space, $\mathcal{E} := \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\varphi \mid \varphi \in \mathcal{K}\}$ spanned by the differentials of the elements of $\mathcal{K}$. Consider a one-form $\omega \in \mathcal{E} : \omega = \sum_i \alpha_i \mathrm{d}\varphi_i$, $\alpha_i, \varphi_i \in \mathcal{K}$. Its derivative $\dot{\omega}$ is defined by $\dot{\omega} = \sum_i \dot{\alpha}_i \mathrm{d}\varphi_i + \alpha_i \mathrm{d}\dot{\varphi}_i$.

3. POLYNOMIAL FRAMEWORK

Polynomial framework is built upon the linear algebraic framework. The differential field $(\mathcal{K}, s)$ induces a ring of left polynomials $\mathcal{K}[Z, s]$. The elements of $\mathcal{K}[Z, s]$ can be uniquely written in the form

$$a(Z) = \sum_{i=0}^{n} a_i Z^{n-i}, \quad a_i \in \mathcal{K}$$

where Z is a polynomial indeterminate and $a(Z) \neq 0$ if and only if at least one of the functions a_i, $i = 0, \ldots, n$ is nonzero. If $a_0 \not\equiv 0$, then the positive integer n is called the degree of the polynomial $a(Z)$ and denoted by $\deg a(Z)$. In addition, we set $\deg 0 = -\infty$. For $a \in \mathcal{K}$ let us define the multiplication

$$Z \cdot a = a \cdot Z + s(a). \tag{3}$$

If the multiplication is defined by (3), the ring $\mathcal{K}[Z, s]$ is proved to satisfy left Ore condition McConnel and Robson (1987), and $Z^n \cdot a \in \mathcal{K}[Z, s]$, for $n \geq 1$, and

$$Z^n \cdot a = \sum_{i=0}^{n} C_n^i s^{n-i}(a) Z^i.$$

A ring D is called an integral domain, if it does not contain any zero divisors. This means that if a and b are two elements of D such that $ab = 0$, then $a = 0$ or $b = 0$.

Lemma 1. McConnel and Robson (1987)

(i) The ring $\mathcal{K}[Z, s]$ is an integral domain.
(ii) If a and b are nonzero left polynomials, then $\deg(a\,b) = \deg a + \deg b$.

Definition 2. Abramov et al. (2005) The adjoint of a Ore polynomial ring $\mathcal{K}[Z, s]$ is defined as the Ore polynomial ring $\mathcal{K}[Z^*, s^*]$, where $s^* = -s$.

From the definition it follows that in the adjoint polynomial ring multiplication is defined by the commutation rule $Z^* \cdot a = a \cdot Z^* - \dot{a}$, where $a \in \mathcal{K}$. If

$$p(Z) = p_n Z^n + \ldots + p_1 Z + p_0 \tag{4}$$

is a polynomial in $\mathcal{K}[Z, s]$ then the adjoint polynomial $p^*(Z^*)$ is defined by the formula

$$p^*(Z^*) = Z^{*n} p_n + \ldots + Z^* p_1 + p_0 \in \mathcal{K}[Z^*, s^*], \tag{5}$$

where the products $Z^{*i} p_i$ must be computed in $\mathcal{K}[Z^*, s^*]$.

For $\Phi \in \mathcal{K}$ we define $\mathrm{d} : \mathcal{K} \to \mathcal{E}$ as follows:

$$\mathrm{d}\Phi := \sum_{i=0}^{n-1} \frac{\partial \Phi}{\partial y^{(i)}}\, \mathrm{d}y^{(i)} + \sum_{j=1}^{m} \sum_{l=0}^{k} \frac{\partial \Phi}{\partial u_j^{(l)}}\, \mathrm{d}u_j^{(l)}.$$

$\mathrm{d}\Phi$ is said to be the total differential (or simply the differential) of the function Φ and it is a differential one-form. It is proved in Conte et al. (2007) that $s(\mathrm{d}\Phi) = \mathrm{d}(s\Phi)$. Let us define $Z^k \mathrm{d}y := \mathrm{d}(s^k y)$ and $Z^l \mathrm{d}u := \mathrm{d}(s^l u)$, for $k, l \geq 0$ in the vector space $\mathcal{E}$. Since every one-form $\omega \in \mathcal{E}$ has the form

$$\omega = \sum_{i=0}^{n-1} a_i \mathrm{d}y^{(i)} + \sum_{j=1}^{m} \sum_{l=0}^{k} b_{j,l} \mathrm{d}u_j^{(l)},$$

where $a_i, b_j \in \mathcal{K}$, so ω can be expressed in terms of the left polynomials

$$\omega = \left(\sum_{i=0}^{n-1} a_i Z^i \right) \mathrm{d}y + \sum_{j=1}^{m} \left(\sum_{l=0}^{k} b_{j,l} Z^l \right) \mathrm{d}u_j.$$

A polynomial can be considered as an operator acting on the elements of $\mathcal{E}$:

$$\left(\sum_{i=0}^{k} a_i Z^i \right) (\alpha \mathrm{d}\nu) := \sum_{i=0}^{k} a_i \left(Z^i \cdot \alpha \right) \mathrm{d}\nu,$$

with $a_i, \alpha \in \mathcal{K}$ and $\mathrm{d}\nu \in \{\mathrm{d}y, \mathrm{d}u_1, \ldots, \mathrm{d}u_m\}$. It is easy to notice that $Z(\omega) = s(\omega)$, for $\omega \in \mathcal{E}$. Additionally, using the induction principle, one can show that $Z^n(\mathrm{d}\Phi) = \mathrm{d}(s^n \Phi)$.

Instead of working with equation (1), describing the control system, we can work with its differential

$$\mathrm{d}y^{(n)} - \sum_{i=0}^{n-1} \frac{\partial \phi}{\partial y^{(i)}} \mathrm{d}y^{(i)} = \sum_{j=1}^{m} \sum_{i=0}^{n-1} \frac{\partial \phi}{\partial u_j^{(i)}} \mathrm{d}u_j^{(i)} \tag{6}$$

that can be rewritten as

$$a(Z)\mathrm{d}y = \sum_{j=1}^{m} b_j(Z) \mathrm{d}u_j, \tag{7}$$

with

$$a(Z) = Z^n - \sum_{i=0}^{n-1} \frac{\partial \phi}{\partial y^{(i)}} Z^i, \quad b_j(Z) = \sum_{i=0}^{n-1} \frac{\partial \phi}{\partial u_j^{(i)}} Z^i \tag{8}$$

and $a(Z), b_j(Z) \in \mathcal{K}[Z, s]$ for $j = 1, \ldots, m$.

4. REALIZABILITY CONDITIONS

Realizability conditions in terms of adjoint polynomials can be found in Halás and Kotta (2009) for single-input single-output (SISO) systems and in Halás and Kotta (2011) for multi-input single-output systems:

Theorem 3. Given a nonlinear control system defined by i/o equation (1), or equivalently by (6), let

$$\omega_i := \sum_{j=1}^{m} b_{j,i-1}^* \mathrm{d}u_j - a_{i-1}^* \mathrm{d}y, \quad i = 1, \ldots, n. \tag{9}$$

Then there exists a state space realization of the form (2) if and only if

$$\begin{aligned} \mathrm{span}_{\mathcal{K}}\{&\mathrm{d}y, \mathrm{d}\dot{y} - \omega_n, \mathrm{d}\ddot{y} - \dot{\omega}_n - \omega_{n-1}, \ldots, \\ &\mathrm{d}y^{(n-1)} - \omega_n^{(n-2)} - \omega_{n-1}^{(n-3)} - \ldots - \omega_2, \} \end{aligned} \tag{10}$$

is integrable.

Realizability conditions based on division of non-commutative polynomials are given in Tõnso and Kotta (2009) for SISO systems and in Belikov et al. (2011) for multi-input multi-output systems:

Theorem 4. Given a nonlinear control system defined by i/o equation (1), or equivalently by (6), let

$$\bar{\omega}_l := \left[\bar{a}_l(Z), -\bar{b}_{\cdot,l}(Z) \right] \begin{bmatrix} \mathrm{d}y \\ \mathrm{d}u \end{bmatrix}, \quad l = 1, \ldots, n \tag{11}$$

where $\bar{a}_l(Z)$ and $\bar{b}_{\cdot,l}(Z)$ can be computed recursively from

$$\begin{aligned} \bar{a}_{l-1}(Z) &= Z\,\bar{a}_l(Z) + r_l \\ \bar{b}_{\cdot,l-1}(Z) &= Z\,\bar{b}_{\cdot,l}(Z) + \rho_{\cdot,l}, \end{aligned} \tag{12}$$

with the initial polynomials

$$\bar{a}_0(Z) = a(Z), \quad \bar{b}_{\cdot,0}(Z) = [b_1(Z), \ldots, b_m(Z)].$$

Then there exists a state space realization of the form (2) if and only if

$$\mathrm{span}_{\mathcal{K}}\{\bar{\omega}_l,\, l = 1, \ldots, n\} \tag{13}$$

is integrable.

State coordinates necessary for realization can be found by integrating the basis vectors of the subspaces (10) or (13). Of course, one cannot find the integrable one-forms $\mathrm{d}x_1(t), \ldots, \mathrm{d}x_n(t)$ for an arbitrary i/o equation. No matter which way the the subspace is calculated, either by Theorem 3 or by Theorem 4, its integrability can be checked by the Frobenius theorem.

Theorem 5. Choquet-Bruhat et al. (1989)(Frobenius) Let $\mathcal{V} = \mathrm{span}_{\mathcal{K}}\{\omega_1, \ldots, \omega_r\}$ be a subspace of $\mathcal{E}$. $\mathcal{V}$ is closed iff $\mathrm{d}\omega_k \wedge \omega_1 \wedge \ldots \wedge \omega_r = 0$, for all $k = 1, \ldots, r$.

5. MAIN RESULT

Since Theorems 3 and 4 both present necessary and sufficient realizability conditions, these conditions are obviously equivalent. The goal of this section is to show the precise relationship between the codistribution (10) and (13).

Proposition 6. The basis one-forms (10) in Theorem 3 coincide with the one-forms $\bar{\omega}_l$, $l = 1, \ldots, n$ in Theorem 4.

Proof. First note that basis one-forms of the codistribution (10) can be written in a form

$$\mathrm{d}y^{(i-1)} - \omega_n^{(i-2)} - \ldots - \dot{\omega}_{n-i+3} - \omega_{n-i+2},$$

where $i = 1, \ldots, n$. Definition (9) allows us to rewrite the basis vectors in the form:

$$(Z^{i-1} + Z^{i-2}a^*_{n-1} + \ldots + Za^*_{n-i+2} + a^*_{n-i+1})\mathrm{d}y$$
$$-\sum_{j=1}^{m}(Z^{i-2}b^*_{j,n-1} + \ldots + Zb^*_{j,n-i+2} + b^*_{j,n-i+1})\mathrm{d}u_j \qquad (14)$$

Next, observe that the (polynomial) coefficient of $\mathrm{d}y$, denoted by $\tilde{a}_{n-i+1}(Z)$, is similar to the polynomial

$$a(Z) = Z^n + Z^{n-1}a^*_{n-1} + \ldots + Za^*_1 + a^*_0.$$

We only need to multiply $\tilde{a}_{n-i+1}(Z)$ from left by Z^{n-i+1} and add a missing part $z^{n-i}a^*_{n-i} + \ldots + Za^*_1 + a^*_0$. The latter means

$$Z^{n-i+1}\tilde{a}_{n-i+1}(Z) + Z^{n-i}a^*_{n-i} + \ldots + Za^*_1 + a^*_0 = a(Z).$$

For the sake of simplicity we can replace in previous equality the index i by $l = n - i + 1$, $l = 1, \ldots n$, keeping in mind that it just means reversing the order of the coefficients:

$$Z^l\tilde{a}_l(Z) + Z^{l-i}a^*_{l-1} + \ldots + Za^*_1 + a^*_0 = a(Z).$$

Thus, we have represented a coefficient $\tilde{a}_l(Z)$ of $\mathrm{d}y$ as a left quotient of $a(Z)$ and Z^l, $l = 1, \ldots, n$. Note that such quotients can be computed using the recursive formula

$$Z\tilde{a}_l(Z) + a^*_{l-1} = \tilde{a}_{l-1}(Z).$$

Since the quotient of two polynomials is unique, the polynomial $\tilde{a}_l(Z)$ has to be equal to $\bar{a}_l(Z)$ in (12), while $r_l = a^*_{l-1}$. Analogously, it is possible to prove that the coefficients of $\mathrm{d}u_j$ for $j = 1, \ldots, m$ in (14) equal to $\bar{b}_{\cdot,l}(Z)$ in (12). Thus, we have shown that the basis one-forms (10) and (11) coincide, except that the order is reversed. ■

Remark about the discrete-time case

Realizability conditions for discrete-time systems are largely analogous to their continuous-time counterparts. The analogue of Theorem 3 for discrete-time SISO systems can be found in Halás and Kotta (2010); the only difference is that in basis (10) instead of the derivative operator there is the forward shift operator δ. The forward shift operator is defined by shifting the arguments of the function according to the rules $\delta y(t) = y(t+1)$, $\delta u_j(t) = u_j(t+1)$, $j = 1, \ldots, m$. The inverse operator of δ is denoted by δ^{-1} and called backward-shift operator. We should also keep in mind that in discrete-time case multiplication of polynomials and adjoint polynomials is defined by different commutation rules, $Z \cdot a = \delta(a) \cdot Z$ and $Z^* \cdot a = \delta^{-1}(a) \cdot Z^*$, respectively. Due to the latter commutation rule, for an adjoint polynomial $p^*(Z^*) = p^*_n Z^{*n} + \ldots + p^*_1 Z^* + p^*_0$, the equalities $p^*_i = \delta^{-i}(p_i)$, $i = 0, \ldots, n$ hold. Therefore, for the discrete-time systems, the equality (14) takes the form

$$(Z^{i-1} + \delta^{i-n-1}(a_{n-1})Z^{i-2} + \ldots + \delta^{i-n-1}(a_{n-i+1}))\mathrm{d}y$$
$$-\sum_{j=1}^{m}(\delta^{i-n-1}(b_{j,n-1})Z^{i-2} + \ldots + \delta^{i-n-1}(b^*_{j,n-i+1}))\mathrm{d}u_j$$
$$(15)$$

for $i = 1, \ldots, n$. The coefficients of $\mathrm{d}y$ and $\mathrm{d}u_j$, $j = 1, \ldots, m$ can be computed by polynomial division as in Theorem 4; however, the shorter way is to use the cut-and-shift operator defined by

$$\delta_c^{-1}(p(Z)) = \delta^{-1}(p(Z) - p_0), \qquad (16)$$

see also Kotta and Tõnso (2008). In terms of the cut-and-shift operator the equality (15) may expressed as

$$\delta_c^{-l}a(Z)\mathrm{d}y - \sum_{j=1}^{m}\delta_c^{-l}(b_{j,l})\mathrm{d}u_j = \delta_c^{-l}[a(Z), -b(Z)]\begin{bmatrix}\mathrm{d}y\\ \mathrm{d}u\end{bmatrix} \qquad (17)$$

for $l = 1, \ldots, n$. The latter formula agrees with the result in Kotta and Tõnso (2008).

6. EXAMPLES

Example 1. Consider the control system $\ddot{y} = u + y\dot{u}^2$. Let us compute the differentials of state coordinates necessary for realization by two different methods, by Theorem 3 and Theorem 4. As a common step of both methods, one has to find the polynomial representation of the system, i.e polynomials $a(Z)$ and $b_1(Z) = b(Z)$:

$$\begin{aligned} a(Z) &= Z^2 - \dot{u}^2 \\ b(Z) &= 2y\dot{u}Z + 1. \end{aligned}$$

Following Theorem 3, one has to compute adjoint polynomials $a^*(Z^*)$ and $b^*(Z^*)$. By (5),

$$\begin{aligned} a^*(Z^*) &= Z^{*2} + \dot{u}^2 \\ b^*(Z^*) &= 2y\dot{u}Z^* + (1 - 2\dot{y}\dot{u} - 2y\ddot{u}). \end{aligned}$$

After computing ω_1 and ω_2 defined by (9)

$$\begin{aligned} \omega_1 &= (1 - 2\dot{y}\dot{u} - 2y\ddot{u})\mathrm{d}u - \dot{u}^2\mathrm{d}y \\ \omega_2 &= 2y\dot{u}\mathrm{d}u \end{aligned}$$

it is easy to write down the basis one-forms (14)

$$\mathrm{span}_{\mathcal{K}}\{\mathrm{d}y,\, \mathrm{d}\dot{y} - \omega_2\} = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}y,\, \mathrm{d}\dot{y} - 2y\dot{u}\,\mathrm{d}u\}. \qquad (18)$$

According to Frobenius condition the latter subspace is not integrable and thus the system does not admit the classic state space representation.

Alternatively, one may follow Theorem 4 and find $\bar{a}_l(Z)$, $\bar{b}_l(Z)$, $l = 1, \ldots, n$ by dividing $a(Z)$ and $b(Z)$ from the left by the polynomial Z repeatedly:

$$\begin{aligned} \bar{a}_0(Z) &= Z^2 - \dot{u}^2 & \bar{b}_0(Z) &= 2y\dot{u}Z + 1 \\ \bar{a}_1(Z) &= Z & \bar{b}_1(Z) &= 2y\dot{u} \\ \bar{a}_2(Z) &= 1 & \bar{b}_2(Z) &= 0 \end{aligned}$$

By (11),

$$\begin{aligned} \bar{\omega}_1 &= Z\mathrm{d}y - 2y\dot{u}\mathrm{d}u = \mathrm{d}\dot{y} - 2y\dot{u}\mathrm{d}u \\ \bar{\omega}_2 &= \mathrm{d}y, \end{aligned}$$

which coincide with the basis one-forms of (18).

Example 2. Consider the control system

$$y^{(3)} = u_1\dot{y} + y\dot{u}_1 + \dot{u}_2^2 + u_2\ddot{u}_2 \qquad (19)$$

that can be described as in (8) as follows:

$$\begin{aligned} a(Z) &= Z^3 - u_1 Z \\ b_1(Z) &= Z + \dot{y} \\ b_2(Z) &= u_2 Z^2 + 2\dot{u}_2 Z + \ddot{u}_2. \end{aligned} \qquad (20)$$

Obeying Theorem 3, the adjoint polynomials $a^*(Z^*)$, $b^*_1(Z^*)$ and $b^*_2(Z^*)$ are as follows:

$$\begin{aligned} a^*(Z^*) &= Z^{*3} - u_1 Z^* + \dot{u}_1 \\ b^*_1(Z^*) &= Z^* + \dot{y} \\ b^*_2(Z^*) &= u_2 Z^{*2}. \end{aligned}$$

Computation of ω_i's, defined by (9), yields

$$\begin{aligned} \omega_1 &= \dot{y}\mathrm{d}u_1 - \dot{u}_1\mathrm{d}y \\ \omega_2 &= \mathrm{d}u_1 + u_1\mathrm{d}y \\ \omega_3 &= u_2\mathrm{d}u_2, \end{aligned} \tag{21}$$

which allow to find the basis one-forms defined by (10):

$$\mathrm{span}_{\mathcal{K}}\{\mathrm{d}y,\ \mathrm{d}\dot{y}-\omega_3, \mathrm{d}\ddot{y}-\dot{\omega}_3-\omega_2\} = \\ \mathrm{span}_{\mathcal{K}}\{\mathrm{d}y,\ \mathrm{d}\dot{y}-u_2\mathrm{d}u_2, \mathrm{d}\ddot{y}-\dot{u}_2\mathrm{d}u_2-u_2\mathrm{d}\dot{u}_2-\mathrm{d}u_1-u_1\mathrm{d}y\}. \tag{22}$$

Alternative solution starts by computing quotients by (12)

$$\begin{aligned} &\bar{a}_0(Z) = Z^3 - u_1 Z && \bar{b}_{\cdot,0} = [\,Z+\dot{y},\ u_2Z^2+2\dot{u}_2Z+\ddot{u}\,] \\ &\bar{a}_1(Z) = Z^2 - u_1 && \bar{b}_{\cdot,1} = [\,1,\ u_2Z+\dot{u}_2\,] \\ &\bar{a}_2(Z) = Z && \bar{b}_{\cdot,2} = [\,0,\ u_2\,] \\ &\bar{a}_3(Z) = 1 && \bar{b}_{\cdot,3} = [\,0,\ 0\,], \end{aligned} \tag{23}$$

that allow to find the basis one-forms $\bar{\omega}_1$, $\bar{\omega}_2$ and $\bar{\omega}_3$ according to (11):

$$\begin{aligned} \bar{\omega}_1 &= (Z^2-u_1)\mathrm{d}y - \mathrm{d}u_1 - (u_2Z-\dot{u}_2)\mathrm{d}u_2 \\ &= \mathrm{d}\ddot{y} - u_1\mathrm{d}y - \mathrm{d}u_1 - u_2\mathrm{d}\dot{u}_2 - \dot{u}_2\mathrm{d}u_2 \\ \bar{\omega}_2 &= Z\mathrm{d}y - 0\cdot\mathrm{d}u_1 - u_2\mathrm{d}u_2 \\ &= \mathrm{d}\dot{y} - u_2\mathrm{d}u_2 \\ \bar{\omega}_3 &= \mathrm{d}y. \end{aligned}$$

As expected, the above one-forms coincide with (22). According to Frobenius condition, the subspace $\mathrm{span}_{\mathcal{K}}\{\bar{\omega}_1, \bar{\omega}_2, \bar{\omega}_3\}$ is integrable and the differentials of the state coordinates

$$\begin{aligned} \mathrm{d}x_1 &= \bar{\omega}_3 = \mathrm{d}y \\ \mathrm{d}x_2 &= \bar{\omega}_2 = \mathrm{d}(\dot{y} - \frac{1}{2}u_2^2) \\ \mathrm{d}x_3 &= \bar{\omega}_1 + u_1\bar{\omega}_3 = \mathrm{d}(\ddot{y} - u_1 - u_2\dot{u}_2) \end{aligned}$$

yield the classical state equations

$$\begin{aligned} \dot{x}_1 &= \frac{1}{2}u_2^2 + x_2 \\ \dot{x}_2 &= u_1 + x_3 \\ \dot{x}_3 &= u_1(\frac{1}{2}u_2^2 + x_2). \end{aligned}$$

7. MATHEMATICA IMPLEMENTATION

We have implemented both realization methods in computer algebra system Mathematica. On that purpose we use the functions from the package NLControl, allowing to solve various modelling, analysis and synthesis problems for nonlinear control systems, see Tõnso et al. (2009); Tõnso (2010). NLControl package also includes the basic tools for polynomials from Ore rings. If compared with the Maple OreTools package, the polynomial functions in NLControl have one essential benefit. Namely, NLControl allows take into account that the derivative (or shift) operator is defined by the control systems equations, see Halás et al. (2009). The OreTools package lacks such possibility, therefore it may sometimes provide a wrong result when applied to nonlinear control problems.

The given Mathematica code below has two advantages worth to mention. First, it is constructed in a way it can handle both continuous- and discrete-time systems; in case of discrete-time systems one has just to replace the word `TimeDerivative` by the word `Shift`. Second, though the scope of this paper is limited to single-output systems, the above code allows to find state coordinates for multi-output systems, too. Therefore an additional pair of curly braces may appear around Mathematica expressions below. Consider the system (19). After loading the package by the command

```
<<NLControl`Master`
```

let us create the object `IO[]`, representing the i/o equations for this system.

```
eqs = {y'''[t]->u1[t]y'[t]+u1'[t]+u2'[t]^2+
u2[t]u2''[t]};
Ut = {u1[t],u2[t]};
Yt = {y[t]};
ioeq = IO[eqs,Ut,Yt,t,TimeDerivative]
```

First, let us compute the state coordinates by Theorem 3. For that we need to construct the Ore ring and adjoint Ore ring associated with the system `ioeq`.

```
R = DefineOreRing[Z, ioeq];
adR = DefineAdjointOreRing[Z, ioeq];
```

The function `FromIOToOreP` finds the polynomials $a(Z)$ and $b_j(Z)$, $j = 1, \ldots, m$ for the system `ioeq`. Note that the polynomials $b_j(Z)$ are chosen with the opposite sign to (7) and (20).

```
AB = MapThread[Join, FromIOToOreP[ioeq]]
```

```
{{OreP[1,0,-u1[t],0],
  OreP[-1,-y'[t]],
  OreP[-u2[t],-2 u2'[t],-u2''[t]]}}
```

The function `OreP[`p_n`, ...,`p_1`, `p_0`]` represents the Ore polynomial in the form (4). The function `Adjoint` allows to compute adjoints of the polynomials being elements of the matrix `AB`.

```
adAB = Map[Adjoint[#, adR]&, AB, 2]
```

```
{{OreP[1,0,-u1[t],u1'[t]], OreP[-1,-y'[t]],
  OreP[-u2[t],0,0]}}
```

By the next code row the coefficients of the polynomials are represented as the individual 0-degree polynomials. The constant terms (the last argument of `OreP[]`) are removed, since they correspond to ω_1, which is not necessary in further computations.

```
adAB = Map[ If[ Head[#]===OreP,
Drop[ List @@ OreP /@ #, -1], ]&, adAB, 2]
```

```
{{{1, 0, OreP[-u1[t]]}, {OreP[-y[t]]},
  {OreP[-u2[t]], 0}}}
```

To obtain the the polynomials, respective to the one-forms ω_i, defined by (9), we need to equalize the length of the rows by adding the missing zeros to the beginning of each row and the transpose the obtained matrix.

```
omega = Transpose/@ PadLeft/@ adAB
```

```
{{{1,0,0}, {0,0,OreP[-u2[t]]},
  {OreP[-u1[t]],-1,0}}}
```

In above output the vector {1,0,0} corresponds to dy, which can be considered as $\omega_{n+1} = \omega_4$, the vector {0,0,OreP[-u2[t]]} corresponds to $-\omega_3$ and {OreP[-u1[t]], OreP[-y[t]],0} to $-\omega_2$, given by (21). Our next task is to compute the basis one-forms (10). Denoting the basis elements by $\theta_1, \ldots, \theta_n$ (and recalling that our ω_i, $i = 2, \ldots, n$ are of opposite signs, to (9)), allows to compute the basis by the time-saving recursive formula $\theta_1 = dy$, $\theta_{i+1} = \dot{\theta}_i + \omega_{n-i+1}$ for $i = 1, \ldots, n-1$. Instead of computing the derivatives of θ_i, $i = 1, \ldots, n-1$ we have chosen to multiply the respective polynomials by s from right. The reason for such choice is that the routines available in NLControl are more suitable for this method and it allows to avoid several clumsy data transformation.

```
statedifpoly = Rest @ FoldList[
OreMultiply[ OreP[1,0],#1,K]+#2&, 0, #]&
/@ omega
```

```
{{{1,0,0}, {OreP[1,0],0,OreP[-u2[t]]},
  {OreP[1,0,-u1[t]], -1,
  OreP[-u2[t],-u2′[t]]}}}
```

The function FromOrePToSpanK converts the list of Ore polynomials into the set of one-forms.

```
statedif = FromOrePToSpanK[
Join @@ statedifpoly, ioeq]
```

```
SpanK[{{1,0,0,0,0,0}, {0,1,0,0,-u2[t],0},
  {-u1[t],0,1,-1,-u2′[t],-u2[t]}},
  {y[t],y′[t],y′′[t],u1[t],u2[t],u2′[t]},
  -1, t]
```

We have computed the basis of the subspace (10). The function BookForm allows to print the result in a user-friendly form:

```
BookForm[statedif]
```

```
SpanK{dy, dy′ - u2 du2,
 -u1 dy + dy′′- du1 - u2′du2 - u2 du2′}
```

Integrating the one-forms

```
states = IntegrateOneForms[statedif]
```

```
{y[t], 1/2 (u2[t]^2 - 2y′[t]),
  u1[t] + u2[t]u2′[t] - y′′[t]}
```

yields the state coordinates, which allow to find the state equations:

```
BookForm[Realization[ioeq,
{x1[t],x2[t],x3[t]}, states]]
```

```
x1′ = (u2^2 - 2*x2)/2
x2′ = -u1 + x3
x3′ = -((u2^2 - 2*x2)*u1)/2
  y = x1

 x1 = y
 x2 = (u2^2 - 2*y′)/2
 x3 = u1 + u2 u2′ - y′′)
```

The basis (10) can be also found by the single function StateDifferentialsAdjoint[ioeq] (the name of the function may be confusing, because in fact, it gives the *linear combination* of the differentials of the state coordinates.) The realization procedure can be performed by the function Realization[ioeq, {x1[t],x2[t],...}, Method->Adjoint].

Second, let us find the state coordinates by Theorem 4. We can employ the Ore ring R and the matrix AB already computed above. The order of the system can be determined by the function MaxPLMOrder, which allows to find the maximal order of time-derivative, shift or any other pseudo-linear map.

```
ni = MaxPLMOrder[eqs, #, TimeDerivative
]& /@ Yt
```

```
{3}
```

According to (12) we have to divide the elements of the matrix AB repeatedly by the polynomial s from left. By abuse of function name, we have defined the function CutAndShift[$p(Z)$,R] as a left quotient of $p(Z)$ and s for the Ore rings, associated with the continuous-time systems. The reason is that in case of discrete-time systems the quotients can be found by applying cut-and-shift operator as in (17). This extension allows to compute the polynomials $\bar{a}_l$ and $\bar{b}_{\cdot,l}$, $l = 1, \ldots, n$ in (12) for both, continuous- and discrete time systems, by the following compact row:

```
ABquot = MapThread[ NestList[
CutAndShift[#, K]&, #1, #2]&, AB, ni]
```

```
{{{OreP[1,0,-u1[t],0],
   OreP[-1,-y′[t]],
   OreP[-u2[t],-2 u2′[t],-u2′′[t]]},
 {OreP[1,0,-u1[t]], -1,
   OreP[-u2[t],-u2′[t]]},
 {OreP[1,0],0,OreP[-u2[t]]},
 {1,0,0}}}
```

The obtained list corresponds to $\{\{\{\bar{a}_0(Z), \bar{b}_{\cdot,0}(Z)\}, \ldots,$ $\{\bar{a}_3(Z), \bar{b}_{\cdot,3}(Z)\}\}\}$, where the polynomials are given by (23). Removing the first row $\{\bar{a}_0(Z), \bar{b}_{\cdot,0}(Z)\}$ and reversing the order of the remaining rows yields

```
ABquot = Reverse /@ Rest /@ ABquot
```

```
{{{1, 0, 0}, {OreP[1, 0], 0, OreP[-u2[t]]},
 {OreP[1, 0, -u1[t]], -1,
    OreP[-u2[t], -u2′[t]]}}}
```

The funtion FromOrePToSpanK converts Ore polynomials into the set of one-forms.

```
FromOrePToSpanK[ Join @@ ABquot, ioeq]
```

```
SpanK[{{1,0,0,0,0,0}, {0,1,0,0,-u2[t],0},
  {-u1[t],0,1,-1,-u2′[t],-u2[t]}},
  {y[t],y′[t],y′′[t],u1[t],u2[t],u2′[t]},
   -1, t]
```

By that the basis of the subspace (13) has been computed. The one-forms (13) can be also found by the function StateDifferentialsLeftQuoteint[ioeq] and the realization can be performed by Realization[ioeq, {x1[t],x2[t]...}, Method->LeftQuotient].

Preliminary comparison of two realization methods, based on relatively small examples, suggests that the program employing adjoint polynomials works faster than the one based on the polynomial quotients. The reason is obvious

– adjoint polynomial method utilizes only polynomial multiplication, which is a primary function (it is immaterial whether we work with polynomials or with adjoint polynomials) and is therefore performed faster than polynomial division.

However, in discrete-time case the situation is different. Polynomial quotient operator can be replaced by cut-and-shift operator, which means that the program can compute the basis one-forms without calling neither multiplication nor division functions and therefore, it can produce the results faster than the program involving adjoint polynomials.

8. CONCLUSION

In this paper the realizability problem of nonlinear control system has been addressed. The paper focuses on establishing the explicit relationship between the two necessary and sufficient realizability conditions. Both conditions are formulated in terms of integrability of a certain subspace. While in the first condition the basis one-forms of the subspace are achieved from the adjoints of the polynomials describing the system; in the second condition the basis one-forms are found using polynomial quotients. As both conditions are necessary and sufficient it is obvious that they are equivalent; however, in this paper it is proved that the basis one-forms used in both conditions are equal.

ACKNOWLEDGMENTS

This work was partially supported by the ESF Grant no 8787 and target funding project of Estonian Ministry of Education and Research SF0140018s08.

REFERENCES

S.A. Abramov, H.Q. Le, and Z. Li. Univariate ore polynomial rings in computer algebra. *Journal of Mathematical Sciences*, 131(5):5885–5903, 2005.

J. Belikov, M. Tõnso, and Ü. Kotta. Realization of MIMO nonlinear equations: polynomial approach. 2011. Submitted for publication.

Y. Choquet-Bruhat, C. DeWitt-Morette, and M. Dillard-Bleick. *Analysis, Manifolds and Physics, Part I: Basics.* North Holland, Amsterdam, 1989.

G. Conte, C. Moog, and A. Perdon. *Algebraic Methods for Nonlinear Control Systems.* Springer-Verlag, London, 2007.

E. Delaleau and W. Respondek. Lowering the orders of derivatives of controls in generalised state space systems. *Journal of Mathematical Systems, Estimation and Control*, 5:1–27, 1995.

M. Halás and Ü. Kotta. Realization problem of SISO nonlinear systems: a transfer function approach. In *IEEE International Conference on Control and Automation*, pages 546–551. Christchurch, New Zealand, 2009.

M. Halás and Ü. Kotta. Extension of the transfer function approach to the realization problem of nonlinear systems to discrete-time case. In *8th IFAC Symposium on Nonlinear Control Systems, University of Bologna*, pages 179–184, Bologna, Italy, 2010.

M. Halás and Ü. Kotta. A polynomial approach to the realization problem of nonlinear systems. 2011. Submitted for publication.

M. Halás, Ü. Kotta, Z. Li, H. Wang, and C. Yuan. Submersive rational difference systems and their accessibility. In *Proceedings of the 2009 international symposium on Symbolic and algebraic computation*, ISSAC '09, pages 175–182, 2009.

E.R. Kolchin. *Differential algebra and algebraic groups.* Academic Press, 1973.

Ü. Kotta and M. Tõnso. Realization of discrete-time nonlinear input-output equations: polynomial approach. In *Proc. of the 7th World Congress on Intelligent Control and Automation*, pages 529–534, Chongqing, China, 2008.

J.C. McConnel and J.C. Robson. *Noncommutative Noetherian Rings.* Birkhäuser, 1987.

M. Tõnso. *Computer Algebra Tools for Modelling, Analysis and Synthesis for Nonlinear Control Systems.* PhD thesis, Tallinn University of Technology, Institute Of Cybernetics, 2010.

M. Tõnso and Ü. Kotta. Realization of continuous-time nonlinear inputoutput equations: polynomial approach. In *Computer Aided Systems Theory - Eurocast 2009, Lecure Notes in Computer Science*, pages 633–640. Springer Berlin/Heidelberg, 2009.

M. Tõnso, H. Rennik, and Ü. Kotta. Webmathematica based tools for discrete-time nonlinear control systems. *Proceedings of the Estonian Academy of Sciences*, 58(4): 224–240, 2009.

Accessibility and Feedback Linearization for SISO Discrete-Time Nonlinear Systems: New Tools [⋆]

Ülle Kotta [*] Tanel Mullari [*] Alexey Ye. Shumsky [**]
Alexey N. Zhirabok [***]

[*] *Institute of Cybernetics at Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia (e-mail: kotta@cc.ioc.ee).*
[**] *Institute of Applied Mathematics, Far Eastern Branch of Academy of Sciences, Radio street 7, 690041 Vladivostok, Russia (e-mail: shumsky@mail.primorye.ru)*
[***] *Dept. of Design and Technology of Radio Equipment, Far Eastern Federal University, Pushkinskaya street 10, 690950 Vladivostok, Russia (e-mail: zhirabok@mail.ru)*

Abstract: The tools of the algebra of functions are applied to readdress the accessibility and static state feedback linearization problems for discrete-time nonlinear control systems. These tools are also applicable for nonsmooth systems. Moreover, the close connections are established between the new results and those based on differential one-forms.

Keywords: Feedback linearization, state feedback, nonlinear control systems, discrete-time systems, algebraic approaches

1. INTRODUCTION

The approach based on the vector spaces of differential one-forms over suitable differential/difference fields of nonlinear functions offers the complementary (dual) tools to the differential geometric methods for studying the nonlinear control systems, either continuous- or discrete-time, see Conte et al. [2007]. These tools are characterized by their inherent simplicity, universality and strong similarity to their linear counterparts.

However, there exists another mathematical approach that relies on a certain algebraic structure, called the algebra of functions, see Zhirabok and Shumsky [2008]. The main idea for developing the algebra of functions traces back to the book by Hartmanis and Stearns [1966], who introduced the algebra of partitions for finite automata defined via the transition tables or graphs. In the algebra of functions the partitions were replaced by functions generating them and the analogous operations and operators for functions were introduced. The four key elements of the algebra of functions are partial preorder relation, binary operations (sum and product, defined in a specific manner), binary relation and certain operators $\mathbf{m}$ and $\mathbf{M}$. The first two elements are defined on the arbitrary set of vector functions whereas the other two are defined for functions with the domain being the state space of the control system. Like the tools based on the differential forms, the algebra of functions provides a unified viewpoint to study the discrete-time as well as the continuous-time control systems; additionally it allows to address also the discrete-event systems like those in Shumsky and Zhirabok [2010a,b]. An important point to stress is that these tools (unlike most previous methods) do not require the system to be described in terms of smooth functions.

The goal of this paper is to compare the tools of the algebra of functions with those based on the differential forms. Our purpose is to compare the assumptions made on the control system, the basic algorithms and the solutions of few chosen control problems, like accessibility and static state feedback linearization. In order to focus on the key aspects and keep the presentation simple, we restrict ourselves in this paper to the discrete-time single-input systems.

Whereas the number of publications on the topic of static state feedback linearization is huge, the situation is different for the discrete-time case, see Jayaraman and Chizek [1993], Nam [1989], Grizzle [1986], Nijmeijer and van der Schaft [1990], Aranda-Bricaire et al. [1996], Jakubczyk [1987], Simões and Nijmeijer [1996]. Except Simões and Nijmeijer [1996], all papers focus on smooth feedback.

The interest in recasting these old problems is that the new solution is not based on the 'tangent linearized system' description of the system but is found directly by manipulating the functions on the system equation level. Therefore, for finding the solution one is not required to solve a partial differential equation or to integrate the differential one-forms. The new approach is based on the algebra of functions. Then we compare the new results with the one described in terms of the differential forms in Aranda-Bricaire et al. [1996].

[⋆] This work was supported by the Estonian governmental funding project no. SF0140018s08, ESF grant no. 8365 and Russian Foundation of Basic Researchers Grants 10-08-00133 and 10-08-91220-CT.

2. TOOLS BASED ON DIFFERENTIAL FORMS

Consider a discrete-time nonlinear control system Σ of the form

$$\sigma(x) = f(x, u), \tag{1}$$

where by $\sigma(x)$ is denoted the forward shift of x, alternatively written as x^+, $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, the variables $x = [x_1, ..., x_n]^T$ [1] and u are the coordinates of the state space $\mathbb{R}^n$ and the input space $\mathbb{R}$, respectively. In the approach based of differential one-forms, one assumes that f in (1) is analytic function. However, in the approach based on the algebra of functions there is no need to assume that f is analytic. Actually, f is allowed even to be nonsmooth.

In the study of discrete-time nonlinear control systems the following assumption is usually made, that guarantees the forward shift operator, defined by equation (1), to be injective. Note that this assumption is not restrictive since it is always satisfied for accessible systems, see Grizzle [1993].

Assumption 1. The system (1) is submersive, i. e. generically, $\text{rank}[\partial f(x, u)/\partial(x, u)] = n$.

Note that this assumption is not restrictive, especially for problems studied in this paper, since by the results of Grizzle [1993], submersivity is a necessary condition for a system to be accessible. Moreover, accessibility is a necessary condition for static state feedback linearizability.

In the approach of differential one-forms one associates with system (1) an inversive difference field $(\mathcal{K}, \sigma)$ of meromorphic functions in a finite number of independent system variables, see Aranda-Bricaire et al. [1996]. The forward shift operator $\sigma : \mathcal{K} \to \mathcal{K}$ is defined by

$$\sigma\varphi(x, u) = \varphi(\sigma(x), \sigma(u)) = \varphi(f(x, u), \sigma(u)).$$

However, not every element in $\mathcal{K}$ has necessarily a preimage with respect to σ. To guarantee that σ is an automorphism, one has to extend equations (1) by

$$\tilde{x} = g(x, u) \tag{2}$$

such that

$$\text{rank}\frac{\partial(f^T, g^T)^T}{\partial(x, u)} = n + 1.$$

Though the choice of the function $g(x, u)$ is not unique, all choices lead to isomorphic differential fields. In what follows we use sometimes the abridged notation $\varphi^+ = \sigma(\varphi)$ and $\varphi^- = \sigma^{-1}(\varphi)$ for $\varphi \in \mathcal{K}$.

Over the field $\mathcal{K}$ one can define a vector space $\mathcal{E} := \text{span}_{\mathcal{K}}\{d\varphi \mid \varphi \in \mathcal{C}\}$ spanned by the differentials of the elements of $\mathcal{C} = \{x, \sigma^k(u), k \geq 0\}$. The elements of $\mathcal{E}$ are called differential one-forms. The forward shift operator $\sigma : \mathcal{K} \to \mathcal{K}$ induces a forward shift operator $\sigma : \mathcal{E} \to \mathcal{E}$ by

$$\Sigma_i a_i d\varphi_i \to \Sigma_i a_i^+ d(\sigma(\varphi_i)).$$

A 1-form $\omega \in \mathcal{E}$ is called exact if $d\omega = 0$ and closed if $d\omega \wedge \omega = 0$, where $\wedge$ denotes the wedge product. The subspace of 1-forms in $\mathcal{E}$ is called completely integrable if it admits a basis which consists only of closed one-forms. The relative degree r of a 1-form ω in $\mathcal{X} := \text{span}_{\mathcal{K}}\{dx\}$ is defined by $r = \min\{k \in \mathbb{N} \mid \omega(k) \notin \mathcal{X}\}$.

Define a sequence of codistributions $\mathcal{H}_k$ as follows

$$\begin{aligned} \mathcal{H}_1 &= \text{span}_{\mathcal{K}}\{dx\} \\ \mathcal{H}_{k+1} &= \{\omega \in \mathcal{H}_k \mid \omega^+ \in \mathcal{H}_k\}, \quad k \geq 1. \end{aligned}$$

Each $\mathcal{H}_k$ contains the one-forms with relative degree equal to k or greater than k. The sequence $\mathcal{H}_k$ is nonincreasing. There exists an integer $k^* \leq n$ such that for $1 \leq k \leq k^*$, $\mathcal{H}_{k+1} \subset \mathcal{H}_k$, $\mathcal{H}_{k^*+1} \neq \mathcal{H}_{k^*}$ but $\mathcal{H}_{k^*+1} = \mathcal{H}_{k^*+2} := \mathcal{H}_\infty$. Obviously, k^* is the minimal integer satisfying $\mathcal{H}_{k^*+1} = \mathcal{H}_{k^*+2}$ and $\mathcal{H}_\infty$ is the maximal codistribution, invariant with respect to the forward shift. Finally, note that the subspaces are invariant with respect to the regular static state feedback and state coordinate transformation Aranda-Bricaire et al. [1996].

3. THE ALGEBRA OF FUNCTIONS

To readdress accessibility and feedback linearization problems, the mathematical technique called the algebra of functions and developed in Zhirabok and Shumsky [2008] will be used. We recall below briefly the definitions and concepts to be used in this paper, see also Shumsky [2009]. Since these tools are not widely known, we provide many illustrative examples to illustrate the definitions.

The elements of algebra of functions are vector functions and its main ingredients are:

(1) relation of partial preorder, denoted by $\leq$,
(2) binary operations, denoted by $\times$ and $\oplus$,
(3) binary relation, denoted by Δ,
(4) operators $\mathbf{m}$ and $\mathbf{M}$.

The first two elements are defined on the arbitrary set S of vector functions whereas the last two are defined for the set S_X of vector functions with the domain being the state space X.

Definition 1. (Relation of partial preorder) Given $\alpha, \beta \in S$, one says that $\alpha \leq \beta$ iff there exists a function γ such that

$$\beta(s) = \gamma(\alpha(s))$$

for $\forall s \in S$.

The definition means that every component of the function β can be expressed as a function of α. Clearly, $\alpha \leq \beta$ iff

$$\text{rank}[\partial\alpha/\partial s] = \text{rank}\begin{bmatrix} \partial\alpha/\partial s \\ \partial\beta/\partial s \end{bmatrix}.$$

Example 2. Let $\alpha(s) = [s_1, s_2]^T$, $\beta(s) = [s_1, s_1 s_2]^T$. Then $\alpha \leq \beta$ since there exists $\gamma(\alpha) = [\alpha_1, \alpha_1\alpha_2]^T$ such that $\beta_1 = \alpha_1$, $\beta_2 = \alpha_1\alpha_2$. The inequality $\beta \leq \alpha$ does not hold in general, since $\alpha_2 = \beta_2/\beta_1$ is not valid for $s_1 = \beta_1(s) = 0$, i.e. on the set of measure zero.

If $\alpha \nleq \beta$ and $\beta \nleq \alpha$, then α and β are said to be incomparable.

Example 3. Let $\alpha(s) = [s_1 s_3, s_2]^T$ and $\beta(s) = [s_1, s_2 s_3]^T$; $\alpha(s)$ and $\beta(s)$ are incomparable. Note that $\alpha(s) \nleq \beta(s)$, since

[1] Note that we often omit the symbol of transposition T in $[\ ,\ ]^T$ for simplicity of presentation.

$$\operatorname{rank}\left(\frac{\partial\alpha}{\partial s}\right) = \begin{pmatrix} s_3 & 0 & s_1 \\ 0 & 1 & 0 \end{pmatrix} \neq \operatorname{rank}\begin{pmatrix} \partial\alpha/\partial s \\ \partial\beta/\partial s \end{pmatrix}$$

$$= \begin{pmatrix} s_3 & 0 & s_1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & s_3 & s_2 \end{pmatrix}.$$

In the similar manner one can show that $\beta(s) \leq \alpha(s)$.

Definition 4. (Strict equivalence) If $\alpha \leq \beta$ and $\beta \leq \alpha$, then α and β are called strictly equivalent, denoted by $\alpha \cong \beta$.

Note that the relation $\cong$ is reflexive, symmetric and transitive. The equivalence relation divides the set S into the *equivalence classes* containing the equivalent functions.

Example 5. The functions $\alpha(s) = [s_1, s_2]^T$ and $\beta(s) = [s_1, s_1 + s_2]^T$ are strictly equivalent since $\beta_1 = \alpha_1$, $\beta_2 = \alpha_1 + \alpha_2$, and $\alpha_1 = \beta_1$, $\alpha_2 = \beta_2 - \beta_1$.

Besides the strict equivalence, we use the notion of equivalence, corresponding to the situation when one of the inequalities $\alpha \leq \beta$ or $\beta \leq \alpha$ may be violated on a set of measure zero.

Example 6. (Continuation of Example 2) The functions α and β are equivalent though not strictly equivalent.

Definition 7. Given $\alpha, \beta \in S$,

$$\alpha \times \beta = \max(\gamma \in S \mid \gamma \leq \alpha, \gamma \leq \beta),$$

and

$$\alpha \oplus \beta = \min(\gamma \in S \mid \alpha \leq \gamma, \beta \leq \gamma).$$

It follows from these definitions that the function $\alpha \times \beta$ is a maximal bottom of the functions α and β while $\alpha \oplus \beta$ is their minimal top. In the simple cases the definition may be used to compute $\alpha \oplus \beta$. For the general case, see Zhirabok and Shumsky [2008].

The rule for operation $\times$ is simple

$$(\alpha \times \beta)(s) = \begin{bmatrix} \alpha(s) \\ \beta(s) \end{bmatrix}.$$

However, the product may contain redundant (functionally dependent) components that have to be found and removed. Moreover, to simplify the computations, one is advised to replace the remaining components by equivalent but more simple functions. At moment, no algorithm exists for these two steps.

Example 8. (Computation of the functions $\alpha \times \beta$ and $\alpha \oplus \beta$). Let $S = \mathbb{R}^3$,

$$\alpha(s) = \begin{bmatrix} s_1 + s_2 \\ s_3 \end{bmatrix}, \quad \beta(s) = \begin{bmatrix} s_1 s_3 \\ s_2 s_3 \end{bmatrix}.$$

Then $(\alpha \times \beta)(s) \cong [s_1 + s_2, s_3, s_1 s_3]^T$, $(\alpha \oplus \beta)(s) \cong s_3(s_1 + s_2)$.

Definition 9. (Binary relation Δ) Given $\alpha, \beta \in S_X$, α and β are said to form an (ordered) pair, denoted as $(\alpha, \beta) \in \Delta$ if there exists a function f_* such that

$$\beta(f(x, u)) = f_*(\alpha(x), u) \tag{3}$$

for all $(x, u) \in X \times U$.

The example below shows that the binary relation is not symmetric.

Example 10. Let $\alpha(x) = x_2$, $\beta(x) = x_1$, and the state transition map in (1)

$$f(x, u) = \begin{bmatrix} \varphi_1(x_2, u) \\ \varphi_2(x_1, x_2, u) \end{bmatrix}.$$

Then

$$\beta(f(x, u)) = \varphi_1(\alpha(x), u)$$

but

$$\alpha(f(x, u)) = \varphi(x_1, x_2, u) \neq f_*(\beta(x), u).$$

The binary relation Δ may be given the following interpretation. From (3), if for states $\tilde{x}(t)$ and $\hat{x}(t)$ at time instant t the equality

$$\alpha(\tilde{x}(t)) = \alpha(\hat{x}(t))$$

holds, then at time instant $t + 1$ we have

$$\beta(\tilde{x}(t + 1)) = \beta(\hat{x}(t + 1))$$

independent of the control $u(t)$ applied.

Another interpretation is also possible. One may ask the question. What do we have to know about $x(t)$ to compute $\beta(x(t + 1))$ for arbitrary but known $u(t)$? Of course, in the case when all the components of $x(t)$ are known, this is possible. However, in many cases, some of this information is unnecessary and the amount of the necessary information is displayed in function $\alpha(x)$, forming a pair with the function $\beta(x)$.

Obviously, given $\beta(x)$, there exist many functions $\alpha(x)$, forming a pair with $\beta(x)$, i.e. $(\alpha, \beta) \in \Delta$. The most important among them is the maximal function with respect to the relation $\leq$, denoted by $\mathbf{M}(\beta)$. In a similar manner, for a given function $\alpha(x)$, there exist many functions $\beta(x)$, forming a pair with $\alpha(x)$, i.e. $(\alpha, \beta) \in \Delta$. We will denote by $\mathbf{m}(\alpha)$ the minimal function among those functions (with respect to relation $\leq$).

Binary relation Δ is used for definition of the operators $\mathbf{m}$ and $\mathbf{M}$. These operators define the functions $\mathbf{m}(\alpha)$ and $\mathbf{M}(\beta)$ respectively that are supposed to satisfy the conditions formulated below in Definitions 11 and 12.

Definition 11. The function $\mathbf{M}(\beta) \in S_X$ is defined by the following two conditions

(i) $(\mathbf{M}(\beta), \beta) \in \Delta$
(ii) if $(\alpha, \beta) \in \Delta$, then $\alpha \leq \mathbf{M}(\beta)$.

Definition 12. The function $\mathbf{m}(\alpha) \in S_X$ is defined by the following two conditions

(i) $(\alpha, \mathbf{m}(\alpha)) \in \Delta$
(ii) if $(\alpha, \beta) \in \Delta$, then $\mathbf{m}(\alpha) \leq \beta$.

The properties of the operators $\mathbf{M}$ and $\mathbf{m}$ are as follows (see Zhirabok and Shumsky [2008]):

(1) $\alpha \leq \beta \Rightarrow \mathbf{M}(\alpha) \leq \mathbf{M}(\beta)$;
(2) $\alpha \leq \beta \Rightarrow \mathbf{m}(\alpha) \leq \mathbf{m}(\beta)$;
(3) $\mathbf{m}(\alpha \oplus \beta) \cong \mathbf{m}(\alpha) \oplus \mathbf{m}(\beta)$;
(4) $\mathbf{m}(\mathbf{M}(\beta)) \leq \beta$;
(5) $\mathbf{M}(\mathbf{m}(\alpha)) \geq \alpha$;

(6) α is f-invariant function $\Leftrightarrow \mathbf{m}(\alpha) \le \alpha \Leftrightarrow \alpha \le \mathbf{M}(\alpha)$

Computation of the operator $\mathbf{m}$. It has proven that the function γ exists that satisfies the condition $(\alpha \times u) \oplus f \cong \gamma(f)$; if f is surjection[2] define $\mathbf{m}(\alpha) \cong \gamma$, see Shumsky [1988]. In this paper we assume that f is surjection.

Note that the latter assumption and the submersivity assumption made in Section 2 are related as follows. The map f is a submersion at a point (x, u) if its differential $\mathrm{d}f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ at this point is a surjective linear map.

Because the composition $\gamma(f)$ may be written as γ^+ and $\mathbf{m}(\alpha) \cong \gamma$, one may alternatively write the rule for computation of the operator $\mathbf{m}$ using a backward shift as follows: $\mathbf{m}(\alpha) \cong ((\alpha \times u) \oplus f)^-$.

Computation of the operator $\mathbf{M}$. In the special case when the composite function $\beta(f(x, u))$ can be represented in the form

$$\beta(f(x,u)) = \sum_{i=1}^{d} a_i(x) b_i(u)$$

where $a_1(x)$, $a_2(x)$, ..., $a_d(x)$ are arbitrary functions and $b_1(u)$, $b_2(u)$, ..., $b_d(u)$ are linearly independent, then

$$\mathbf{M}(\beta) := a_1 \times a_2 \times \cdots \times a_d.$$

For the general case, see Zhirabok and Shumsky [2008].

4. MINIMAL f-INVARIANT FUNCTION

The goal of this section is to find a minimal (containing the maximal number of functionally independent components) vector function $\alpha^0(x)$ such that its forward shift $\alpha^0(f(x, u))$ does not depend on control u. Note that if f is smooth, $\alpha^0(x)$ satisfies the condition

$$\frac{\partial}{\partial u} \alpha^0(f(x,u)) \equiv 0.$$

Though $\alpha^0(x)$ is not unique, all possible choices are equivalent functions.

Note that since the relative degrees of the components of α^0 are two or more, the differentials of $\alpha^0(x)$ span the integrable part of the codistribution $\mathcal{H}_2$ of the one-forms, denoted by $\hat{\mathcal{H}}_2$, i.e. $\hat{\mathcal{H}}_2 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\alpha^0(x)\}$.

Algorithm 1. (Computation of the minimal f-invariant function α satisfying the condition $\alpha^0 \le \alpha$). Given α^0, compute recursively, using the formula below

$$\alpha^{i+1} = \alpha^i \oplus \mathbf{m}(\alpha^i) \tag{4}$$

the sequence of nondecreasing functions α^i, $i \ge 1$. By Theorem 1 in Shumsky and Zhirabok [2010c] there exists a finite k such that α^{k+1} is equivalent to α^k, denoted by $\alpha^{k+1} \cong \alpha^k$. Note that the sequence α^k converges at most in n steps. Define $\alpha_* := \alpha^k$, and $\delta^i := \alpha^{i-1}$, for $i = 1, \ldots, n$.

The proposition below demonstrates that δ^k corresponds to the integrable subspace of $\mathcal{H}_{k+1}$, denoted by $\hat{\mathcal{H}}_{k+1}$.

Proposition 13. $\hat{\mathcal{H}}_{k+1} = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\delta^k(x)\}$.

Proof. We give the proof for δ^2 and $\mathcal{H}_3$. The justification for the following steps is completely similar. As shown above $\hat{\mathcal{H}}_2 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\alpha^0(x)\} = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\delta^1(x)\}$. Next, note that $\mathcal{H}_3$ may be alternatively defined as $\mathcal{H}_3 = \mathcal{H}_2 \cap \mathcal{H}_2^-$[3]. Note that the integrable subspace of $\mathcal{H}_3$, denoted by $\hat{\mathcal{H}}_3$ may be computed alternatively as $\widehat{\hat{\mathcal{H}}_2 \cap \hat{\mathcal{H}}_2^-}$. Indeed, if the exact one-form $\mathrm{d}\zeta \in \hat{\mathcal{H}}_3$, one necessarily has $\mathrm{d}\zeta \in \mathcal{H}_2$, $\mathrm{d}\zeta \in \mathcal{H}_2^-$, and so, also into their intersection. Since $\mathrm{d}\zeta$ is exact, $\mathrm{d}\zeta \in \hat{\mathcal{H}}_2$ and $\mathrm{d}\zeta \in \hat{\mathcal{H}}_2^-$, and therefore, also into their intersection as well as into the integrable subspace of the intersection. To show the converse, note that $\mathcal{H}_3 = \mathcal{H}_2 \cap \mathcal{H}_2^-$ necessarily yields $\hat{\mathcal{H}}_2 \cap \hat{\mathcal{H}}_2^- \subset \mathcal{H}_3$. Next, though $\mathrm{span}_{\mathcal{K}}\{\mathrm{d}\mathbf{m}(\delta^1)\} \ne \hat{\mathcal{H}}_2^-$ completely, these two distributions differ by a single basis element. Since this basis element is missing in $\hat{\mathcal{H}}_2$, it does not affect the intersection. In particular, by definition of the operator $\mathbf{m}$, $\delta^1 \times u \le \mathbf{m}(\delta^1) \circ f = \mathbf{m}(\delta^1)^+$, and therefore $(\delta^1)^- \times u^- \le \mathbf{m}(\delta^1)$. By definition, the function $(\delta^1)^-$ contains the variable $\tilde{x}$ whereas $\mathbf{m}(\delta^1)$ is free from this variable. Therefore, because $\mathbf{m}(\delta^1)$ is the minimal function, satisfying this inequality,

$$\begin{aligned}\mathrm{span}_{\mathcal{K}}\{\mathrm{d}\mathbf{m}(\delta^1)\} &= \mathrm{span}_{\mathcal{K}}\{\mathrm{d}(\delta^1)^-\} - \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\tilde{x}\} \\ &\quad + \mathrm{span}_{\mathcal{K}}\{\mathrm{d}u^-\} \\ &= \hat{\mathcal{H}}_2^- - \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\tilde{x}\} + \mathrm{span}_{\mathcal{K}}\{\mathrm{d}u^-\}.\end{aligned}$$

Then

$$\begin{aligned}\mathrm{span}_{\mathcal{K}}\{\mathrm{d}(\delta^2)\} &= \mathrm{span}_{\mathcal{K}}\{\mathrm{d}(\delta^1 \oplus \mathbf{m}(\delta^1))\} \\ &= \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\delta^1\} \cap \mathrm{span}_{\mathcal{K}}\{\mathrm{d}\mathbf{m}(\delta^1)\} \\ &= \hat{\mathcal{H}}_2 \cap (\hat{\mathcal{H}}_2^- + \mathrm{span}_{\mathcal{K}}\{\mathrm{d}u^-\})\end{aligned}$$

that corresponds to $\hat{\mathcal{H}}_3 = \hat{\mathcal{H}}_2 \cap \hat{\mathcal{H}}_2^-$. ■

Examples 14 and 15 below illustrate the Proposition 13. Namely, note that in the span of $\hat{\mathcal{H}}_2^-$ the differential of the control variable $\mathrm{d}u^-$ is missing but $\hat{\mathcal{H}}_2^-$ contains instead the element of $\mathrm{d}\tilde{x}^-$ whereas $\tilde{x}^-$ is missing in $\mathbf{m}(\delta^1)$. Moreover, $\delta^1 \oplus \mathbf{m}(\delta^1)$ corresponds to $\hat{\mathcal{H}}_2 \cap \hat{\mathcal{H}}_2^-$.

Example 14. Consider the system

$$\begin{aligned}x_1^+ &= x_1 + x_3 \\ x_2^+ &= x_2 + x_5 \\ x_3^+ &= u \\ x_4^+ &= x_3 x_4 \\ x_5^+ &= x_1\end{aligned}$$

Note that for this example (if we take $\tilde{x} = x_5$)

$$\begin{aligned}x_1^- &= x_5 \\ x_2^- &= x_2 - \tilde{x}^- \\ x_3^- &= x_1 - x_5 \\ x_4^- &= x_4/(x_1 - x_5) \\ x_5^- &= \tilde{x}^-\end{aligned}$$

and $u^- = x_3$.

Compute

$$\delta^1 = [x_1, x_2, x_4, x_5]^T$$

and

$$\begin{aligned}\mathbf{m}(\delta^1) &= [[x_1, x_2, x_4, x_5, u] \oplus [x_1 + x_3, x_2 + x_5, u, x_3 x_4, x_1]]^- \\ &= [x_2 + x_5, u, x_1, x_4]^- = [x_2, x_3, x_5, \frac{x_4}{x_1 - x_5}]\end{aligned}$$

[2] For non-surjective f the formula is more complicated.

[3] Note that the application of the backward shift to the codistribution has to be understood componentwise.

Note that δ^1 corresponds to $\mathcal{H}_2 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_1, \mathrm{d}x_2, \mathrm{d}x_4, \mathrm{d}x_5\}$, which is integrable and $\mathbf{m}(\delta^1)$ corresponds to the $\mathcal{H}_2^-$, where

$$\begin{aligned} \mathcal{H}_2^- &= \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_1^-, \mathrm{d}x_2^-, \mathrm{d}x_4^-, \mathrm{d}x_5^-\} \\ &= \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_1^-, \mathrm{d}x_2^- + \mathrm{d}x_5^-, \mathrm{d}x_4^-, \mathrm{d}x_5^-\} \\ &= \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_5, \mathrm{d}x_2, \mathrm{d}\left(\frac{x_4}{x_1 - x_5}\right), \mathrm{d}\tilde{x}^-\} \end{aligned}.$$

The only difference between $\mathbf{m}(\delta^1)$ and $\mathcal{H}_2^-$ is that whereas $\mathbf{m}(\delta^1)$ contains $x_3 = u^-$, $\mathcal{H}_2^-$ contains $\mathrm{d}\tilde{x}^- = \mathrm{d}x_5$. All the other components coincide.

Furthermore, compute

$$\delta^2 = \delta^1 \oplus \mathbf{m}(\delta^1) = \left[x_2, x_5, \frac{x_4}{x_1 - x_5}\right],$$

and

$$\begin{aligned} \mathbf{m}(\delta^2) &= \Bigg[\left[x_2, x_5, \frac{x_4}{x_1 - x_5}, u\right] \\ &\quad \oplus [x_1 + x_3, x_2 + x_5, u, x_3x_4, x_1]\Bigg]^- \\ &= [x_2 + x_5, u]^- = [x_2, x_3]. \end{aligned}$$

Note that δ^2 corresponds to (integrable)

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}}\left\{\mathrm{d}x_2, \mathrm{d}x_5, \mathrm{d}\left(\frac{x_4}{x_1 - x_5}\right)\right\}$$

In a similar manner, compute

$$\delta^3 = \delta^2 \oplus \mathbf{m}(\delta^2) = x_2$$

and

$$\begin{aligned} \mathbf{m}(\delta^3) &= [[x_2, u] \oplus [x_1 + x_3, x_2 + x_5, u, x_3x_4, x_1]]^- \\ &= u^- = x_3 \end{aligned}$$

Note that δ^3 corresponds to the integrable subspace of $\mathcal{H}_4$.

Therefore

$$\delta^4 = \delta^3 \oplus \mathbf{m}(\delta_3) \cong \text{const}.$$

Finally, note that $\delta^4 \cong$ const corresponds to $\mathcal{H}_\infty$ being trivial, $\mathcal{H}_\infty = \{0\}$.

Example 15. Consider the system

$$\begin{aligned} x_1^+ &= x_1 + x_3 \\ x_2^+ &= x_2 \\ x_3^+ &= u \\ x_4^+ &= x_3x_4 \\ x_5^+ &= x_1 \end{aligned}$$

Compute

$$\delta^1 = [x_1, x_2, x_4, x_5]^T$$

and

$$\begin{aligned} \mathbf{m}(\delta^1) &= [[x_1, x_2, x_4, x_5, u] \oplus [x_1 + x_3, x_2, u, x_3x_4, x_1]]^- \\ &= [x_2 + x_5, u, x_1, x_4]^- = [x_2, x_3, x_5, \frac{x_4}{x_1 - x_5}] \end{aligned}$$

Note that δ^1 corresponds to $\mathcal{H}_2 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_1, \mathrm{d}x_2, \mathrm{d}x_4, \mathrm{d}x_5\}$. Furthermore,

$$\delta^2 = \delta^1 \oplus \mathbf{m}(\delta^1) = \left[x_2, x_5, \frac{x_4}{x_1 - x_5}\right],$$

and

$$\begin{aligned} \mathbf{m}(\delta^2) &= \left[\left[x_2, x_5, \frac{x_4}{x_1 - x_5}, u\right] \oplus [x_1 + x_3, x_2, u, x_3x_4, x_1]\right]^- \\ &= [x_2, u]^- = [x_2, x_3]. \end{aligned}$$

Note that δ^2 corresponds to

$$\mathcal{H}_3 = \mathrm{span}_{\mathcal{K}}\left\{\mathrm{d}x_2, \mathrm{d}x_5, \mathrm{d}\left[\frac{x_4}{x_1 - x_5}\right]\right\}.$$

In a similar manner, compute

$$\delta^3 = \delta^2 \oplus \mathbf{m}(\delta^2) = x_2$$

and

$$\begin{aligned} \mathbf{m}(\delta^3) &= [[x_2, u] \oplus [x_1 + x_3, x_2, u, x_3x_4, x_1]]^- \\ &= u^- = [x_2, u]^- = [x_2, x_3]. \end{aligned}$$

Note that δ^3 corresponds to the integrable subspace of $\mathcal{H}_4$, i.e. $\hat{\mathcal{H}}_4 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_2\}$.

Therefore

$$\delta^4 = \delta^3 \oplus \mathbf{m}(\delta_3) = x_2 = \delta^3$$

Finally, note that $\delta^4 \cong \delta^3 = x_2$ corresponds to the fact that $\mathcal{H}_\infty = \mathcal{H}_5 = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_2\}$.

5. ACCESSIBILITY

Note that accessibility is a necessary condition for static state feedback linearizability. Therefore, we recall below the accessibility definition and condition formulated in terms of codistribution.

Following the notation in Jakubczyk and Sontag [1990] we denote by $A_k(x)$ the set of points reachable from x in k forward time steps using arbitrary sequences of controls $\mathbf{u} = (u(0), \ldots, u(k-1)) \in (R^m)^k$, and by $A(x)$ the set of points reachable from x in any number of forward steps using arbitrary sequences of controls. That is,

$$A(x) = \bigcup_{k \geq 0} A_k(x).$$

The system is said to be forward accessible from x if its reachable set $A(x)$ has non-empty interior. A generic notion of accessibility has been derived from this pointwise definition in Albertini and Sontag [1993].

Definition 16. The system (1) is said to be (forward) accessible if its reachable set $A(x)$ has a non-empty interior in $\mathbb{R}^n$ for almost all $x \in \mathbb{R}^n$.

Proposition 17. Aranda-Bricaire et al. [1996] The system (1) is accessible iff $\mathcal{H}_\infty = \{0\}$.

Proposition 18. The following statements for system (1) are equivalent

(i) $\mathcal{H}_\infty \neq \{0\}$

(ii) For some k, $\delta^{k-1} \cong \delta^k \neq \text{const}$.

Proof. Suppose $\mathcal{H}_\infty \neq 0$. This means that f-invariant function β_* exists such that $\alpha^0 \leq \beta_*$. Due to the 6th property of the operators $\mathbf{m}$ and $\mathbf{M}$, $\mathbf{m}(\beta_*) \leq \beta_*$. The inequality $\alpha^0 \leq \beta_*$ implies $\mathbf{m}(\alpha^0) \leq \mathbf{m}(\beta_*)$ which together with the above inequality gives $\mathbf{m}(\alpha^0) \leq \beta_*$. By analogy one may prove that for all i, $\mathbf{m}^i(\alpha^0) \leq \beta_*$. It follows from (4) that $\delta^i \leq \beta_*$ for all i. Due to (4) $\delta^1 \leq \delta^2 \leq \ldots$ and since this sequence is bounded, then $\delta^{k-1} \cong \delta^k \neq \text{const}$ for some k.

Suppose that $\delta^{k-1} \cong \delta^k \neq \text{const}$ for some k. This means that the function $\alpha_* \cong \delta^k$ is f-invariant such that $\alpha^0 \leq \alpha_*$, i.e. the system (1) may be decomposed into the form containing an autonomous subsystem. The latter means that that the system (1) is nonaccessible since it admits an autonomous variable, see Aranda-Bricaire et al. [1996] and therefore $\mathcal{H}_\infty \neq \{0\}$. ■

6. FEEDBACK LINEARIZATION

Definition 19. A regular static state feedback $u = \alpha(x, v)$ is a mapping $\alpha : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ such that generically $[\partial\alpha(\cdot)/\partial v] \neq 0$.

System (1) is said to be static state feedback linearizable if, generically, there exist

(i) a state coordinate transformation $\varphi : \mathbb{R}^n \to \mathbb{R}^n$ and
(ii) a regular static state feedback of the form $u = \alpha(x, v)$, such that, in the new coordinates $z = \varphi(x)$, the compensated system reads $z^+ = Az + Bv$, where the pair (A, B) is in Brunovsky canonical form.

Theorem 20. Aranda-Bricaire et al. [1996] System (1) is static state feedback linearizable if and only if

(i) $\mathcal{H}_k$ is completely integrable for all $k = 1, \ldots, n$,
(ii) $\mathcal{H}_\infty := \mathcal{H}_{n+1} = \{0\}$.

Theorem 21 below suggests an alternative solution to the static state feedback linearization problem. Consider a special form of the system (1), the so-called controller canonical form, see Kotta [2005]:

$$\begin{aligned} z_1^+ &= z_2, \\ z_2^+ &= z_3, \\ &\vdots \\ z_{n-1}^+ &= z_n \\ z_n^+ &= \psi(z, u). \end{aligned} \tag{5}$$

The goal of Theorem 21 below is to find out under which conditions formulated in terms of the algebra of functions the system (1) can be transformed into the form (5) using the state coordinate transformation $z = \varphi(x)$. From this form a regular static state feedback may be easily found by defining $v = \psi(z, u)$, in order to solve the feedback linearization problem.

Note that by the results of Kotta [2005] the conditions to transform the system (1) into the form (5) coincide with those of Theorem 20.

Theorem 21. The system (1) can be transformed into the form (5) iff $\delta^i \neq \text{const}$, for $i = 1, \ldots, n-1$, but $\delta^n = \text{const}$ where $\delta^i = \alpha^{i-1}$ for $i = 1, 2, \ldots, n$.

Proof. Sufficiency. Define $\varphi_1 := \delta^{n-1}$, $\varphi_{i+1} = \mathbf{M}(\varphi_i)$, $i = 1, 2, \ldots, n-1$ and $z_i = \varphi_i(x)$, $i = 1, 2, \ldots, n$. According to the definition of the operator $\mathbf{M}$, one has $(\mathbf{M}(\varphi_i), \varphi_i) \in \Delta$ that together with $\varphi_{i+1} = \mathbf{M}(\varphi_i)$ gives $(\varphi_{i+1}, \varphi_i) \in \Delta$. By definition of the binary relation Δ, there exists a function f_{*i} such that

$$\varphi_i(f(x, u)) = f_{*i}(\varphi_{i+1}(x), u),$$

for all $(x, u) \in X \times U$ and $i = 1, 2, \ldots, n-1$. It follows from (4) that $\alpha^0 = \delta^1 \leq \delta^2 \leq \ldots \leq \delta^{n-1}$. Since $\delta^{n-1} = \delta^{n-2} \oplus \mathbf{m}(\delta^{n-2})$, we have $\delta^{n-1} \geq \mathbf{m}(\delta^{n-2})$, and therefore, by the 1st and 5th properties of the operators $\mathbf{m}$ and $\mathbf{M}$ the following holds

$$\begin{aligned} \varphi_2 := \mathbf{M}(\varphi_1) = \mathbf{M}(\delta^{n-1}) &\geq \mathbf{M}(\mathbf{m}(\delta^{n-2})) \\ &\geq \delta^{n-2} \geq \alpha^0. \end{aligned}$$

In a similar manner, one may obtain the inequalities

$$\begin{aligned} \varphi_3 &= \mathbf{M}(\varphi_2) \geq \delta^{n-3} \geq \alpha^0, \\ &\vdots \\ \varphi_{n-1} &= \mathbf{M}(\varphi_{n-2}) \geq \delta^1 \geq \alpha^0. \end{aligned}$$

The definition of the function α^0 and the inequalities $\varphi_i \geq \alpha^0$, $i = 1, 2, \ldots, n-1$, yield

$$\frac{\partial}{\partial u}\varphi_i(f(x, u)) = \frac{\partial}{\partial u} f_{*i}(\varphi_{i+1}(x), u) = 0,$$

therefore the function f_{*i} does not depend on u and

$$\varphi_i(f(x, u)) = f_{*i}(\varphi_{i+1}(x)).$$

The last equality and the rule for computation of the operator $\mathbf{M}$ yield $\mathbf{M}(\varphi_i) = f_{*i}(\varphi_{i+1})$. Since $\varphi_{i+1} = \mathbf{M}(\varphi_i)$, one may take $f_{*i}(\varphi_{i+1}) = \varphi_{i+1}$, $i = 1, 2, \ldots, n-1$. Then

$$z_i^+ = \varphi_i(f(x, u)) = \varphi_{i+1}(x) = z_{i+1}$$

for $i = 1, 2, \ldots, n-1$. Since the function φ_n does not satisfy the condition $\varphi_n \geq \alpha^0$, the equation for the variable z_n has the general form:

$$z_n^+ = \varphi_n(f(x, u)) = \psi(z, u).$$

Necessity. Suppose the system (1) can be transformed into the canonical form (5), i.e. there exists the state transformation $\varphi : X \to Z$ such that for $i = 1, \ldots, n-1$

$$z_i^+ = \varphi_i(x^+) = \varphi_i(f(x, u)) = \varphi_{i+1}(x) = z_{i+1}, \tag{6}$$

and

$$z_n^+ = \varphi_n(x^+) = \psi(z, u). \tag{7}$$

The equality $\varphi_i(f(x, u)) = \varphi_{i+1}(x)$ in (6) and a definition of the binary relation Δ yields the inclusion $(\varphi_{i+1}, \varphi_i) \in \Delta$, for $i = 1, 2, \ldots, n-1$. It is obvious that

$$\frac{\partial}{\partial u}\varphi_i(f(x, u)) = \frac{\partial}{\partial u}\varphi_{i+1}(x) = 0,$$

therefore, by the definition of the function α^0, $\alpha^0 \leq \varphi_i$ for $i = 1, 2, \ldots, n-1$. By the 2nd property of the operator $\mathbf{m}$, this inequality implies $\mathbf{m}^j(\alpha^0) \leq \mathbf{m}^j(\varphi_i)$ for all $j \geq 1$, and $i = 1, 2, \ldots, n-1$.

Consider the inclusion $(\varphi_2, \varphi_1) \in \Delta$ which is by Definition 12 equivalent to the inequality $\mathbf{m}(\varphi_2) \leq \varphi_1$ that together with $\mathbf{m}(\alpha^0) \le \mathbf{m}(\varphi_2)$ gives $\mathbf{m}(\alpha^0) \le \varphi_1$. In the similar manner the inclusion $(\varphi_3, \varphi_2) \in \Delta$ is equivalent to the inequality $\mathbf{m}(\varphi_3) \le \varphi_2$ which implies $\mathbf{m}^2(\varphi_3) \le \mathbf{m}(\varphi_2)$. Since $\mathbf{m}^2(\alpha^0) \le \mathbf{m}^2(\varphi_3)$ and $\mathbf{m}(\varphi_2) \le \varphi_1$, one obtains from these three inequalities $\mathbf{m}^2(\alpha^0) \le \varphi_1$. Analogously, it can be proved that $\mathbf{m}^i(\alpha^0) \le \varphi_1$, $i = 1, 2, \ldots, n-2$. By definition of the operation $\oplus$, these inequalities are equivalent to the single inequality

$$\alpha^0 \oplus \mathbf{m}(\alpha^0) \oplus \ldots \oplus \mathbf{m}^i(\alpha^0) \le \varphi_1, \quad i = 1, 2, \ldots n-2. \quad (8)$$

It follows from the definition of the functions δ^i, $i = 1, 2, \ldots, n-1$, and the 3rd property of the operator $\mathbf{m}$ that

$$\begin{aligned} \delta^3 &= \delta^2 \oplus \mathbf{m}(\delta^2) \\ &= (\delta^1 \oplus \mathbf{m}(\delta^1)) \oplus \mathbf{m}(\delta^1 \oplus \mathbf{m}(\delta^1)) \\ &\cong \alpha^0 \oplus \mathbf{m}(\alpha^0) \oplus \mathbf{m}^2(\alpha^0). \end{aligned}$$

In general, it can be proved that

$$\delta^i \cong \alpha^0 \oplus \mathbf{m}(\alpha^0) \oplus \ldots \oplus \mathbf{m}^{i-1}(\alpha^0), \quad i = 1, 2, \ldots, n-1.$$

Since $\varphi_1 \neq \text{const}$, then due to (8), $\delta^i \neq \text{const}$, for $i = 1, 2, \ldots, n-1$.

Suppose now contrarily to the claim of the theorem that $\delta^n = \alpha^0 \oplus \mathbf{m}(\alpha^0) \oplus \ldots \oplus \mathbf{m}^n(\alpha^0) \neq \text{const}$. By analogy with the proof of sufficiency part of the theorem, it can be shown that $\dfrac{\partial}{\partial u}\varphi_n(f(x,u)) = \dfrac{\partial}{\partial u}\psi(z,u) = 0$, i.e. the right-hand side of the equation $z_n^+ = \varphi_n(f(x,u))$ does not depend on the control u that contradicts the equation (5). ■

Proposition 22. The following two conditions for system (1) are equivalent

(i) $\mathcal{H}_\infty = \{0\}$ and $\mathcal{H}_k$, for $k = 1, \ldots, n$ are completely integrable
(ii) $\delta^i \neq \text{const}$ for $i = 1, \ldots, n-1$ and $\delta^n = \text{const}$.

Proof. (ii) → (i) Suppose that $\delta^i \neq \text{const}$ for $i = 1, 2 \ldots, n-1$, and $\delta^n = \text{const}$. According to Theorem 21, in this case the system (1) can be transformed into the controller canonical form (5) and therefore, it is accessible. Then by Proposition 17, $\mathcal{H}_\infty = \{0\}$. Moreover, by the results of Kotta [2005], $\mathcal{H}_1, \ldots, \mathcal{H}_n$ are completely integrable.

(i) → (ii) Consider a sequence of functions $\alpha^0 := \delta^1 \le \delta^2 \le \ldots$. Since the sequence converges at most in n steps, for some k, $\delta^{k+1} = \delta^k$. If $\delta^k \neq \text{const}$, then it follows from Proposition 18 that $\mathcal{H}_\infty \neq \{0\}$ which contradicts the condition (i). Therefore for some k, $\delta^k = \text{const}$. According to Theorem 20, the system (1) is static state feedback linearizable, and therefore, due to the structure of the Brunovsky canonical form, $k = n$. ■

Remark 1. If $\delta^{k-1} \cong \delta^k \neq \text{const}$ holds for some k, then $\varphi_1 \cong \mathbf{M}(\varphi_1)$. The latter means that the function φ_1 is f-invariant and the variable $z_1 := \varphi_1(x)$ satisfies the equation $z_1^+ = z_1$, and the system (1) is not transformable into the canonical form (5).

Remark 2. If the system is not transformable into the canonical form (5) one may alternatively say that function φ_1 in Remark 1 is an autonomous variable for system (1), and therefore, the system (1) is non-accessible.

7. EXAMPLES

Example 23. (Continuation of Example 14) Since for this example $n = 5$, but already $\delta^4 \cong \text{const}$, the system admits only partial linearization.

Example 24. (Continuation of Example 15) Since $\delta^3 = \delta^4 = x_2$, the system is not accessible.

Example 25. Consider the system

$$\begin{aligned} x_1^+ &= x_1(x_3^2+1)^2 \\ x_2^+ &= x_2(x_3^2+1)^3 \\ x_3^+ &= x_3 + u \end{aligned} \quad (9)$$

Compute first according to Aranda-Bricaire et al. [1996]

$$\mathcal{H}_\infty = \mathcal{H}_3 = \text{span}_{\mathcal{K}}\{3x_2\text{d}x_1 - 2x_1\text{d}x_2\};$$

therefore the system (9) admits an autonomous variable x_1^3/x_2^2.

Next, using the tools of the algebra of functions one may compute

$$\delta^1 := \alpha^0(x) = [x_1, x_2]^T.$$

Furthermore, by (4) [4] [5],

$$\delta^2(x) = \delta^1(x) \oplus \mathbf{m}(\delta^1(x)) = [x_1, x_2]^T \oplus \frac{x_1^3}{x_2^2} = \frac{x_1^3}{x_2^2}.$$

Since

$$\mathbf{m}\left(\frac{x_1^3}{x_2^2}\right) = \frac{x_1^3}{x_2^2},$$

we get

$$\delta^3(x) \cong \delta^2(x) \neq \text{const}.$$

Example 26. Consider the system with non-smooth state transition map $f(x,u)$,

$$\begin{aligned} x_1^+ &= x_2^2 u \\ x_2^+ &= x_1 \text{sign } x_2 \\ x_3^+ &= u \end{aligned}$$

Compute

$$\delta^1 := \alpha^0 = \left[\frac{x_1}{x_3}, x_2\right]$$

and

$$\begin{aligned} \mathbf{m}(\delta^1) &= \left[[\delta^1, u] \oplus f(x,u)\right]^- \\ &= \left[\left[\frac{x_1}{x_3}, x_2, u\right] \oplus [x_2^2 u, x_1 \text{sign } x_2, u]\right]^- \\ &= [x_2^2 u, u]^- = [x_1, x_3]. \end{aligned}$$

So,

$$\delta^2 = \delta^1 \oplus \mathbf{m}(\delta^1) = \left[\frac{x_1}{x_3}, x_2\right] \oplus [x_1, x_3] = \frac{x_1}{x_3}$$

Furthermore, compute

$$\begin{aligned} \mathbf{m}(\delta^2) &= [[\delta^2, u] \oplus f(x,u)]^- \\ &= \left[\left[\frac{x_1}{x_3}, u\right] \oplus [x_2^2 u, x_1 \text{sign } x_2, u]\right]^- \\ &= u^- = x_3 \end{aligned}$$

[4] $\mathbf{m}(\zeta)$ is a function of x at the time instant $t+1$ which can be computed if ζ is known at time instant t.

[5] $\delta^2(x)$ is a function of x which can be computed both from $\delta^1(x)$ and $\mathbf{m}(\delta^1(x))$.

and

$$\delta^3 = \delta^2 \oplus \mathbf{m}(\delta^2) = \frac{x_1}{x_3} \oplus x_3 = \text{const.}$$

Therefore, define

$$\begin{aligned} z_1 &:= \delta^2 = \frac{x_1}{x_3} \\ z_2 &:= z_1^+ = \mathbf{M}(\delta^2) = x_2^2 \\ z_3 &:= z_2^+ = \mathbf{M}^2(\delta^2) = \mathbf{M}(x_2^2) = x_1^2(\text{sign } x_2)^2 \end{aligned}$$

and find the state equations

$$\begin{aligned} z_1^+ &= z_2 \\ z_2^+ &= z_3 \\ z_3^+ &= z_2^2 u^2 \left(\text{sign}\sqrt{z_3}\right)^2 \end{aligned}$$

Example 27. Consider

$$\begin{aligned} x_1^+ &= u x_1 \\ x_2^+ &= x_1 x_3 \\ x_3^+ &= x_3 \end{aligned}$$

Compute

$$\delta^1 := \alpha^0 = [x_2, x_3]^T$$

and

$$\begin{aligned} \mathbf{m}(\delta^1) &= [[x_2, x_3, u] \oplus [u x_1, x_1 x_3, x_3]]^- \\ &= x_3^- = x_3. \end{aligned}$$

Furthermore,

$$\delta^2 \cong \delta^1 \oplus \mathbf{m}(\delta^1) = x_3.$$

Obviously, $\delta^3 \cong \delta^2 \neq$ const. The system admits an autonomous element x_3. The system is neither accessible nor linearizable via static state feedback.

Example 28. Consider the system

$$\begin{aligned} x_1^+ &= \zeta(x_2) u \\ x_2^+ &= x_1 x_2 \\ x_3^+ &= u \end{aligned}$$

where ζ is an invertible analytic function, and compute

$$\mathcal{H}_2 = \text{span}_{\mathcal{K}} \left\{ \text{d}\left(\frac{x_1}{x_3}\right), \text{d}x_2 \right\}$$

and

$$\mathcal{H}_3 = \text{span}_{\mathcal{K}} \left\{ \text{d}\left(\frac{x_1}{x_3}\right) \right\}, \quad \mathcal{H}_3 = \{0\}.$$

Define the new state variables

$$\begin{aligned} z_1 &= \frac{x_3}{x_3} \\ z_2 &= \left(\frac{x_1}{x_3}\right)^+ = \zeta(x_2) \\ z_3 &= \zeta^+(x_2) = \zeta(x_2^+) = \zeta(x_1, x_2) \end{aligned} \tag{10}$$

and so the state equations in the controller canonical form are

$$\begin{aligned} z_1^+ &= z_2 \\ z_2^+ &= z_3 \\ z_3^+ &= \zeta(\zeta^{-1}(z_3) z_2 u). \end{aligned} \tag{11}$$

Note that using the results of Theorem 21, one does not have to assume $\zeta(x_2)$ to be analytic. Compute

$$\delta^1 := \alpha^0 = [x_1/x_3, x_2]$$

and

$$\begin{aligned} \mathbf{m}(\delta^1) &= [[\delta^1, u] \oplus f(x,u)]^- \\ &= [[x_1/x_3, x_2, u] \oplus [\zeta(x_2)u, x_1 x_2, u]]^- \\ &= [\zeta(x_2)u, u]^- = [x_1, x_3] \end{aligned}$$

So, $\delta^2 = \delta_1 \oplus \mathbf{m}(\delta^1) = x_1/x_3$. Then

$$\begin{aligned} \mathbf{m}(\delta^2) &= [[\delta^2, u] \oplus f(x,u)]^- \\ &= [[x_1/x_3, u] \oplus [\zeta(x_2)u, x_1 x_2, u]]^- \\ &= u^- = x_3 \end{aligned}$$

and $\delta^3 = \delta^2 \oplus \mathbf{m}(\delta^2) = \text{const.}$

Therefore, define $z_1 = \delta^2 = x_1/x_3$, $z_2 = \mathbf{M}(\delta^2) = \zeta(x_2)$, $z_3 = \mathbf{M}^2(\delta^2) = \zeta(x_1 x_2)$ that agrees with (10) yielding (11).

Example 29. Consider the system

$$\begin{aligned} x_1^+ &= x_3, \\ x_2^+ &= \text{sign}(x_3) + x_1, \\ x_3^+ &= u x_1. \end{aligned} \tag{12}$$

Note that this system cannot be studied using the results of Theorem 21 since f in (12) is not a smooth function. Compute

$$\delta^1 = [x_1, x_2],$$

and

$$\begin{aligned} \mathbf{m}(\delta^1) &= [[x_1, x_2, u] \oplus [x_3, sign(x_3) + x_1, u x_1]]^- \\ &= [x_1, u x_1]^- = [x_2 - sign(x_1), x_3]. \end{aligned}$$

Therefore,

$$\delta^2 = x_2 - sign(x_1).$$

Next, compute $\mathbf{m}(\delta^2) = \text{const}$, yielding $\delta^3 = \text{const}$. By the results of Theorem 21 (see the proof), we have

$$\begin{aligned} z_1 &:= \delta^{n-1} = \delta^2 = x_2 - sign(x_1), \\ z_2 &:= z_1^+ = x_1, \\ z_3 &:= z_2^+ = x_3, \end{aligned}$$

yielding the equations in the controller canonical form

$$\begin{aligned} z_1^+ &= z_2, \\ z_2^+ &= z_3, \\ z_3^+ &= u z_2 \end{aligned}$$

that are easily linearizable by using the feedback $u = z_2/v$.

8. CONCLUSIONS

For the discrete-time nonlinear SISO control systems the problems of accessibility and static state feedback linearizability have been readdressed in terms of the new tools, called the algebra of functions. Unlike the differential geometric methods the new tools allow to study the non-smooth systems. The new results are compared to the existing ones and the relationship is demonstrated on the numerous examples.

The extension of the results for the continuous-time case is not immediate, since the inequality $\delta^{k-1} \geq \mathbf{m}(\delta^{k-2})$ in the continuous-time case, unlike the discrete-time case, does not yield the inequality $\mathbf{M}(\delta^{k-1}) \geq \mathbf{M}(\mathbf{m}(\delta^{k-2}))$ which was used in the proof.

REFERENCES

F. Albertini and E.D. Sontag. Discrete-time transitivity and accessibility: analytic systems. *SIAM J. Contr. Optim.*, 31:1599–1622, 1993.

E. Aranda-Bricaire, Ü. Kotta, and C.H. Moog. Linearization of discrete-time systems. *SIAM J. Contr. Optim.*, 34(6):1999–2023, 1996.

G. Conte, C.H. Moog, and A.M. Perdon. *Algebraic Methods for Nonlinear Control Systems. Theory and Applications.* Springer, 2007.

J.W. Grizzle. Feedback linearization of discrete-time systems. In *Lecture Notes in Control and Information Sciences*, volume 83, pages 273–281. Springer, New York, 1986.

J.W. Grizzle. A linear algebraic framework for the analysis of discrete-time nonlinear systems. *SIAM J. Contr. Optim.*, 31:1026–1044, 1993.

J. Hartmanis and R. Stearns. *The Algebraic Structure Theory of Sequential Machines.* Prentice-Hall, New York, 1966.

B. Jakubczyk. Feedback linearization of discrete-time systems. *Systems and Control Letters*, 9:411–416, 1987.

B. Jakubczyk and E.D. Sontag. Controllability of nonlinear discrete-time systems: a lie algebraic approach. *SIAM J. Contr. Optim.*, 28:1–33, 1990.

G. Jayaraman and H.J Chizek. Feedback linearization of discrete-time systems. In *Proc. of the 32nd IEEE Conference on Decision and Control*, pages 2372–2977. San Antonion, Texas, 1993.

Ü. Kotta. Controller and controllability canonical forms for discrete-time nonlinear systems. *Proceedings of the Estonian Academy of Sciences.Physics.Mathematics*, 54 (1):55–62, 2005.

K. Nam. Linearization of discrete-time nonlinear systems and a canonical structure. *IEEE Trans. Autom. Control*, 34:119–122, 1989.

H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems.* Springer, New York, 1990.

A. Shumsky and A. Zhirabok. Method of accommodation to the defect in finite automata. *Automation and Remote Control*, 71(5):837–846, 2010a.

A. Shumsky and A. Zhirabok. Unified approach to the problem of fault diagnosis. In *Proc. Conf. on Control and Fault Tolerant Systems*, pages 450–55. Nice, France, 2010b.

A.Ye Shumsky. Model of faults for discrete systems and its application to functional diagnosis problem. *Engineering Simulation*, 12(4):56–61, 1988.

A.Ye. Shumsky. Full decoupling via feedback in nonlinear descriptor systems: algebraic approach. In *Proc. of the 6th IFAC Symposium on Robust Control Design*, pages 261–266. Haifa, Israel, 2009.

A.Ye. Shumsky and A.N. Zhirabok. Unified approach to the problem of full decoupling via output feedback. *European Journal of Control*, 16(4):313–325, 2010c.

C. Simões and H. Nijmeijer. Nonsmooth stabilizability and feedback linearization of discrete-time systems. *International Journal of Robust and Nonlinear Control*, 6: 171–188, 1996.

A.N. Zhirabok and A.Ye. Shumsky. *The algebraic methods for analysis of nonlinear dynamic systems (In Russian).* Dalnauka, Vladivostok, 2008.

DISCRETE-TIME SOLUTION TO THE DISTURBANCE DECOUPLING PROBLEM OF COUPLED TANKS

Miroslav Halás

*Institute of Control and Industrial Informatics, Fac. of Electrical
Engineering and IT, Slovak University of Technology
Ilkovičova 3, 81219 Bratislava, Slovakia
e-mail : miroslav.halas@stuba.sk*

Abstract: Mathematical technicalities, involved in the modern theory of non-linear control systems, many times prevent a wider use of the impressive theoretical results in practice. Attempts to overlap this gap between theory and practice are usually more than welcome and form the main scope of our interest in this work. An important control problem given by the disturbance decoupling is studied for a real laboratory model of coupled tanks. Since the theoretical solution to the disturbance decoupling problem does not satisfy practical control requirements it is modified accordingly. Experiments on the real plant are included as well and show that the disturbances practically do not affect the system output.

Keywords: nonlinear discrete-time systems, applications, algebraic methods, disturbance decoupling, coupled tanks

1. INTRODUCTION

The modern theory of nonlinear control systems all, continuous-time, discrete-time and time-delay, owes a large part of its succes to the systematic use of differential algebraic methods. Since early 80's of the last century this has been forming the scope of interest of many authors in a number of works, see for instance Fliess (1985 1992); Conte et al. (1993); Aranda-Bricaire et al. (1995 1996); Kotta et al. (2001); Xia et al. (2002); Conte et al. (2007) and references therein. Nowadays, such methods offer solutions to a wide range of nonlinear control problems including feedback linearization, model matching, disturbance decoupling, realization problem, non-interacting control, observer design and many others.

However, a price one has to pay for such impressive and elegant solutions is given by a necessity to involve many mathematical technicalities. Obviously, this prevents a wider use of the theoretical results in practice, making the big gap between control theory and control practice even bigger in this case. It is generally known that in practice the way of dealing with nonlinear control systems is many times based just on the linearization in a fixed operating point and then methods of linear control systems are applied. Therefore, attempts to overlap the gap are usually more than welcome and form the main scope of our interest in this work. In particular, an important control problem given by the disturbance decoupling, which is quite frequent control problem in practice, is studied. We begin with the theoretical solutions of Conte et al. (2007) and apply them to the laboratory model of coupled tanks, which is a demonstrative and well know system having contact points to many real control processes, for instance from chemical engineering. It is shown that the theoretical solutions cannot be directly applied and additional problems, related for instance to the difference between model and real system, have to be considered as well. Similar solution as discussed in this paper has recently been given in Žilka and Halás (2010) for continuous-time case, while here the discrete-time counter part is treated. Certain contact points exist also to the non-interacting problem studied in Halás and Žilka (2011). Finally, for additional existing results of the disturbance decoupling problem for nonlinear discrete-time systems the reader is referred for instance to Kotta (1995) where a simple inversion-based solution is given and to Grizzle (1985) where a more advanced differential geometric solution can be found.

2. DISTURBANCE DECOUPLING

We begin with an introduction to the disturbance decoupling problem of nonlinear control systems as discussed in Conte et al. (2007) to which the reader is referred for additional details and references. The ideas can easily be carried over to the discrete-time systems.

For the sake of simplicity we introduce the following notation. For any variable $\xi(t)$ we write only ξ and for its time shifts $\xi(t+T)$, $\xi(t+2T)$ we write ξ^+, ξ^{++} respectively, or, in general, $\xi^{[k]}$ for $\xi(t+kT)$, where T is a sampling period.

Using the above introduced notation the systems considered in this paper are objects of the form

$$\begin{aligned} x^+ &= f(x,u) \\ y &= g(x) \end{aligned} \tag{1}$$

where $x \in \mathbf{R}^n$, $u, y \in \mathbf{R}$ and entries of f and g are meromorphic functions from the difference field denoted by $\mathcal{K}$. For more details see Aranda-Bricaire et al. (1996); Kotta et al. (2001); Halás et al. (2009).

In the disturbance decoupling our task is to design, if possible, a control law such that the disturbances do not affect the system output. Technically speaking, the solution consists of finding a feedback under which a subspace of the state space, affected by disturbances, becomes unobservable in the compensated system. This situation can be explained by the following introductory system

$$\begin{aligned} x_1^+ &= x_2 u \\ x_2^+ &= w \\ y &= x_1 \end{aligned}$$

where w is the disturbance.

As can be seen, through x_2 the disturbance w affects the system output

$$y^{++} = u^+ w$$

However, the state feedback $u = v/x_2$, with v being an input to the compensated system, makes x_2 unobservable in the compensated system and thus decouples the disturbance w from the system output

$$y^+ = v$$

The general solution follows the same idea. That is, if possible, make unobservable the subspace of the state space affected by the disturbance.

Problem statement. Consider the SISO system

$$\begin{aligned} x^+ &= f(x,u) + p(x)w \\ y &= g(x) \end{aligned}$$

where the state $x \in \mathbf{R}^n$, the disturbance $w \in \mathbf{R}^q$ and the entries of f, g and p are elements of the difference field of meromorphic functions $\mathcal{K}$. Find, if possible, a static state feedback

$$u = \alpha(x,v)$$

such that

$$\mathrm{d}y^{[i]} \in \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x, \mathrm{d}v, \ldots, \mathrm{d}v^{[i]}\}$$

for any $i \in \mathbf{N}$.

Theorem 1. Let $\mathcal{X} = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x\}$ and $\mathcal{Y} = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}y^{[i]}; i \geq 0\}$. The disturbance decoupling problem is solvable if and only if $p(x) \perp \mathcal{X} \cap \mathcal{Y}$.

PROOF. The proof follows the same line as in Conte et al. (2007), however, carried over to the discrete-time case.

3. COUPLED TANKS

Coupled tanks are well-known and illustrative system having contact points to many real control processes. For that reason practically each laboratory which activities are related to the system and control theory possesses such a plant. In this section, the mathematical model of the laboratory plant is built up, from its identification to the nonlinear discrete-time state-space model. Then, the disturbance decoupling is applied.

3.1 *System identification*

We restrict our attention to a standard coupled two-tank system, however, with all three valves active. The structure of such a system is depicted in Fig. 1. Our aim is to control the level in the first tank which is, however, coupled with the second tank by a valve with the flow coefficient c_{12}. Each of the tanks is equipped by a valve itself, having the flow coefficients c_1 and c_2 respectively. However, the valve c_2 is considered here as the disturbance w. Thus, we deal here with a SISO

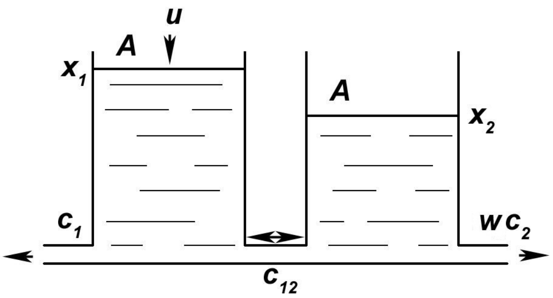


Fig. 1. Coupled tanks

system which can be modelled by the following state-space equations

$$
\begin{aligned}
\dot{x}_1 &= \frac{1}{A}u - c_{12}\,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} - c_1\sqrt{x_1}\\
\dot{x}_2 &= c_{12}\,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} - wc_2\sqrt{x_2}\\
y &= x_1
\end{aligned}
\tag{2}
$$

where x_1 and x_2 are levels in tank 1 and tank 2 respectively and A is a cross-section of the tanks, see Fig. 1. Note that both tank 1 and tank 2 have the identical cross-sections here. The disturbance $w \in \{0, 1\}$, depending on whether the valve c_2 is switched off or on respectively. In this case the level of a liquid in tank 2 might be greater than in tank 1. For that reason a more general model (2) has to be used.

To identify the system we have to find, besides the cross-sections A, the values of flow coefficients c_1, c_2 and c_{12}. The usual methods to treat the identification are based on applying a couple of certain experiments and measurements. Then the coefficients are computed by using either the steady-states of the system or the system linearization in a fixed operating point. However, both of them are rather slow. In addition, it is, usually, recommended to find a set of values in different steady states or operating points respectively and take their average finally. For that reason we, in what follows, suggest a different approach to the system identification which is based on finding a solution to the reduced nonlinear differential equations of the system (2). As a result we will be able to compute all the coefficients only by measuring the time of the respective experiments.

To identify the flow coefficient c_1 suppose that all valves are closed and the pump is inactive. Let $x_{10} \neq 0$ be a level of a liquid in tank 1. The experiment consists of opening the valve c_1 only and measuring the time τ that it takes to empty the tank from an initial value x_{10} to a final value x_{11}. Obviously, this situation can be modelled by the reduced nonlinear differential equation

$$\dot{x}_1 = -c_1\sqrt{x_1}$$

Even though the equation is nonlinear the solution can easily be found as follows

$$
\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= -c_1\sqrt{x_1}\\
\int_{x_{10}}^{x_{11}} \frac{1}{\sqrt{x_1}}\mathrm{d}x_1 &= -\int_0^{\tau} c_1\mathrm{d}t\\
\Big[2\sqrt{x_1}\Big]_{x_{10}}^{x_{11}} &= \big[-c_1 t\big]_0^{\tau}\\
2(\sqrt{x_{11}} - \sqrt{x_{10}}) &= -c_1\tau
\end{aligned}
$$

Finally

$$c_1 = \frac{2(\sqrt{x_{10}} - \sqrt{x_{11}})}{\tau}$$

If the final value x_{11} is chosen to be 0, which is usually the most reasonable choice, then the formula reduces to

$$c_1 = \frac{2\sqrt{x_{10}}}{\tau} \tag{3}$$

where τ is the time it takes to empty tank 1 completely from the initial value x_{10}.

Clearly, the analogous experiment can be repeated for the second tank giving us the formula

$$c_2 = \frac{2\sqrt{x_{20}}}{\tau} \tag{4}$$

where this time τ is the time it takes to empty tank 2 completely from its initial value x_{20}.

To identify the flow coefficient c_{12} a more advanced experiment is needed. Suppose that all valves are closed and both pumps inactive. Let $x_{10} \neq 0$ be a level of a liquid in tank 1 and $x_{20} < x_{10}$ be a level of a liquid in tank 2. This time the experiment consists of opening the valve c_{12} only and measuring the time τ it takes the level in tank 1 decrease from the initial value x_{10} to the final value x_{11}. Such a situation can be modelled by the following nonlinear differential equations

$$
\begin{aligned}
\dot{x}_1 &= -c_{12}\sqrt{x_1 - x_2}\\
\dot{x}_2 &= c_{12}\sqrt{x_1 - x_2}
\end{aligned}
$$

However, one can use either of them to find the solution. For instance the first equation yields

$$
\begin{aligned}
\frac{\mathrm{d}x_1}{\mathrm{d}t} &= -c_{12}\sqrt{x_1 - x_2}\\
\frac{1}{\sqrt{x_1 - x_2}}\mathrm{d}x_1 &= -c_{12}\mathrm{d}t
\end{aligned}
$$

Note that the situation during the experiment implies that $x_{10} - x_1 = x_2 - x_{20}$ and thus substituting $x_2 = x_{10} + x_{20} - x_1$ gives us

$$
\begin{aligned}
\int_{x_{10}}^{x_{11}} \frac{1}{\sqrt{2x_1 - x_{10} - x_{20}}}\mathrm{d}x_1 &= -\int_0^{\tau} c_{12}\mathrm{d}t\\
\Big[\sqrt{2x_1 - x_{10} - x_{20}}\Big]_{x_{10}}^{x_{11}} &= \big[-c_{12}t\big]_0^{\tau}
\end{aligned}
$$

$$\sqrt{2x_{11} - x_{10} - x_{20}} - \sqrt{x_{10} - x_{20}} = -c_{12}\tau$$

Finally

$$c_{12} = \frac{\sqrt{x_{10} - x_{20}} - \sqrt{2x_{11} - x_{10} - x_{20}}}{\tau}$$

Here, if the initial value x_{20} is chosen to be 0 and the final value x_{11} is chosen to be $x_{10}/2$, that is the levels in both tanks finally equal each other (note that the tanks have the identical cross-sections) the formula reduces to

$$c_{12} = \frac{\sqrt{x_{10}}}{\tau} \tag{5}$$

where τ is the time it takes the levels in both tanks equal each other.

Using the above formulas (3), (4) and (5) the flow coefficients c_1, c_2 and c_{12} of the laboratory plant were identified as $1.17 \cdot 10^{-2}$, $1.17 \cdot 10^{-2}$ and $2.65 \cdot 10^{-2}$ respectively. Finally, the cross sections of both tanks are approximately $A = 10.18 \cdot 10^{-4} m^2$.

3.2 *Discrete-time state-space representation*

To find a discrete-time state-space representation of the system (2) one needs to find a solution to the set of nonlinear differential equations and sample it by the sampling period T. Since the system equations (2) involve nonlinear functions we are, in general, not able to find any. In such a case one usually has to rely on approximations only. One of the possibilities is to employ the Taylor series expansion.
Assume that

$$\dot{x}(t) = f(x(t), u(t))$$

where f is analytic. Then one can write

$$x(t+T) = x(t) + \dot{x}(t)T + \frac{\ddot{x}(t)}{2!}T^2 + \frac{x^{(3)}(t)}{3!}T^3 + \cdots$$

However, it is usually sufficient to consider only the first two terms of the Taylor series expansion to approximate the system behaviour in which case one gets the well-known Euler approximation

$$\dot{x}(t) \approx \frac{x(t+T) - x(t)}{T}$$

Using such an approximation one can find the discrete-time state-space model from (2) as

$$\begin{aligned} x_1^+ &= x_1 + \frac{T}{A}u - c_{12}T\,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \\ &\quad -c_1 T\sqrt{x_1} \\ x_2^+ &= x_2 + c_{12}T\,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \\ &\quad -wc_2 T\sqrt{x_2} \\ y &= x_1 \end{aligned} \tag{6}$$

3.3 *Disturbance decoupling problem*

The standard theoretical solution to the disturbance decoupling problem, as outlined in Section 2, does not meet basic practical control requirements, as shown in what follows, and thus it is necessary to modify it accordingly.
To proceed with the disturbance decoupling we compute

$$\begin{aligned} y^+ &= x_1 + \frac{T}{A}u - c_{12}T\,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \\ &\quad -c_1 T\sqrt{x_1} \end{aligned}$$

Since y^+ directly depends on the input u, that is the relative degree of the system is 1, and it is not affected by the disturbance w, it can be decoupled. Note that in according to Theorem 1 we have $\mathcal{X} \cap \mathcal{Y} = \mathrm{span}_{\mathcal{K}}\{\mathrm{d}x_1\}$ and thus $p(x) = (0, -c_2\sqrt{x_2})^T$ is orthogonal to $\mathcal{X} \cap \mathcal{Y}$.
By solving for u the equation

$$y^+ = v$$

one gets

$$\begin{aligned} u &= \frac{A}{T}v - \frac{A}{T}x_1 + c_{12}A\,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \\ &\quad +c_1 A\sqrt{x_1} \end{aligned} \tag{7}$$

where v represents input to the compensated system which is reduced to the first order linear system $y^+ = v$ with the transfer function

$$F(z) = \frac{1}{z} \tag{8}$$

However, from a practical point of view the compensated system cannot respond in one sampling period T, like its transfer function (8) says, at least for lower sampling periods T, for we have a controller output constraint $u \in \langle 0, q_{max} \rangle$ where q_{max} is upper limit of the pump capacity. On the other side for higher sampling periods T the discrete-time approximation (6) of the continuous-time system (2) might no longer be sufficient. In addition, there obviously exist additional differences between the real plant and its continuous-time model (2) that have not been considered. For that reason, the real compensated system will possess oscillations even for not that high sampling periods T when the discrete-time approximation (6) is still accurate. Last but not least, the feedback (7) is not a controller at all. Obviously, it is only a static state feedback achieving the disturbance decoupling, however, with no intention to track the reference signal or to eliminate unmodelled disturbances. For all the aspects listed above, such a solution is practically not applicable and needs to be modified accordingly.

There exist several possibilities how to overcome the problems. One of them, discussed in Žilka and

Halás (2010) for continuous-time case, suggests to modify the feedback (7) such that the whole second tank, together with the disturbance, becomes unobservable. Then, one only has to design a controller for a one-tank system which is, obviously, a trivial task and plenty of solutions have been given. This seems to be a reasonable choice also in the discrete-time case. The feedback (7) can easily be modified to the form

$$u = v + c_{12} A \,\mathrm{sign}(x_1 - x_2)\sqrt{|x_1 - x_2|} \qquad (9)$$

under which the compensated system takes now the form of one-tank system with the discrete-time state-space model

$$\begin{aligned} x_1^+ &= x_1 + \frac{T}{A} v - c_1 T \sqrt{x_1} \\ y &= x_1 \end{aligned} \qquad (10)$$

Then the controller can easily be designed by the system linearization in a fixed operating point (x_{10}, v_0, y_0) which reads

$$\begin{aligned} \Delta x_1^+ &= \Delta x_1 + \frac{T}{A}\Delta v - \frac{c_1 T}{2\sqrt{x_{10}}}\Delta x_1 \\ \Delta y &= \Delta x_1 \end{aligned}$$

where $\Delta x_1 = x_1 - x_{10}$, $\Delta v = v - v_0$ and $\Delta y = y - y_0$. It has the transfer function

$$F(z) = \frac{K}{z - D}$$

where $K = T/A$ and $D = 1 - \frac{c_1 T}{2\sqrt{x_{10}}}$.
If one wants the transfer function of the compensated system

$$G(z) = \frac{R(z)F(z)}{1 + R(z)F(z)}$$

to take the form of a first order linear system with the time constant T_1, then the solution is given by a linear discrete-time PI controller with the transfer function

$$R(z) = \frac{(1-\lambda)(z-D)}{K(z-1)} = \frac{1-\lambda}{K}\left(1 + \frac{1-D}{z-1}\right)$$

where $\lambda = e^{-T/T_1}$.

Remark 2. Note that more advanced solution, dealing also with the system linearization, the controller output constraint and two different disturbances to decouple, has been suggested in Žilka et al. (2009).

The closed loop structure is depicted in Fig. 2. The responses of the real laboratory plant are shown in Fig. 3 where one can observe the differences between the linear PI-controller with and without the disturbance decoupling (9). In the latter the disturbances practically do not affect the system output. However, since we have the constrained controller output and since only a standard PI-controller has been used to control the system both solutions admit an overshoot. A non-overshooting solution has been suggested in Žilka and Halás (2010).
The parameters were chosen as follows: $T = 0.25s$, $T_1 = 5s$ and $x_{10} = 0.2m$.

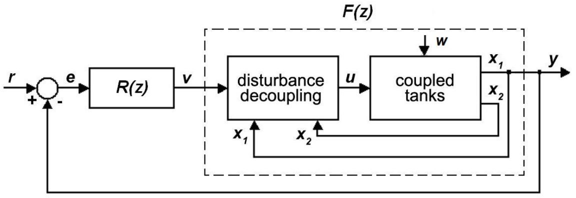


Fig. 2. Closed loop

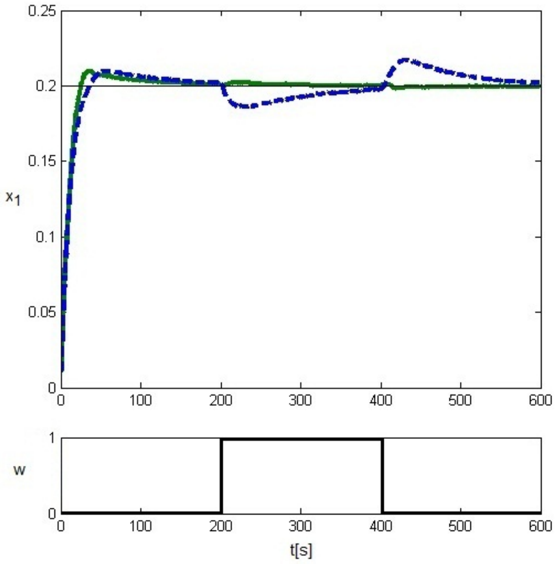


Fig. 3. Closed loop responses: PI-controller with (solid, green line) and without (dashed, blue line) the disturbance decoupling.

Our final note is related to the slight modification of the disturbance decoupling (9) which is appropriate from a practical point of view and has been implemented in our solution. When the valve c_2 is closed the equations (2) imply that in a steady state one, theoretically, has $x_1 = x_2$. However, in practice, there are differences between x_1 and x_2 caused at least by sensors calibration and noise. Therefore the term $\mathrm{sign}(x_1 - x_2)$ in (9) oscillates between 1 and -1 and thus produces small oscillations of the controller output especially in steady states, which is, of course, inconvenient. The problem can and has been overcome easily by adding a deadzone to the disturbance decoupling making it inactive whenever the difference between x_1 and x_2 is less than $2 \cdot 10^{-3}m$.

4. CONCLUSIONS

In this work, an attempt to overlap the gap between control theory and control practice was studied. An important practical control problem given by the disturbance decoupling problem were applied on coupled tanks. It was shown that the initial theoretical solution to the disturbance decoupling problem does not satisfy the basic control requirements. For that reason, the solution was modified accordingly. This resulted in the PI-controller with the disturbance decoupling. As a result the disturbances practically did not affect the system output of the real laboratory plant.

ACKNOWLEDGEMENTS

This work was supported by the Slovak Grant Agency by grants No. VG-1/0656/09 and VG-1/0369/10 and by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

References

E. Aranda-Bricaire, C. Moog, and J. Pomet. A linear algebraic framework for dynamic feedback linearization. *IEEE Transactions on Automatic Control*, 40:127–132, 1995.

E. Aranda-Bricaire, Ü. Kotta, and C. Moog. Linearization of discrete-time systems. *SIAM Journal of Control Optimization*, 34:1999–2023, 1996.

G. Conte, A.M. Perdon, and C.H. Moog. The differential field associated to a general analytic nonlinear dynamical system. *IEEE Transactions on Automatic Control*, 38:1120–1124, 1993.

G. Conte, C.H. Moog, and A.M. Perdon. *Algebraic Methods for Nonlinear Control Systems. Theory and Applications.* Communications and Control Engineering. Springer-Verlag, London, 2nd edition, 2007.

M. Fliess. A new approach to the noninteracting control problem in nonlinear systems theory. In *23rd Allerton Conference*, Monticello, 1985.

M. Fliess. Reversible linear and nonlinear discrete-time dynamics. *IEEE Transactions on Automatic Control*, 37:1144–1153, 1992.

J. W. Grizzle. Controlled invariance for discrete-time nonlinear systems with an application to the disturbance decoupling problem. *IEEE Transactions on Automatic Control*, 30:868–873, 1985.

M. Halás and V. Žilka. Noninteracting control of coupled tanks: from theory to practice. In *Eurocast*, Gran Canaria, Spain, 2011.

M. Halás, Ü. Kotta, Z. Li, H. Wang, and C.M. Yuan. Submersive rational difference systems and formal accessibility. In *International Symposium on Symbolic and Algebraic Computation (ISSAC)*, Seoul, Korea, 2009.

Ü. Kotta. *Inversion Method in the Discrete-Time Nonlinear Control Systems Synthesis Problems.* Springer, Berlin, 1995.

Ü. Kotta, A.S.I. Zinober, and P. Liu. Transfer equivalence and realization of nonlinear higher order input-output difference equations. *Automatica*, 37:1771–1778, 2001.

V. Žilka and M. Halás. Disturbance decoupling of coupled tanks: from theory to practice. In *IFAC Symposium on System, Structure and Control*, Ancona, Italy, 2010.

V. Žilka, M. Halás, and M. Huba. Nonlinear controllers for a fluid tank system. In *R. Moreno-Díaz, F. Pichler, A. Quesada-Arencibia: Computer Aided Systems Theory*, Lecture Notes in Computer Science, Springer, 2009.

X. Xia, L.A. Márquez-Martínez, P. Zagalak, and C.H. Moog. Analysis of nonlinear time-delay systems using modules over non-commutative rings. *Automatica*, 38:1549–1555, 2002.

ULTRASONIC HYBRID MAP FOR NAVIGATION OF MOBILE ROBOT

F. Duchoň*, L. Jurišica**

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, Bratislava, Slovakia

** e-mail: frantisek.duchon@stuba.sk*

*** e-mail: ladislav.jurisica@stuba.sk*

Abstract: Paper deals with principles of ultrasonic hybrid map proposed for indoor mobile robot system. This map is applied in navigation of mobile robot. At the beginning, paper presents brief description of indoor mobile robot system used for testing and developing of algorithms. Major section of paper deals with map creating. In the first step, it is local metric map with probabilistic model of ultrasonic sensor. In the second step, it is global metric map, which is created by connection of local metric maps. In the last step, it is simplification of environment representation from global metric map to topological map. In this manner ultrasonic hybrid map is created and it can be used in both reactive and global navigation of mobile robot.

Keywords: mobile robot, metric map, topological map, hybrid map

1 INTRODUCTION

Mobile robots can be utilized in many places of human life. Generally, they are used as service robots. Definition of service robot is unclear and complicated, but it is possible to claim, that service robot performs such tasks, that help, assist or completely replace human in various activities. Reason of service robotics development is to raise human productivity.

Mobile robots can be designed for various environments. Essential tasks, that every robot has to perform is to localize and navigate itself in environment. For successful achievement of these tasks, robot needs some sensors. These sensors are used for perceptions likewise humans do. In world of mobile robotics exist many sensors, but generally we can divide them in two groups:

Sensors used for localization (GPS, encoders, INS, gyroscope, accelerometer, magnetic compass, etc.)

Sensors used for navigation (ultrasonic sensors, laser rangefinders, infrared sensors, visual systems, proximity sensors, contact sensors etc.)

Moreover, for specific applications, it exists many others high specialized sensors. And next, with application of some mathematic methods, sensors used for navigation can also be used for localization. It seems difficult, but majority of mobile robots use for its navigation and localization some specific and abstract representation of environment. This representation is called map of environment.

There are three types of these maps. First type is metric map. Metric map represents environment with grid, in which every cell represents exactly specified size of environment. Key problem is to set appropriate size of the cell. If size of cell is too small, then number of cells is too big and there are great demands on CPU performance and memory. If cell size is too big, representation of environment is not so accurate and it can cause problems within navigation and localization.

Second type is topological map. Topological map represents environment in graph structure. There are no metric properties stored in this map. Graph consists of places and edges. Edges represent passages between two places. Advantages of topological map are lower consumption of CPU performance within path planning and lower memory consumption within map creating.

Third type is hybrid map, which combines two antecedent types. It means, that robot is globally planning its path in global topological map, but for its reactive navigation and obstacle avoidance it uses local metric map. Local metric map is used in places of global topological map. Combination of particular types of maps improves robot possibilities for localization, navigation and it also decreases demands on CPU performance. This combination is near to human reflection.

2 INTRODUCTION

Mobile robot used for experiments is indoor mobile robot. Robot has three wheels, two of them are driven, so it is a differentially driven robot (Miková 2008). Kinematics scheme and configuration of sensors can be seen on Fig. 1. Circular construction of robot enables rotation around axis z without intervention of robot to the environment. Sensorial system consists of nine ultrasonic sensors and nine infrared sensors. Seven forward ultrasonic sensors are used for construction of local metric map, two backward sensors are used just for obstacle avoidance like all of nine infrared sensors. Ultrasonic sensors used in mobile robot are MUST01 sensors with detection range from several decimetres up to 5 (3 metres guaranteed) metres with 0,1% accuracy over entire range at stable temperatures. Beam angle is 15° and repetition rate is 10Hz astable (for our experiments repetition rate was 600ms stable).

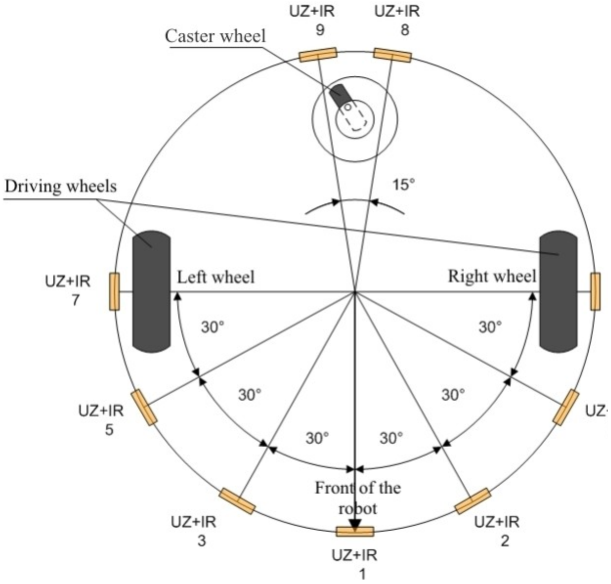


Fig. 1. Kinematics scheme of mobile robot and configuration of sensors.

Outputs from ultrasonic sensors are analog 0-5 VDC. That's why Advantech I/O card is used. Moreover, for interpretation of measured distances, it is need to know transfer characteristic (Fig. 2, Tab. 1).

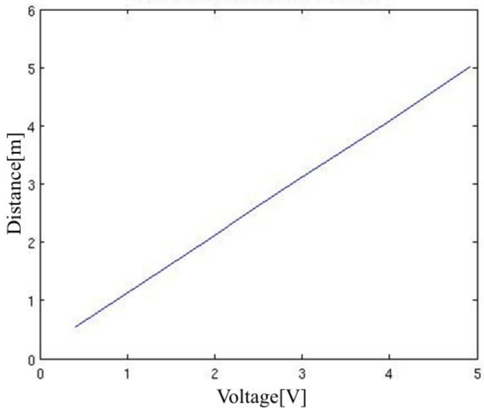


Fig. 2. Transfer characteristic of ultrasonic sensor MUST01.

Voltage [V]	0.415	0.910	1.420	1.920	2.425	2.935	3.450	3.970	4.460	4.920
Distance [m]	0.550	1.050	1.550	2.050	2.550	3.050	3.550	4.050	4.550	5.020

Table 1. Transfer characteristic of ultrasonic sensor MUST 01.

Because the robot is differentially driven with negligible influence of wheels slippage and friction, its estimation of position is derived from number of wheels revolutions. That's why there are used incremental encoders MR ENC type L with 1024 impulses per revolution. Incremental encoders are placed on motor shaft. Because there are used motors Maxon RE40 with gearbox GP42C (15Nm, 43:1), every revolution of wheel evokes 1024*43 pulses. Moreover, the position of mobile robot is derived from odometry, which is based on kinematics scheme (Fig. 3). Inputs are velocities of wheels derived from number of wheels revolutions.

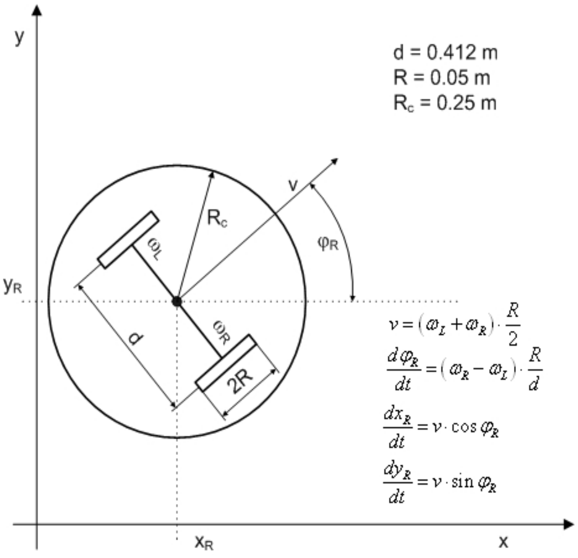

y
d = 0.412 m
R = 0.05 m
R$_c$ = 0.25 m
v
R$_c$
φ_R
ω_L
y$_R$
d
ω_R
2R
$v=(\omega_L+\omega_R)\cdot\frac{R}{2}$
$\frac{d\varphi_R}{dt}=(\omega_R-\omega_L)\cdot\frac{R}{d}$
$\frac{dx_R}{dt}=v\cdot\cos\varphi_R$
$\frac{dy_R}{dt}=v\cdot\sin\varphi_R$
x$_R$
x

Fig. 3. Kinematics scheme used for odometry.

3 LOCAL METRIC MAP

Local metric map of environment is created following data from ultrasonic sensors. For single ultrasonic sensors, we can divide area covered by this sensor in four regions (Fig. 4) (Murphy 2000) (Hanzel 2007).

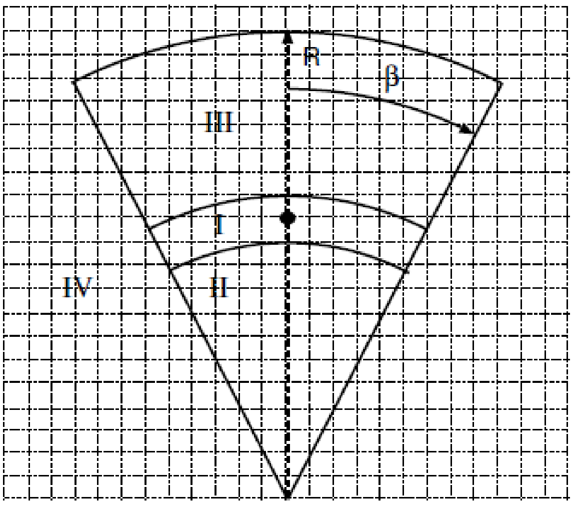


Fig. 4. Four regions within ultrasonic sensor measurement (Murphy 2000).

Region I. stands for detected object. This region has curved shape, because information about obstacle must be propagated through entire arc. Height of this region is specified by resolution of local metric map (thus ultrasonic sensor) and by errors of measurement. In our model, region I. and III. are the same. Region II. is interpreted as free region. It is logical, because if there is another obstacle, region I. will be closer to the ultrasonic sensor. Region III. is theoretically covered by sensor, but practically there is no information about this region based on data from sensor. On the other hand, majority of models join region I. and III. Region IV. is region outside of measurement range. For ultrasonic sensor MUST01 is R equal to 3m and β is equal to 7,5°.

Data from ultrasonic sensor can be written into local metric map with simple additive model. However usage of some probabilistic model is more correct. Model like that is more corresponding to properties of real ultrasonic sensor. Our model is derived from basic Elfes model (Elfes 1989)(Elfes 1992). For region II. is probability of occupation for each cell computed as:

$$P(x,y) = P_\beta . P_R \, , \tag{1}$$

where:

$$P_\beta = 1 - \left(\frac{\frac{16}{3}.\beta^2 - \frac{2}{3}\alpha.\beta}{\alpha^2} \right) \tag{2}$$

$$P_R = \left(R_{po\,\mathrm{int}} - R_{\min}\right).\Delta_e \tag{3}$$

$$\Delta_e = \frac{p_e}{R_{mes} - R_{\min}} \tag{4}$$

β is angular radius of sensor cone, α is angular width of sensor cone, $R_{po\,\mathrm{int}}$ is radius of actually written point into local metric map derived from position of ultrasonic sensor, $R_{\min}$ is minimal radius derived from minimal measurement range of sensor, R_{mes} is actually measured radius of obstacle derived from sensor data and p_e is range of probability interpreted for region II.

For region I. and III. is probability of occupation for each cell computed as:

$$P(x,y) = P_\beta . P_r \, , \tag{5}$$

where:

$$P_\beta = 1 - \left(\frac{\frac{16}{3}.\beta^2 - \frac{2}{3}\alpha.\beta}{\alpha^2} \right) \tag{6}$$

$$P_r = 1 - \left(R_{po\,\mathrm{int}} - R_{mes}\right).\Delta_o \tag{7}$$

$$\Delta_o = \frac{p_o}{R_{\max} - R_{mes}} \tag{8}$$

β is angular radius of sensor cone, α is angular width of sensor cone, $R_{po\,\mathrm{int}}$ is radius of actually written point into local metric map derived from position of ultrasonic sensor, $R_{\max}$ is maximal radius derived from minimal measurement range of sensor, R_{mes} is actually measured radius of obstacle derived from sensor data and p_o is range of probability interpreted for region II. Equations (2) and (6) were derived heuristically.

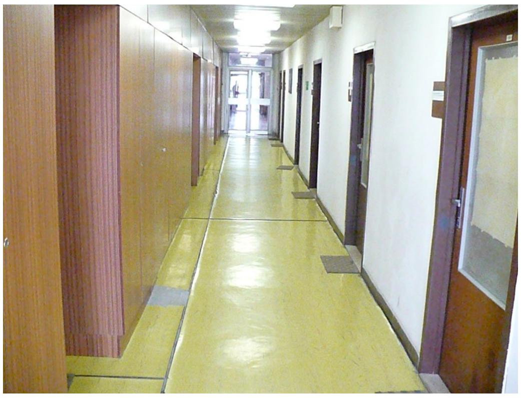
In our model values of probabilities ranging from 0 to 0,4 represent unoccupied space, values of probabilities ranging from 0,6 to 1 represent occupied space, values ranging from 0,4 to 0,6 represent lack of information. It follows, that both p_e and p_o are equal to 0,4. With modification of mutual rate of p_e and p_o, it can be given emphasis on faster record of occupied space or on faster record of unoccupied space. This is important, if robot performs its activities in environment with movable obstacles.

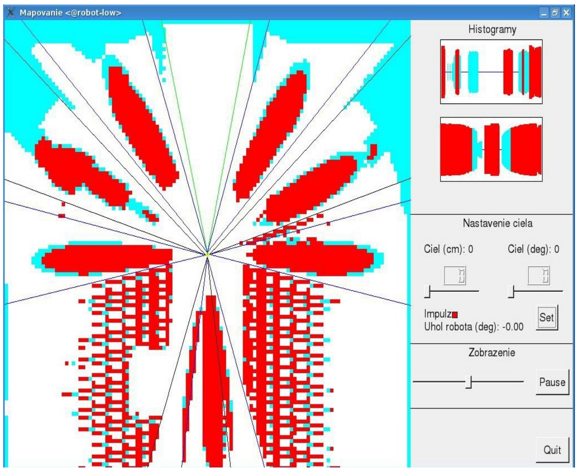
Consequently, one reading of ultrasonic sensor is written to local metric map. This model is used for each ultrasonic sensor of the robot. However, one sensors reading is not useful for navigation. That's why multiple readings must be put to correlation. For this purpose, it's used well known Bayesian update rule:

$$P(O\,|\,NR) = \frac{P(NR\,|\,O).P(O)}{P(NR\,|\,O).P(O) + [(1 - P(NR\,|\,O)).(1 - P(O))]} \tag{9}$$

Where $P(O\,|\,NR)$ is new empirical probability of occupation of cell, $P(NR\,|\,O)$ is priory probability of occupation of cell defined from new sensor reading and $P(O)$ is priory probability of occupation of cell defined from local metric map with previous sensors readings. Despite of the simplicity of update rule, it has some disadvantages. If any of priory probabilities is equal to 0 or 1, new empirical probability will not change. This is ineligible in environment with movable obstacles. That's why this rule in our model was modified by means of restriction of probability values. Probability of occupation of cell cannot reach value of 0 or 1. It can be just quite near to these values.

Seeing that robot is moving in environment, this fact must be reflected in local metric map. The movement of robot is divided into two types: translation and rotation. Translation express itself in local metric map as shift of cells in direction of movement. This means that cells in direction of movement, regarding the amount of the movement, are filled with value 0,5. Thus, this cells were not scanned by sensors and robot has lack of information about these cells. Cells in opposite direction of movement, regarding the amount of the movement, are simple deleted. Remaining cells are moved to new positions, that are defined by the amount of the movement. There are two ways, how to achieve rotation of local metric map. First way is complicated and computational hungry. It is rotation of grid structure around its centre. Second way, used in our model, is to rotate robot in local metric map itself. In this regard regions of sensors records are expressed through outlines. Afterwards it is simple to rotate outlines and fill corresponding cells with probabilities.





Mapovanie <@robot-low>

Histogramy

Nastavenie ciela

Ciel (cm): 0 Ciel (deg): 0

Impulz🟥
Uhol robota (deg): -0.00

Set

Zobrazenie

Pause

Quit

Fig. 5. Example of real environment and one local metric map of this environment.

4 GLOBAL METRIC MAP

The use of the local metric map alone is not very useful for navigation of robot. It can navigate through environment reactively, but for autonomous robot is needed to know and record information about already known environment. Afterwards, this information can be applied in global planning of path (Župa 2008).

Global metric map is constructed as fusion of multiple local metric maps. We assume, that startup position of robot is known and it is predetermined. Local metric maps can be easily and simply connected, because their orientation is equivalent. This property of equivalent orientation is guaranteed by rotation of robot in local metric map. Accordingly, local metric maps are connected in terms of shifts of their centres. In addition there is applied compensation from robot movement in order to preserve perpendicularity of map. For these reasons it is not needed to record positions from which were cells of global map recorded.

After connection of local metric maps in one global metric map, some threshold must be applied. Local metric map is described by probabilities for each cell. In global metric map used for global planning cannot figure probabilities, because global planning usually uses exact mathematical methods without probabilities. After threshold application, area of local metric map is erased from global metric map. Cause of this step are fast changes in local metric map used for reactive navigation, that are not very effective for global planning.

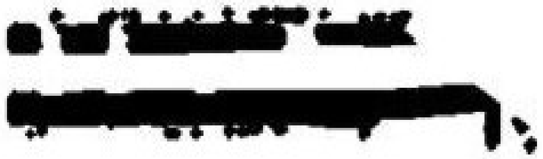


Fig. 6. Example of global metric map (one pixel represents 5x5cm, length of area approx. 20m).

5 TOPOLOGICAL MAP

In most cases, global metric map of environment is very demanding on memory. That's why the structure of global metric map is simplified to topological map. One way how to construct topological map from metric map is to create Voronoi diagram. Then with set of critical points topological map can be created (Thrun 1999)(Siegwart 2004). Usually construction of Voronoi diagram based on metric map is very computational hungry (Niemuller 2003)(Choset 2005). Request on our model, that every step of hybrid map creation must be done online, leads to own method called orthogonal equidistant diagram.

This method seeks for equidistant points between two obstacles in orthographic directions. Thereby way originates discontinuous diagram, which is determined in online mode. Despite of discontinuities in diagram, many continuous lines appear. These lines represent nodes of topological map. Discontinuities of environment (such as doors, passages, etc.) express themselves as discontinuities in map, so they divide environment on nodes.

If every isolated continuous line represents one topological node, then it is possible to search for neighbourhood nodes. Likewise it is possible to find two marginal points of each node. Between these marginal points, it can be created edges between marginal points (thus nodes) and from distance of marginal points, each edge can be priced by value of this distance. For detection of marginal points, some masks are used (Fig. 7).

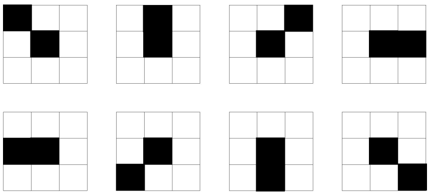


Fig. 7. Examples of masks.

Generally for each marginal point stands:

$$|f(x,y)| = 0 \wedge \sum_{\substack{i=-1\\ j=-1}}^{\substack{1\\ 1}} |f(x+i,y+j)| = 1785 \qquad (10)$$

Where $|f(x,y)|$ is size of image intensity on position (x,y). Equation (10) says, that marginal point is in the middle of the mask and in its surroundings are seven white points with image intensity equals to 255 and one black point with image intensity equals to 0.

Fig. 8. Examples of found marginal points (red points).

Node of topological map is defined, if there is a connection between two marginal points and the distance between these marginal points is higher than empirically specified distance. Edge of topological map is defined, if there isn't connection between two visible (i.e. no obstacle between them) marginal points and the distance between these marginal points is smaller than empirically specified distance. Moreover, edges and nodes can have distance values between its marginal points. This property can be useful in global path planning.

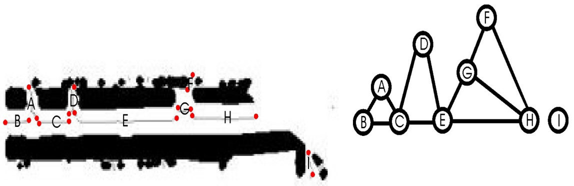


Fig. 9. Example of topological map created from global metric map.

6 CONCLUSIONS

Ultrasonic hybrid map is very powerful and robust tool for autonomous mobile robot navigation. Despite of many drawbacks, map created with ultrasonic sensors is accurate enough to provide resources for mobile robot navigation. Of course, there are sensors which provide more accurately and more precisely information, such as laser scanner or camera, but ultrasonic sensors can be used in many less expensive robots, that's why they are still popular in scientific world.

ACKNOWLEDGMENTS

This work was supported by MS SR under contract VMSP-P-0004-09.

6 REFERENCES

Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E. and Thrun, S. (2005). Principles of Robot Motion (Theory, Algorithms and Implementations). Massachusetts Institute of Technology. ISN 0-262-03327-5.

Murphy, R.R. (2000). Introduction to AI Robotics. Massachusetts Institute of Technology. ISBN 0-262-13383-0.

Hanzel, J. (2007). Environment mapping on base of ultrasonic measurements (In Slovak: Mapovanie prostredia na báze meraní ultrazvukom.). Bratislava, STU v Bratislave FEI, 2007.

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. Computer, Volume 22, Issue 6, p. 46-57. Carnegie Mellon University.

Elfes, A. (1992). Dynamic Control of Robot Perception Using Multi-Property Inference Grids. IEEE International Conference on Robotics and Automation.

Thrun, S. and Bücken, A. (1999). A Learning Maps for Indoor Mobile Robot Navigation. AI Magazine, Volume 99, p. 21-71.

Siegwart, R. and Nourbakhsh, I.R. (2004). Introduction to Autonomous Mobile robots. MIT Press, Cambridge, Massachusetts Institute of Technology. ISBN 0-262-19502-X.

Niemuller, T. and Widyadharma, S. (2003). Artifical Intelligence - An Introduction to Robotics. Proseminar Artifical Intelligence.

Miková, Ľ., Kelemen ,M., Kelemenová, T. (2008). Four wheels inspection robot with differential wheels control (In Slovak: Štvorkolesový inšpekčný robot s diferenčným riadením kolies). Acta Mechanica Slovaca, vol. 12, Issue 3-B, p. 548-558. ISSN 1335-2393.

Župa, T., Židek, K., Líška O., Živčák, J. (2008). Global navigation and collision protection system of mobile robots for interoperation transport. ICCC 2008 : 6th IEEE International Conference on Computational Cybernetics. November 27-29, 2008, Stará Lesná, Slovakia. - Budapest : IEEE, 2008, p. 199-201. ISBN 978-1-4244-2875-5.

CONTROL OF AN RC HELICOPTER MODEL THROUGH USB INTERFACE

Ivan Pestún[*], **Miroslav Halás**[*], **Peter Kurčík**[*]

** Slovak University of Technology, Ilkovicova 3, 812 19 Bratislava, Slovakia*
fax : +........and e-mail : [ivan.pestun, miroslav.halas, peter.kurcik]@stuba.sk

Abstract: This paper presents interface between personal computer and radio controlled helicopter model through an electronic component – microcontroller board. The communication is established through USB interface. The paper describes philosophy of manual RC model control and exploitation of a classical RC model transmitter for controlling an RC model through the computer. Collection of feedback data and GUI for controlling the helicopter are also described.

Keywords: *helicopter control, USB interface, RC model, PWM, PPM*

1 INTRODUCTION

The helicopter is an unstable MIMO system and therefore suitable for tryout of various control problems and algorithms. To retain the helicopter in a stable and controlled flight is a very complex problem. A number of works have been dedicated to study the problem and related subtasks, as for instance creating a mathematical model of a helicopter in (Fogh, et al., 2004; Mettler, 2003; Pestun, 2009), UAV helicopter design (Cai, et al., 2008), stabilization of an RC helicopter in a hover (Hald, et al., 2006; Hald, et al., 2005), to mention a few. Designed control algorithms are usually tested on helicopter models of various sizes and types. For such experiments, it is convenient to put a helicopter in a laboratory and ensure it against damage. There are many possibilities how to control the helicopter. Frequently used approach for controlling the helicopter is to put the control unit directly on the helicopter's body with all the algorithms ensuring autonomous flight and to control desired position and other flight parameters wireless with a ground station PC, see for example (Fogh, et al., 2004; Hald, et al., 2006). This method is good to use especially when the helicopter is about to use outdoors for free flying with already tested control algorithms. The main disadvantage is lower performance of the control unit and the requirement of reprogramming the whole unit in case of every change. Another way how to control the RC helicopter, more suitable for laboratory tests, is to make interface with a high-performance PC and edit the flight and stabilization algorithms there, see for example (Andersen, et al., 2008). Our goal was to create interface between the helicopter and PC with high performance. The interface is also small control unit, but it is only used for transforming and forwarding signals from computer to the helicopter and for collecting feedback from sensors and sending them back to computer. So there is no need to reprogram the unit during the experiments, because all the control algorithms are programmed on the PC where they can be, thanks to enough performance, more easily edited in more sophisticated and user friendlier programs than microcontroller programming language is. Hence, this way was found to be more suitable for testing miscellaneous controllers for helicopter stabilization and control in laboratory conditions, forming the main scope of our interest in this work. In particular, the attention is paid to the communication established through USB interface. For that reason the so-called trainer mode is employed, allowing us to interconnect PC and the helicopter. Note that the similar idea was suggested in (Net-Scale Technologies, 2004) where an RC model of a car was controlled.

The paper is organized as follows. Section 2 describes signals used for control of the RC model, transmitter's mode used for interfacing the RC model and microcontroller board used as the interface element, Section 3 describes sensors used for feedback and their inadequacies and, finally, Section 4 describes the developed software for the helicopter control.

2 BASIC PRINCIPLES OF MANUAL RC MODEL CONTROL

2.1 *Signals used for control of an RC model*

The RC models, whether they fly, drive or float, consist of receiver, which receives the signals from a transmitter and controls of all actuators like driving gears and servo motors. Driving gears give the RC model propulsion and servo motors handle the controls. All this RC models actuators are controlled by an RC pulse signal. This signal is a specialized form of the so-called PWM (Pulse Width Modulation) signal. Typically the signal's period is 20 *ms* (50 *Hz* PWM frequency), but the signal itself has a width from 1 *ms* to 2 *ms* (The Model Electronics Company, Tech. note).

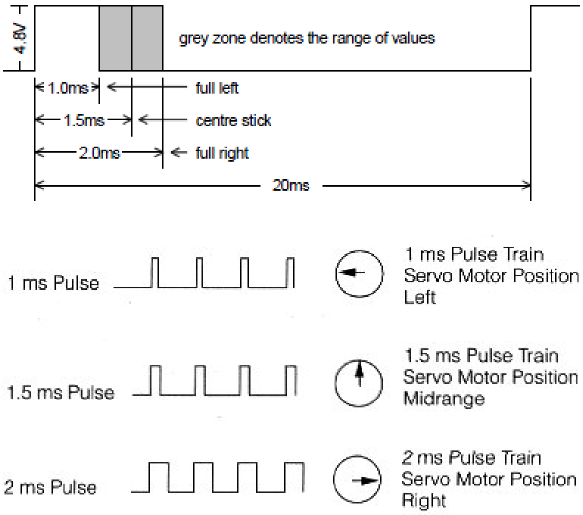

4.8V
grey zone denotes the range of values
1.0ms
full left
1.5ms
centre stick
2.0ms
full right
20ms
1 ms Pulse
1 ms Pulse Train Servo Motor Position Left
1.5 ms Pulse
1.5 ms Pulse Train Servo Motor Position Midrange
2 ms Pulse
2 ms Pulse Train Servo Motor Position Right

Fig. 1. RC pulse signal and the corresponding servo position examples.

The signal width represents the position of the servo or the power of the driving gear. The midrange (or center) position represents the signal with width of 1.5 ms. Then the left position is represented by the signal with width of 1 ms, and the right position is represented by the signal width of 2 ms (Vastianos, dipl. thesis), as shown in Fig. 1.

Each single actuator is controlled by its own PWM signal supplied by the receiver. For example, when the model has 6 actuators (like in our case of RC helicopter), 6 independent PWM signals (6 channels) are needed. Sending each PWM signal from the transmitter for each actuator separately would cost a lot of energy and the update frequency would depend on a number of signals. In order to be more effective, the information related to each channel is sent in a serial fashion, one after the other, over a single RF link (The Model Electronics Company, Tech. note). This signal is the so-called Pulse Position Modulation, or PPM.

PPM is a modulation which uses pulses with variable width and uniform height and time between the pulses. The PPM signal consists of channel sections (the number depends on the number of actuators) and a synchronization time space. Channel section is composed of a fixed time, usually 0.5 ms, and of a variable time with length of 0.5 ms to 1.5 ms (Fig. 2a). The synchronization time is the "dead time" and varies with the number of channels and also with the channel content. However, even for a system with more than 8 channels, it is much longer than the time between channel pulses. The receiver uses this synchronization time to synchronize itself to the pulse train, so that the positional information for channel 1 always drives the correct servo (The Model Electronics Company, Tech. note). Also in case of a signal loss in the middle of the transmitting, the receiver knows that after a time space much longer than 0.5 ms the first channel follows. Thus, when the signal recovers, the channels will not be disordered.

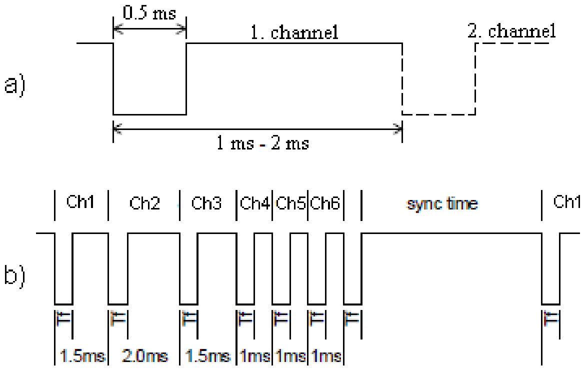


Fig. 2. a) single channel section of a PWM signal; b) example of a PPM signal carrying information for six channels

The variable time of the pulses can be compared to the pulse width in PWM. For example, 6 PWM pulses with the pulse widths of 1.5 ms, 2 ms, 1.5 ms, 1 ms, 1 ms, 1 ms will be modulated to a PPM signal shown in Fig. 2b.

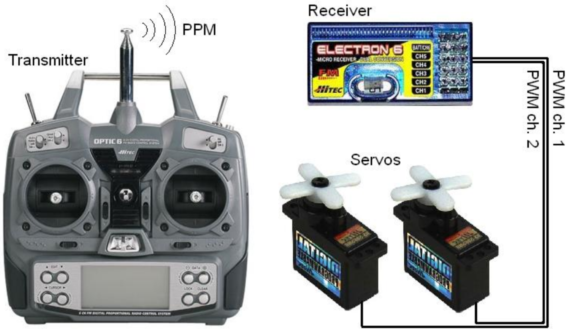


Fig. 3. Communication between the transmitter and receiver through PPM signals and servo control from the receiver through PWM signals.

In a standard manual hand control, the amount of variations on the transmitter's joysticks changes into the above mentioned PPM signal carrying information for controlling all the actuators, which is transformed into a radio signal and transmitted in a specific frequency. The receiver on the model, which is controlled by the transmitter, constantly monitors this frequency. When the radio burst from the transmitter is received, the receiver transforms the PPM signal to the PWM signals and split them up to the channels - for each actuator its own PWM signal, see Fig. 3.

2.2 *Description of a "trainer" mode*

In this section the so-called trainer mode is described, for it is employed to interconnect PC and the RC model.

In case when for instance the trainee pilot has problems with the control of the RC model, most of the transmitters are fitted with a trainer mode. For using this mode, two transmitters have to be interconnected with a cable, see Fig. 4. The signal, which the interconnected transmitter sends through the cable, is the inverted PPM signal with amplitude of transmitter's supply voltage.

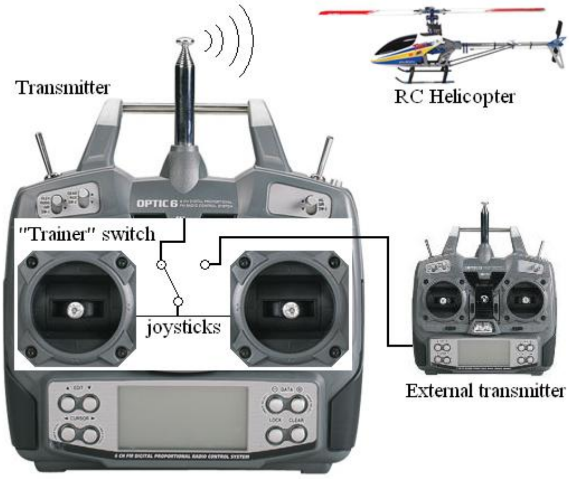


Fig. 4. Trainer switch function.

When the trainer mode on the transmitter, which is controlling the model, is turned on, the transmitter stops to send signals created from the positions of the joysticks and starts to send the signals from the interconnected transmitter, which is for instance in the hands of experienced pilot. However, in this work this idea was used to develop and establish an interconnection with PC, rather than with the second transmitter.

2.3 *Microcontroller interface board description*

To achieve such an interconnection the interface board, which creates PPM signals asked by the computer, was connected to the trainer port of the receiver. Through the USB the PC sends desired values for all channels to microcontroller interface board which creates, based on this values, corresponding PPM signal and sends it to the transmitter. For this purpose the transmitter is set to the trainer mode all the time, so the signals are forwarded from the interface board to the helicopter's receiver. Besides the sending of control signal, the microcontroller interface board ensures data collection from sensors and shifts them to the computer for evaluation and handling. The scheme of connection is shown in Fig. 5.

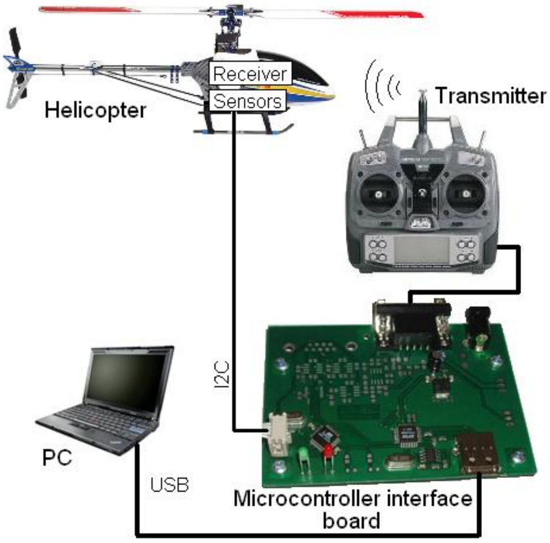


Fig. 5. The scheme of Helicopter – PC interconnection.

In order to match the desired transmitter's input polarity of the PPM signal via "trainer" connection, the PPM signal's amplitude have to be switched to the level of the transmitter's supply voltage. For the case of the chosen transmitter, it is the voltage 9.6 *V*. Example of the PPM signal entering the transmitter with all channels set to width of 1 *ms*, which represents left position on all the servos, can be seen in Fig. 6a.

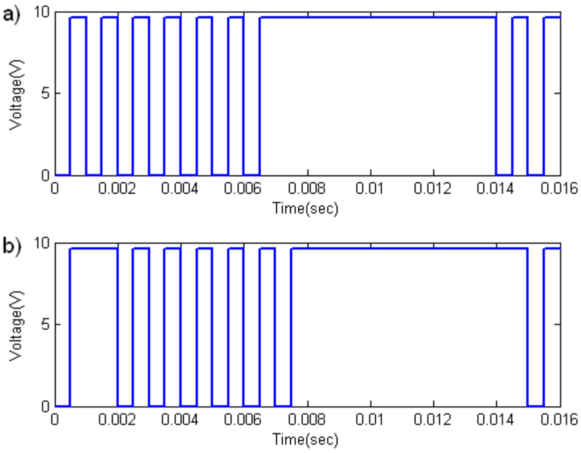


Fig. 6. PPM signal entering the transmitter from the "trainer" cable. a) All channels set to width of 1 *ms*. b) Channel 1 set to 2 *ms*.

Example of the PPM signal entering the transmitter with the first channel set to width of 2 *ms*, which represents right position of the servo connected to the first channel, can be seen in Fig. 6b.

To switch the PPM signal's amplitude to the level of the transmitter's supply voltage an NPN transistor was used in the "trainer" cable. However, the transistor also inverts the PPM signal. Therefore the signal generated by the microprocessor needs to be created in an inverted form (see Fig. 7). Such a PPM signal modification was necessary, otherwise the transmitter would not respond to the signal TTL levels from the microprocessor.

The PPM signal is generated in the microprocessor using a 16 bit counter TCNT1, set in a fast PWM mode with a variable length of the counter's TOP register. Since the 6 channels are needed, a state machine is created within the counter's overflow interrupt routine to alter the counter's registers after each transmitted impulse and counter overflow. The state machine sequence cycles through all 6 channels and then holds the signal low to transmit the synchronization delay. The whole process repeats in every cycle. In case of change of some channel, the new information affects the PPM signal in the next cycle. The outgoing signal from the microprocessor has amplitude of 5 *V*, synchronization time 7.5 *ms* and seven pulses with width 0.5 *ms*, displaced from each other 0.5 *ms* – 1.5 *ms*. Example of the signal generated by the microprocessor, which will lead, after inverting by the transistor in the "trainer" cable to the PPM signal shown in Fig. 6, can be seen in Fig. 7.

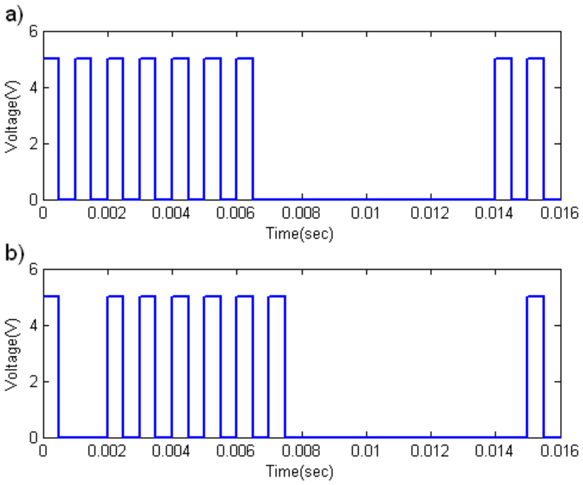


Fig. 7. Inverted PPM signal generated by the microprocessor. a) All channels set to width of 1 ms. b) Channel 1 set to 2 ms.

The PPM generator hardware is based on the Atmel ATmega8 microcontroller which features counters with PWM generation option and also serial port which is used to interface with PC using the serial to USB converter. FTDI FT232RL chip play a role of a serial to USB converter. It is easy to interface and incorporate in user programs either in C++ or Delphi. The microcontroller is also used to interface the sensors like 3-axis accelerometer and gyroscopes and is able to provide the acceleration and angular velocity readings via the USB interface for the PC. There are also unused free pins on the board which may be used for connection of additional sensors or other hardware if the need arises. The whole microcontroller interface board can be seen in Fig. 8.

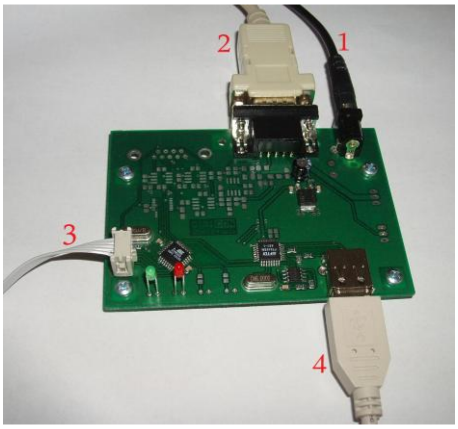


Fig. 8. Microcontroller board used as interface between the RC helicopter and computer. (1- Power connection; 2- connection with the transmitter; 3- sensors connection; 4- USB connection with PC)

3 FEEDBACK FROM HELICOPTER

To control the helicopter, a feedback is needed to detect the helicopter states. For measuring accelerations of the helicopter's body a triple axis accelerometer has been connected to the microcontroller interface board. The accelerometer measures accelerations in three axes orthogonal to each other. For detecting the heeling of the helicopter two single axis gyroscopes are employed and measure the angular velocities about the helicopter's longitudinal and lateral axes.

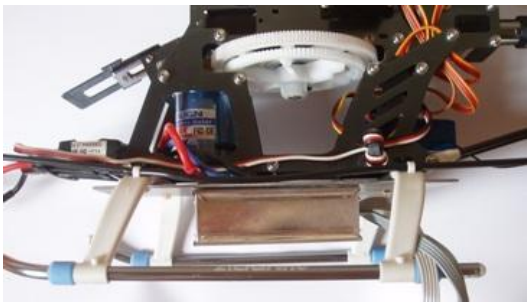


Fig. 9. Housing of the sensors attached to the helicopter.

These sensors were put into a thin metal plate box (see Fig. 9), which dispose of similar characteristics to a Faraday cage and bring the protection against spurious fields.

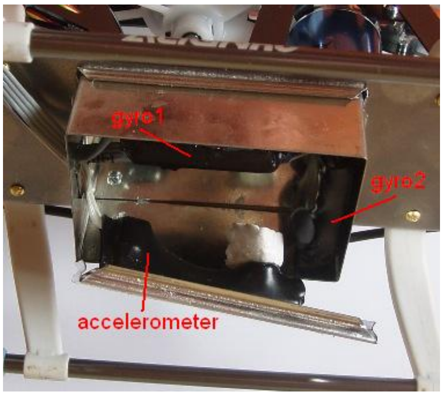


Fig. 10. Inside of the sensors housing.

The gyroscopes were mounted on the front side and the right side of the box such that they are orthogonal to each other (see Fig. 10) and measure the angular velocities about two axes. The accelerometer was mounted on the bottom side of the box (see Fig. 10).

3.1 *Triple axis accelerometer*

For our purposes, the triple axis accelerometer based on a LIS3LV02DQ chip (see Fig. 11) was chosen and connected to the microprocessor board through the I2C bus. The LIS3LV02DQ is a capacitive accelerometer chip. When the acceleration is applied to the sensor, the proof mass displaces from its nominal position, causing an imbalance in the capacitive half-bridge. This imbalance is measured using charge integration in response to a voltage pulse applied to the sense capacitor. The complete measurement chain is composed by a low-noise capacitive amplifier, which converts into an analog voltage the capacitive unbalanced voltage of the MEMS sensor and by three ΣΔ analog-to-digital converters, one for each axis, that translate the produced signal into a digital bit stream. The acceleration data may be accessed through an I2C/SPI interface, thus making the device particularly suitable for direct interfacing with a microcontroller (STMicroelectronics, 2005).

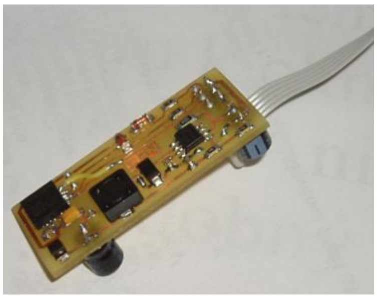


Fig. 11. Tripple axis accelerometer used for sensing accelerations of the helicopter’s body.

The accelerometer measures three types of accelerations: decomposition of the gravitational acceleration, accelerations caused by the translational displacement of the helicopter and accelerations caused by a centrifugal force when the helicopter is turning.

3.2 *Inadequacy of an accelerometer*

The accelerometer is mainly used for sensing the heeling of the helicopter. The principle is to measure the gravitational acceleration decomposed into the three orthogonal axes of the accelerometer, as can be seen in Fig. 12. From the components of the gravitational acceleration measured by the accelerometer, the longitudinal and lateral heeling of the helicopter can be estimated.

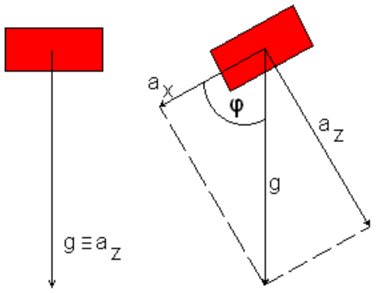


Fig. 12. Decomposition of the gravitational acceleration by the heeling of the acceleromete (mounted on the helicopter’s body).

As mentioned in Section III A, acceleration is measured by the displacement of the proof mass from its nominal position in the sensor. To depict this idea, a ball in a box is chosen to represent the accelerometer (Fig. 13). If the box heels the ball starts to move towards the direction of the heeling. This is, therefore, understood as an acceleration measurement in this direction, as shown in Fig. 13a. However, if the box is accelerating horizontally (and not heeling), the ball starts to move in to the opposite direction, which is also understood as an acceleration measurement but in the opposite direction, as shown in Fig. 13b. Clearly, this causes problems, for one cannot distinguish whether the helicopter heels or accelerates in the opposite direction. Therefore, the helicopter feedback has to be improved and is, for that reason, usually equipped with gyroscope sensors.

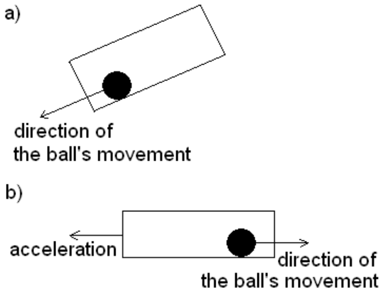

a)
direction of the ball's movement
b)
acceleration
direction of the ball's movement

Fig. 13. Simplified figure of acceleration measuring principle in one axis. a) Acceleration measured by the heeling of the accelerometer. b) Acceleration measured by the linear acceleration of the accelerometer.

3.3 *Piezoelectric gyroscopes*

Heeling of the helicopter can be measured with the gyroscope. Classical rotating mechanical gyroscopes are rather expensive and for our purposes even too heavy and big-sized.

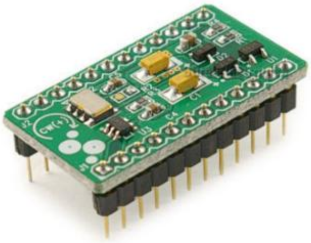


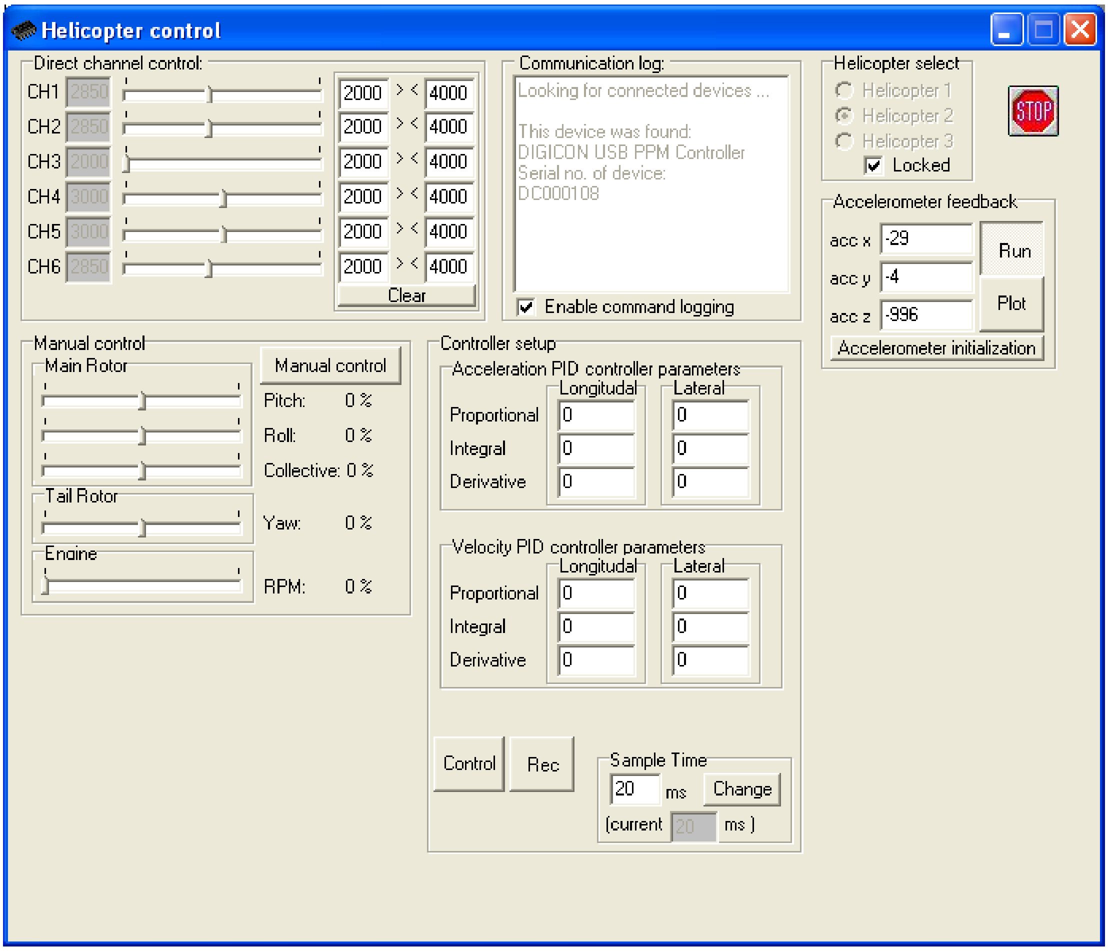
Fig. 14. Piezoelectric gyroscope with XV-3500CB chip used for sensing angular velocities of the helicopter's body.

Therefore, piezoelectric gyroscopes were chosen. Unlike the classical rotating mechanical gyroscope, the piezoelectric gyroscope does not measure the turn, but the angular velocity by the turning. The piezoelectric gyroscope consists of a vibrating piezoelectric material which tends to keep the vibrations in the same plane as its support is rotated. A Coriolis force can be measured to produce a signal related to the rate of rotation (Wikipedia, 2010).

For our purposes two angular rate piezoelectric gyroscope boards with XV-3500CB chip (see Fig. 14) were chosen and connected to the microprocessor board through the I2C bus, like the accelerometer. One of the gyroscopes is used for measuring the angular velocity about the longitudinal axis of the helicopter and the other for measuring the angular velocity about the lateral axis of the helicopter. The gyroscope board uses MCP3421 A/D converter chip for interfacing the I2C communication. This chip has a fixed address 1101000X (if "X" stands for 1, the AD conversion value and configuration bytes can be read, if "X" stands for 0, the configuration bytes writing would follow) (Sure electronics, 2008). Because the communication with two devices with the same address connected to the I2C bus would not be working, only one gyroscope was connected to the hardware I2C bus, used also for collecting the data from the accelerometer. For the second gyroscope, the new software I2C bus was created.

4 SOFTWARE FOR HELICOPTER CONTROL

The software for controlling the helicopter with the PC was programmed in BORLAND C++ Builder. The main GUI window of the software consists of



Helicopter control

Direct channel control:

CH1	2850	2000 >	< 4000
CH2	2850	2000 >	< 4000
CH3	2000	2000 >	< 4000
CH4	3000	2000 >	< 4000
CH5	3000	2000 >	< 4000
CH6	2850	2000 >	< 4000

Clear

Communication log:

Looking for connected devices ...

This device was found:
DIGICON USB PPM Controller
Serial no. of device:
DC000108

☑ Enable command logging

Helicopter select
○ Helicopter 1
● Helicopter 2
○ Helicopter 3
☑ Locked

STOP

Accelerometer feedback
acc x -29
acc y -4
acc z -996
Run
Plot
Accelerometer initialization

Manual control
Main Rotor
Tail Rotor
Engine
Manual control
Pitch: 0 %
Roll: 0 %
Collective: 0 %
Yaw: 0 %
RPM: 0 %

Controller setup

Acceleration PID controller parameters

	Longitudal	Lateral
Proportional	0	0
Integral	0	0
Derivative	0	0

Velocity PID controller parameters

	Longitudal	Lateral
Proportional	0	0
Integral	0	0
Derivative	0	0

Control
Rec

Sample Time
20 ms Change
(current 20 ms)

Fig. 15. Main window of the program.

several blocks with various functionalities, as can be seen in Fig. 15. The helicopter can have controlled each channel manually in a "Direct channel control" block. This can be done with moving 6 different track bars.

Longitudinal input (heeling of the helicopter forward/backward), lateral input (heeling of the helicopter to the sides) and collective input (ascent/descent of the helicopter) are controlled by the variable pitch of the helicopter's main rotor blades, executed by three servos. Hence, the control of these three channels has to be mixed such that the main rotor can be controlled correctly. Mixed control signals for controlling the direct helicopter inputs are controlled in the program with "Direct control" block. The software also offers sampling time modification, graphs for measured accelerations and angular velocities and basic PID controllers which serves as a first choice in experiments with the helicopter stabilization. For first experiments with helicopter stabilization, only feedback data from accelerometers were used.

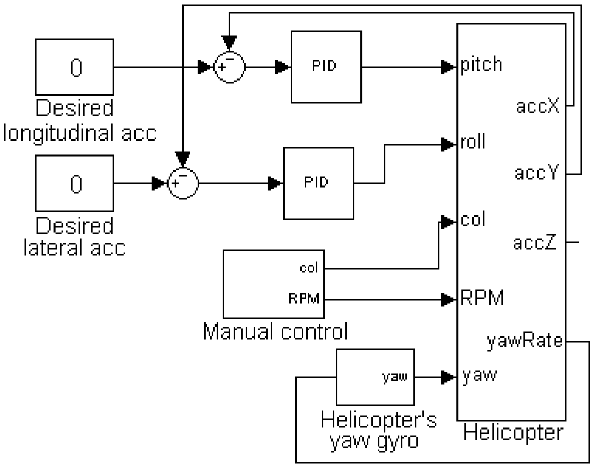


Fig. 16. Helicopter stabilization scheme using feedback data from accelerometer.

The aim was to keep longitudinal and lateral accelerations equal zero. Computer controlled pitch and roll of the helicopter, while the operator has a manual control of rotor RPM and collective angle of attack for controlling ascent/descent of the helicopter. RC helicopters are equipped with yaw gyroscope, which can hold the yaw position to make the control of the helicopter simpler. This yaw control was exploited to simplify the stabilization. The scheme used for this approach can be seen in Fig. 16. Values of PID controllers for longitudinal and lateral stabilizations can be set in "Acceleration PID controller parameters" block of the program (see Fig. 15).

Another approach for stabilizing the helicopter, which can be seen in Fig. 17., was to use feedback data from gyroscopes. Computer controls pitch and roll of the helicopter so that pitch and roll rates measured by the gyroscopes were zero. For this approach the operator also has manual control of the helicopter's ascent/descent and the yaw control is kept on the helicopter's gyro

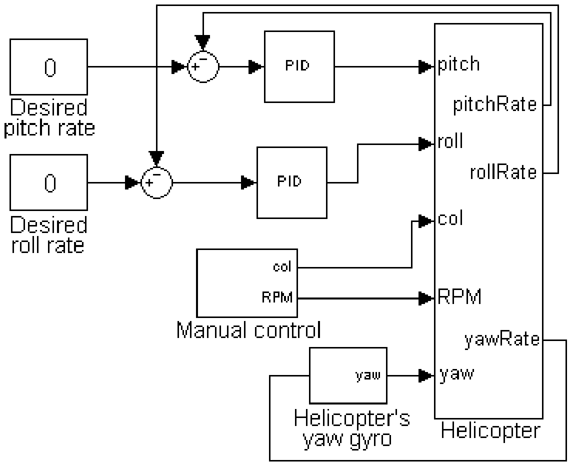


Fig. 17. Helicopter stabilization scheme using feedback data from accelerometer.

5 CONCLUSION

The aim of this work was to describe the developed interface between the helicopter and PC, which controls the helicopter and collects the feedback data. The PC sends through the USB desired values for all channels to the developed microcontroller interface which creates, based on these values, the corresponding PPM signal and sends it to the transmitter. The transmitter, which is set to the "trainer" mode, forwards the signals to the receiver and controls the RC helicopter. The helicopter has a triple-axis accelerometer and two piezoelectric gyroscopes ensuring the feedback from the helicopter. The sensors are connected through the I2C to the microcontroller interface board which collects the feedback data from sensors and sends them back to the computer through the USB. Evaluation of the feedback signals and creation of the control signals is treated by the PC. For this purpose, the software for controlling the helicopter manually and stabilization based on a PID controller was developed. However, due to the delays caused by the feedback data evaluation and processing, the stabilization did not work properly. By demand of the real time control, our next task is to develop software in Linux environment. In addition, the feedback data evaluation is to be improved. From that point of view, the complementary filter, described in (Rodina, et al., 2009), seems to be appropriate. As a control algorithm, the LQ controller described in (Kozakova, 2008) and, for example, used for the helicopter stabilization in (Andersen, et al., 2008), seems to be convenient.

ACKNOWLEDGMENTS

The work was supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

6 REFERENCES

Andersen, T. L., D. F. Lauritzen, J. T. Madsen, M. M. Sørensen, B. A. Mertz, Autonomous Inverted Hover of a Small Scale Helicopter, Report, Institute of Electronic Systems, Aalborg University, 2008.

Cai. G., L. Feng, B. M. Chen, T. H. Lee, Systematic design methodology and construction of UAV helicopters, Mechatronics, vol. 18, 2008, pp 545–558.

Fogh, M., T. H. Mikkelsen, E. Mustafic, R. Pettersen, J. M. Sommerset, Autonomous Helicopter, Report, Institute of Electronic Systems, Aalborg University, 2004.

Hald, U. B., M. V. Hesselbæk, M. Siegumfeldt, Nonlinear Modeling and Optimal Control of a Miniature Autonomous Helicopter, Diploma Thesis, Institute of Electronic Systems, Aalborg University, 2006.

Hald, U. B., M. V. Hesselbæk, J. T. Holmgaard, Ch. S. Jensen, S. L. Jakobsen, M. Siegumfeldt, Autonomous Helicopter - Modelling and Control, Report, Institute of Electronic Systems, Aalborg University, 2005.

Kozakova, A., Design of discrete-time compensator for reference tracking with disturbance rejection, Transactions of the VSB, in press, 2008.

Mettler, B., Identification Modelling and Characteristics of Miniature Rotorcraft, Kluwer Academic Publishers, 2003.

Pestun, I., Helicopter control, Diploma Thesis (in Slovak), Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology, 2009.

Rodina, J., P. Hubinsky, “Stability control of segway like differential drive by using MEMS sensors”, Conference on Modeling of the Mechanical and Mechatronic Systems, Zemplinska Sírava, Slovakia, pp. 483-487, 2009.

Vastianos, G., 16 channel servo controller for robotic applications, Diploma thesis, Faculty of Technological Applications, Technological Educational Institute of Piraeus,.

Net-Scale Technologies, Inc., Autonomous Off-Road Vehicle Control, Final Technical Report, 2004.

STMicroelectronics, LIS3LV02DQ MEMS inertial sensor, datasheet, 2005.

Sure electronics, Angular-Rate Sensor XV-3500CB Prototype PCB User’s Guide, 2008.

The Model Electronics Company, The PPM Radio Control System: Part 2, Technical Note MTN004.

http://en.wikipedia.org/wiki/Vibrating_structure_gyroscope , 2010.

3D Off-Line Path Planning for Autonomous Airships in Restricted Known Environments.

Naef Al-Rashedi and Michael Gerke

Fakultät für Mathematik und Informatik
Lehrgebiet Prozesssteuerung und Regelungstechnik
Fernuniversität in Hagen
Universitätstrasse 27, D-58097 Hagen.
Tel: +49 23 31 / 9 87 – 17 02
Fax: +49 23 31 /9 87 – 3 54
Naef.Al-Rashedi@FernUni-Hagen.de

Abstract: The paper presents a Genetic Algorithm (G.A.) for off-line path planning of Autonomous Small Airships in known 3D environments with special consideration of restricted areas. The algorithm assumes that the airship is used in Fire Fighting and Mine Detection projects, so the aircraft will fly only a few meters above the ground, which means there is a high possibility of collision with obstacles. The task of the Off-Line Path Planner algorithm is to find an optimal route to visit all the predefined locations for airborne measurement exactly once per mission, without any collisions with environmental obstacles and to avoid fly over a defined restricted area. The planner task posed here is an NP problem. This paper proposes a 3D Off-Line Path Planner using G.A. including chromosome representation, G.A. crossover and collision avoidance with known obstacles. The proposed algorithm is implemented using MATLAB with Genetic Algorithms and Mapping Toolboxes. The proposed algorithm is tested using real maps of our research airfield and the result shows that the algorithm finds a near-optimal collision free path for the airship.

1. Introduction

Autonomous Small Airships have many applications in civil and military areas[3], e.g. for different missions including fire fighting, mine detection, traffic surveillance, maintenance of high power electric lines and advertising. The main advantages of using an airship are: it has low energy consumption, there is no vital risk for operators during performance of hazardous missions, and its long endurance in the air.

The airship need a control system that can make both low-level control decisions in real-time, medium-level decisions such as path planning, and high-level decisions, such as cooperative task assignment, for long time missions without human interference. Task assignment is crucial for designing successful missions in difficult environments while the path planners which generate collision-free and optimized paths are needed to give autonomous operation capability to the airship. The combined solution of both aspects for the mission planning problem leads to a near optimal flight trajectory [6].

Figure (1): Fernuniversität Airship

The paper is organized as follows. Section 2 formulates the path planning problem. The description of the proposed path planner follows in section 3. In section 4 experiments and results are given. The conclusion of this paper is drawn in section 5.

2. Problem Definition

The mobile robot path planning problem is typically formulated as follows: given a mobile robot and a description of an environment and set of user-defined way-points (wp), the problem is to calculate a route that visits all user-defined way-points exactly once. The resulting path should be free of collision and satisfy certain optimization criteria (i.e., shortest path)[3]. This problem corresponds to finding the shortest Hamiltonian cycle in a complete graph G = (V, E) of an ***n*** nodes. Thus the path planner consists of finding a permutation of the set $\{wp_1,wp_2,wp_3,\ldots,wp_N\}$ that minimize the quantity:

$$\text{Minimize} \quad \left[\sum_{i=1}^{N-1} d\left(wp_i, wp_{i+1}\right) + d\left(wp_N, wp_1\right)\right]$$

Where d(wp$_{i,}$ wp$_j$) denotes the distance between waypoint wp$_i$ and waypoint wp$_j$.

Researchers distinguish between various methods used to solve the path planning problem according to two factors, (1) the environment type (i.e., static or dynamic), (2) the path planning algorithms (i.e., Off-Line or On-Line). A static environment is defined as the environment which doesn't contain any moving objects other than a navigating robot; while any dynamic environment includes moving objects (i.e., human beings, vehicles and other robots).

The Off-Line path planning algorithm requires a complete knowledge about the search environment and is based on the fact that all terrain should be static. On the other hand, On-Line path planning means that the path planning algorithm is calculated in real-time while the robot is moving around. In other words, the algorithm is capable of producing a new path in response to environmental changes [2]. Many studies try to solve the path planning problem for mobile robotics by using evolutionary approaches like genetic algorithms [1,5,7,10], but most of them solve the problem as how to go from one location to another and this study try to find an optimal trajectory to visit many locations exactly once with respect of collision avoidance, also this study is different from [9] by introducing the existence of user-defined restricted area that make the problem of path planning more complicated.

3. The Proposed Path Planner

The proposed path planner accomplishes its task into two phases: the preparation phase and the G.A.

3.1. The Preparation Phase

In this phase all inputs data such as digital 3D maps and user-defined way-points are represented and prepared to be use by the genetic algorithm.

3.1.1 Environment Representation

This paper considers that the airship will fly in static and well known environments, while these environments are represented as digital maps (e.g. for processing in MATLAB Mapping Tool Box).

3.1.2 Restricted Area Avoidance

The procedure starts with checking if the direct (straight line) path between any two of the User-Defined locations is crossing a defined restricted area or not. If it is crossing a restricted area then, a process starts to find new subway points to avoid this area. There are only two possible ways to avoid the restricted area, one is to go left way side and the other is to go right way, the decision of where to go (left or right) is taken according which one is the short. The technique used to determine the short way to avoid the restricted area can be described as the following: suppose that (UDP$_i$, UDP$_j$) are two User-Define points, and {v$_1$,v$_2$,v$_3$,v$_4$,v$_5$} are set of vertices of a restricted area as shown in figure(3). The direct path from UDP$_i$ to UDP$_j$ is crossing the restricted area and divide its vertices into two sets: the vertices {v$_1$,v$_2$} that are located to the right of direct path line, and the vertices {v$_3$,v$_4$,v$_5$} that are located to the left. The alternative path to avoid the restricted area in this example can be defined as:

Minimum(|(UDP$_i$,v$_1$)+(v$_1$,v$_2$)+(v$_2$,UDP$_j$)|, |(UDP$_i$,v$_3$)+(v$_3$,v$_4$)+(v$_4$,v$_5$)+(v$_5$,UDP$_j$)|)

It is clear from figure(3) that |(UDP$_i$,v$_1$)+(v$_1$,v$_2$)+(v$_2$,UDP$_j$)| is the minimum in length so, the vertices {v$_1$,v$_2$} are added as new subway points into the path between (UDP$_i$ and UDP$_j$) to avoid the restricted area. Another issue in restricted area avoidance is how to determine the elevation of the new subway points? The elevations of UDP$_i$ and UDP$_j$ are defined by user and the elevation of the new subway points can be defined as:

Maximum (eUDP$_i$, eUDP$_j$, eh)

Where: eUDP$_i$,eUDP$_j$ are the elevations of UDP$_i$,UDP$_j$ in order, and **eh** is the elevation of the highest point of the terrain in the path [UDP$_i$,v$_1$,v$_2$,UDP$_j$] .

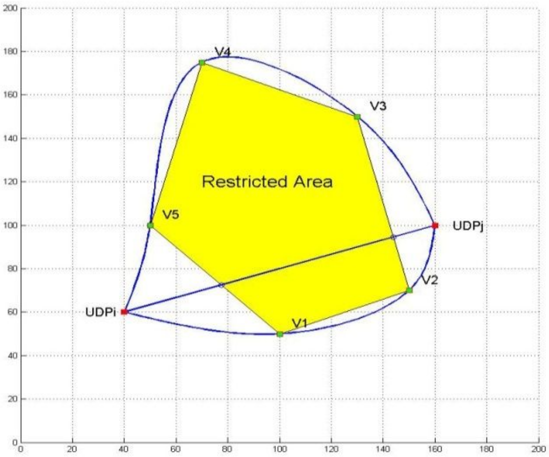
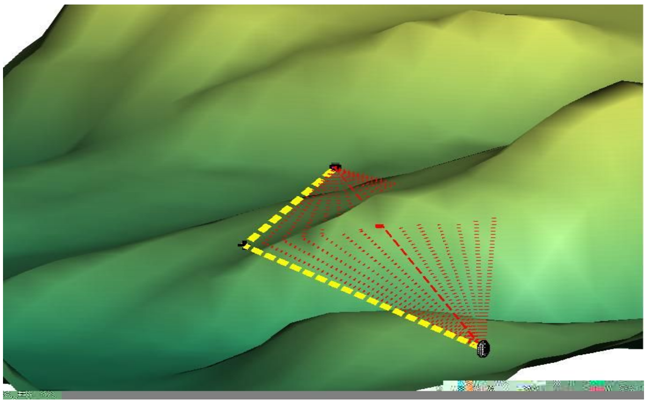


Figure (3): Example of restriced area avoidence

3.1.3. Verifying Feasibility and Adding New Subway Points

The Feasibility of all direct paths between each set of user-defined way-point are verified and if there is a feasible direct path between any pair of these points, its path length is calculated and stored in a cost table, otherwise (i.e. there is an obstacles between two waypoints), a process is being started to find an indirect feasible path between these two user-defined way-points by adding a new subway point



Figure(3) Example of obstacles avoidance to avoid this disturbing obstacle. The strategy of adding new subway points is based on making a displacement around the obstacle in all directions in order to find an intermediate way-point that is feasible from both user-defined way-points under consideration. The main advantage of this step is to reduce the calculation and the execution time of G.A.

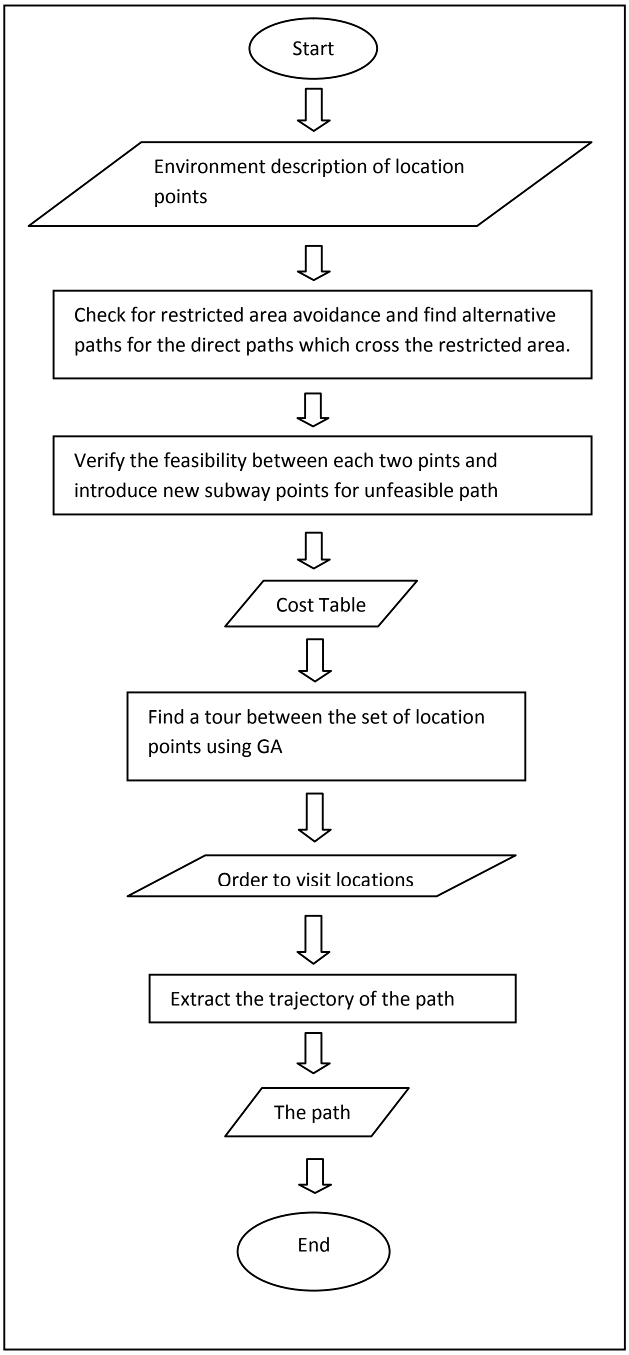

Start
Environment description of location points
Check for restricted area avoidance and find alternative paths for the direct paths which cross the restricted area.
Verify the feasibility between each two pints and introduce new subway points for unfeasible path
Cost Table
Find a tour between the set of location points using GA
Order to visit locations
Extract the trajectory of the path
The path
End

Figure (4): Path Planner Process Flowchart

3.2. The G.A. Algorithm:

This new pre-processing strategy enhances the G.A. search that is used to solve the travel salesman problem in [8] to be used as an optimization tool for the airship path planner. A description of the G.A. search for 3D-problems is as follows:

3.2.1 Path Representation:

The proposed algorithm represents a robot mission as an Upper Triangle Binary Matrix (UTBM) as in **Figure (5)** which represents the tour (1, 3, 5, 2, 4, 6). Every chromosome is represented in binary form, which means: if the element (i,j) of the matrix is equal (1) there is a feasible path between user-defined way-points (i) and (j) in the tour.

$$
\begin{array}{c|c|c|c|c|c|}
 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 0 & 1 \\
\hline
 & 2 & 0 & 1 & 1 & 0 \\
\cline{3-6}
 & & 3 & 0 & 1 & 0 \\
\cline{4-6}
 & & & 4 & 0 & 1 \\
\cline{5-6}
 & & & & 5 & 0 \\
\cline{6-6}
\end{array}
$$

Figure (5): The Proposed Representation of a Chromosome

The matrix representation must satisfy the following conditions in order to represent a feasible tour:

1- The number of elements in the matrix that have the value (1) must be equal to the number of the user-defined way-points.

$$\sum_{i=0}^{N-2} \sum_{j=1}^{N-1} wp_{ij} = N$$

2- The number of matrix elements that have the value of (1) in each row and each column of the same way-point must be equal to two.

$$Col_j = \sum_{i=0}^{N-2} c_{ij}$$

$$Row_i = \sum_{j=1}^{N-1} c_{ij}$$

$$\forall (i = j) : Row_i + Col_j = 2$$

3.2.2. The Crossover

The crossover operation generates an offspring from two parents as follows:

1- Unify the two parents matrices in one matrix by executing (Or) operation.

2- The result matrix from step 1 may be an invalid tour (do not satisfy the two conditions from above). So it must be repaired by counting the number of edges of each way point. If any way point has more than two edges, then keep the shortest two edges and delete the others. The way points that have less than two edges are added to a list of disconnected way points.

3- Adding the missing edges to disconnected way points by using the greedy algorithm.

3.2.3. The Evaluation Function:

The evaluation function is an important part of any evolutionary process using G.A. Only an appropriate selection of the evaluation function will lead the search towards the desired optimal solution. For the airship that is used in complex flight missions, the optimal path has two type of constrains:

- Hard constrains: the path should be free of collision and every user-defined way- point of the flight mission should be visited exactly once. (The algorithm proposed here guarantees that all the solutions that are produced meet these conditions).
- Soft constrains which can be defined by user requirements: in specific scenarios such as fire fighting and mine detection some additional constrains can be formulated, e.g. enforcement of a shortest path and of path types keeping the airship in low altitude above the ground. The following equation is used as evaluation function for the algorithm proposed here:

$$F = \text{minimize } [\ |T| + H\ ]$$

Where:

Parameter **|T|** is the length of the flight mission [e.g. in meters] and is computed as the sum of the paths lengths between every two way-points of the tour.

Parameter **H** is a value computed as following:

$$H = h * Hf$$

Where:

h is the sum of the difference between the maximum height of the airship during the tour and the height of the user-defined way-points.

Hf is a user defined value; for a large value the GA will select the solution that avoids the airship to go up even if the tour will be longer.

4. Experiments and Results

The proposed algorithm is implemented using MATLAB and tested both in a real three-dimensional map of our experimental area in Hemer (Northrhine-Westfalia), and sample Matlab DEM files like Geoid and South San Francisco. The following figures are showing the results of the experiments.

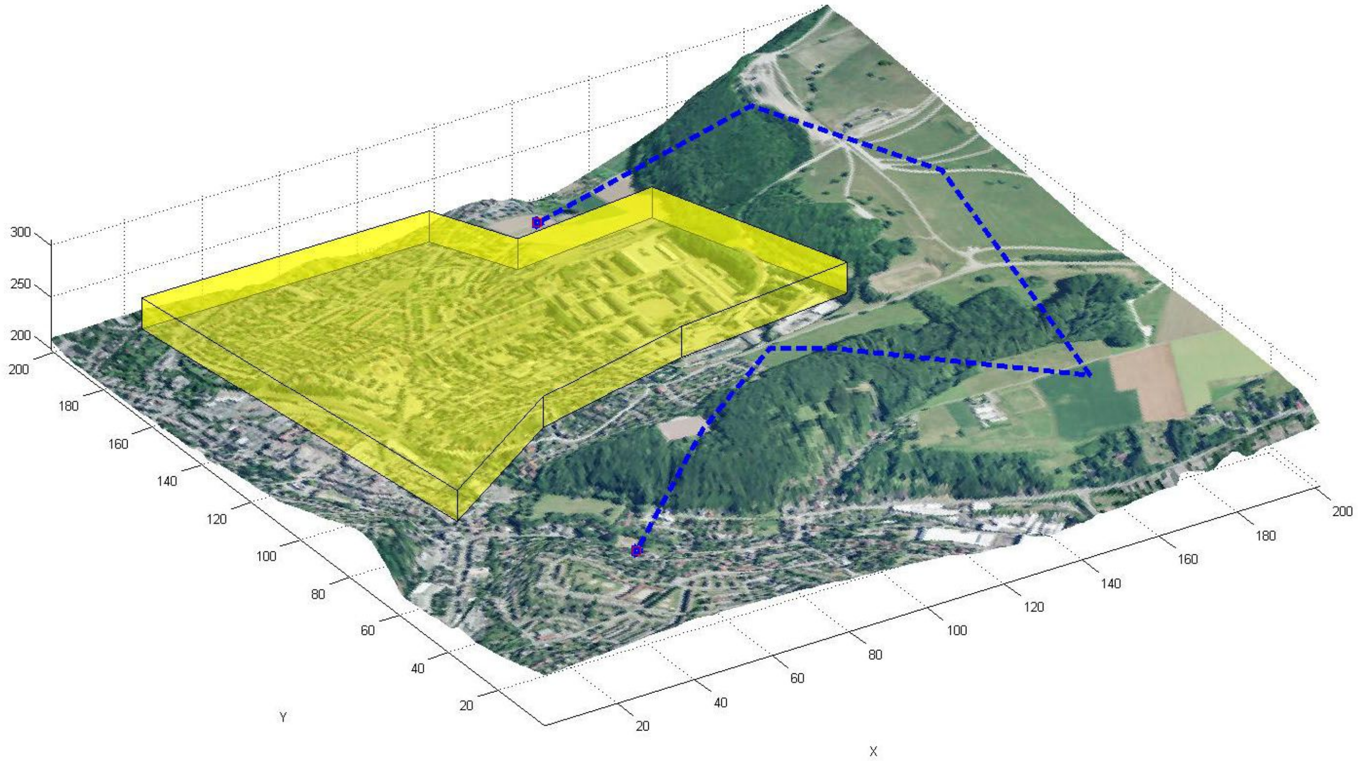


Figure (6): Sample tour for 6 user defined way-points in map of Hemer.

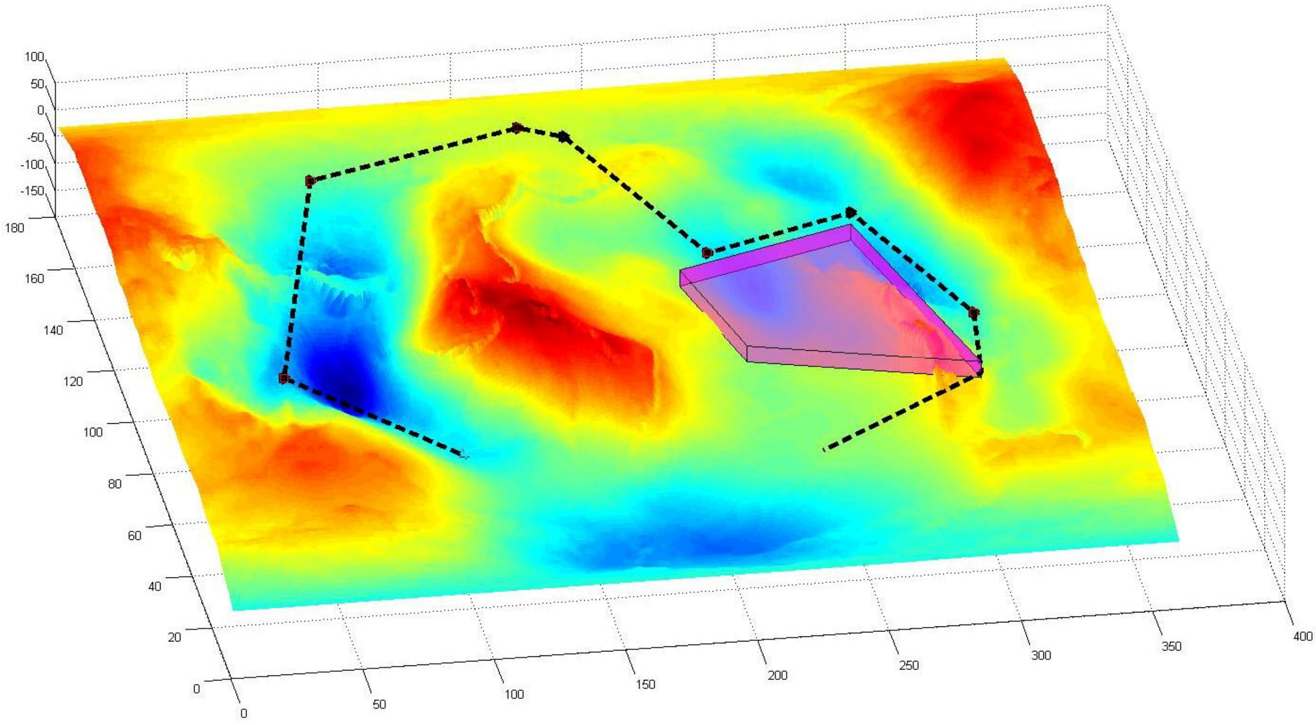


Figure (7): A sample tour of 7 user defined way-points in map Geoid

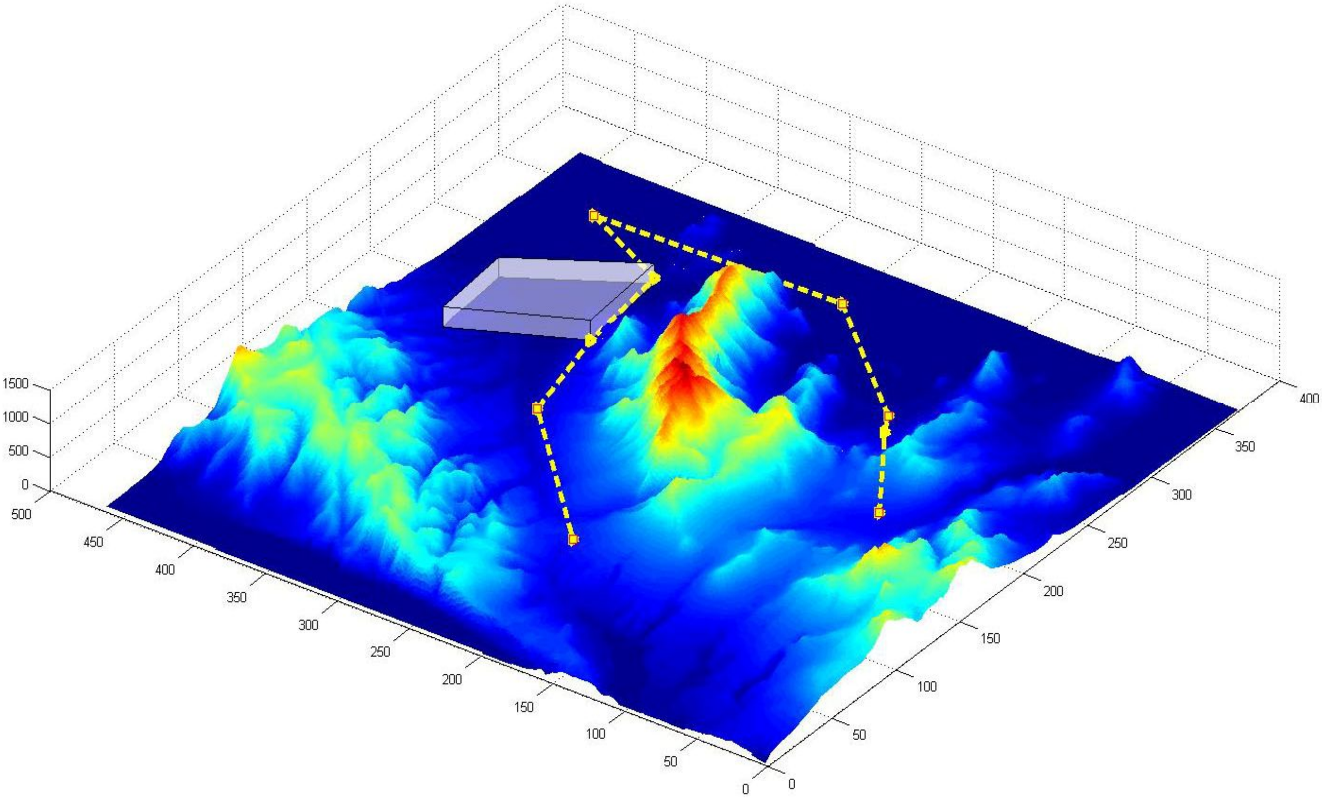


Figure (8): Sample tour of 6 User-Define points in South San Francisco map.

5. Conclusion:

This paper represents an Off-Line path planner for small autonomously operating airships in restricted but well known static 3D environments. Using G.A. search, the separation between Preparation phase and the G.A. process reduces the execution time of G.A. and makes the path planner more flexible. For example, it is relatively simple to modify the proposed G.A., or to integrate optimization criteria considering further factors of influence (like the direction of wind force, the energy consumed by airship devices …etc.). Another advantage of using GA as problem solver is, that in case of any changes in the environment (e.g. the boundaries of the restricted area are change.) the last solution of the GA planner is still important and it can be used as initial solution for an updated search procedure. This speeds-up the calculation of an optimal solution with respect to recent changes in the environment and it makes the GA path planner more efficient for on-line operation during the airship mission in a dynamically changing environment.

6. References

[1]. Anargyros N. Kostaras, Ioannis K. Nikolos, Nikos C. Tsourveloudis, and Kimon P.Valavanis: *Evolutionary Algorithm Based Off-Line/On-Line Path Planner for UAV Navigation*. Proceedings of the 10th Mediterranean Conference on Control and Automation – MED 2002 Lisbon, Portugal, July 9-12. 2002.

[2]. Bruno Siciliano, Oussama Khatib, Frans Groen: *Robot Navigation from Nature Simultaneous Localisation, Mapping, and Path Planning Based on Hippocampal Models.* Springer Tracts in Advanced Robotics Volume 41 ISSN 1610-7438. Springer-Verlag Berlin Heidelberg.2008

[3]. Christopher Bolkcom, *Potential Military Use of Airships and Aerostats*, CRS Report for Congress, USA, 2004

[4]. Florian Adolf, Franz Andert, Sven Lorenz, Lukas Goormann, and Jörg Dittrich: *An Unmanned Helicopter for Autonomous Flights in Urban Terrain.* German Workshop on Robotics GWR2009, Braunschwrig-Germany, 2009.

[5]. Francois C.J. Allaire, Mohamed Tarbouchi, Gilles Labonte and Giovanni Fusina: *FPGA Implementation of Genetic Algorithm for UAV Real-Time Path Planning.* Springer Science. 2008

[6]. Isil Hasirciolgu Haluk Rahmi Murat Ermis: *3-D Path Planning for the Navigation of Unmanned Aerial Vehicles by Using Evolutionary Algorithms*, GECCO’08, Atlanta-Georgia USA, 2008

[7]. Ismail Al-Taharwa, Alaa Sheta and Mohammed Al-Weshah: *A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment*, Journal of Computer Science 4 (4): 341-344, ISSN 1549-3636, Science Publication, 2008.

[8]. Naef Taher Al-Rashedi and Jalal Atoum: *Solving Travel salesman Problem Using New Operator in Genetic Algorithms*, American Journals of Applied Sciences 6 (8): 1586-1590, 2009.

[9]. Naef Al-Rashedi and Michael Gerke, *3D Off-Line Path Planning for Autonomous Airships in Known Environments by using Genetic Algorithms.* Proceeding 20. Workshop Computational Intelligence, Dortmund Germany, 1-3 December 2010.

[10]. Zixing, CAI., Lingli, YU., Chang, XIAO., Lijue, LIU.: *Path Planning for Mobile Robots in Irregular Environment Using Immune Evolutionary Algorithm*. Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008.

Data Management Architecture for Tele-operated UAV System

P. Bahnik* J. Pilka**

* Control Systems Engineering Group, Department of Mathematics and Informatics, FernUniversität in Hagen, Germany
(Tel: +4923319871118; e-mail: pavol.bahnik@fernuni-hagen.de)
** Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology
(e-mail: xpilkaj@stuba.sk)

Abstract: Nowadays, more frequently than ever, the unmanned aerial vehicles (UAVs) are used effectively as mobile sensor platforms. The UAV system equipped with an airborne camera and special sensors is a valuable source of various important information helping to build an actual overview of an environment. It can take place like an observer in disaster situations as well as a special mobile monitoring device which is able to collect required data from a predefined area. This paper introduces our approach to design effective data management architecture to be able to manage, reliably distribute and represent different types of measured data with taking many aspects and limitations of the tele-operated UAV system to consideration.

1. INTRODUCTION

In the last few years various UAV systems became very popular as an effective platform to observe particular areas and collect the data using specific sensors. The UAVs enable us to obtain a bird's eye view of the environment, having access to areas, where often only incomplete and inconsistent information is available. To get actual and precise information or data from a desired place is important in many situations of a modern crisis management or plenty of inspection or data acquisition tasks. Proper utilization of collected data usually varies according to the purpose of the required mission of the UAV system. The main task of the mission normally forces to use specific, narrow focused type of sensors that need to be carried by the UAV. For this reason we decide to develop a modular, service based data management architecture, which can provide all desired functionality to complete the mission successfully and enable the operator to control the UAV system reliably and comfortably.

Management and distribution of data provide an interesting field of research in different domains, ranging from hardware architecture over communication and network architecture, resource awareness to categorization, deployment, flow control and representation of data.

This paper is organized as follows. In the first part the overview and the desired functionality of our system is described. The main limitations which need to be taken to consideration by design are presented as well. In the second part the system architecture as our base hardware (HW) and software (SW) platform is detailed and the engineering tradeoffs considered by specification are mentioned. The categorization and deployment of data measured by the airborne sensors is discussed in the third section. The fourth part explains the main data flow control between the HW/SW modules itself and the base station and shows the process of the representation of data. In the last part our experiences with measuring data by using a laser scanner as a new additional on-board sensor are described.

2. OVERVIEW OF OUR UAV SYSTEM FUNCTIONALITY AND LIMITATIONS

At our department we are focused on the development of an autonomous flying airship. We use a robotic airship (BLIMP) filled with helium which is 9 meters long with maximal 2.5 meters diameter and its payload is about 5 to 6 kilograms.

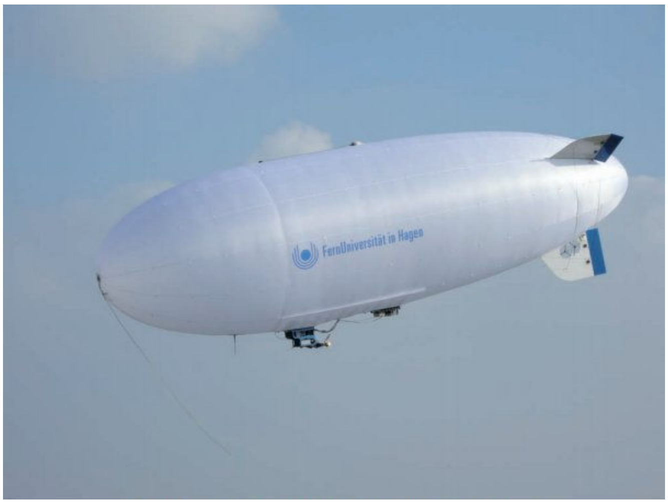


Fig. 1. Robotic Airship during a field test in Hemer, Germany, 2009

The required task of the mission has a big influence on the desired functionality of an UAV system. The functionality of an UAV system and its data management architecture from various aspects could be split into several levels.

The first and base level of the main functionality of an UAV system and its data management architecture is of course to provide the possibility to be controlled by the operator during the flight. This requirement is closely coupled with communication capabilities and the on-board autonomy of the system. Depending on the increasing on-board autonomy there appear three possible control modes.

- Remote Control
- Teleoperation
- Automatic mode

In the remote control mode the data management architecture provides to transmit primer and necessary data or commands from the operator to UAV's actuator system. In this case the level of autonomy is very low and the operator has to control the UAV manually using a remote control. This flying mode delivers many constraints depending on a potential application of the UAV. In our case one of the main limitations is that controlling the airship manually via remote control is a quiet difficult task for the pilot. It assumes to have some skills to keep the airship flying smoothly by rough weather conditions. Moreover the airship has to be visible for the pilot permanently during the flight. On the other hand this type of control is useful for some maneuvers which could be problematic for the autonomous control algorithms as well as a very important backup control system by any unexpected failure of the on-board systems.

The teleoperation mode increases the on-board autonomy into a semi-autonomous control, which means that the operator is able to control the UAV via joystick with support of basic automatic control algorithms to assure the desired course and altitude during the flight. The operator doesn't need to promptly react on each disturbance appeared during the flight and doesn't need to keep a visual contact with the UAV as well. It effects also the data management architecture, because it is necessary to deliver and intuitively represent an information of the actual position and many additional information from UAV's sensor system, which can give the operator feedback and a better overview of the environment. In this mode it is very important to provide and keep reliable communication with low latency between the UAV and the operator, because the system is usually not able to make any decision itself. In fact many UAV systems are working properly on semi-autonomous mode of control, because it enables them to complete many types of missions successfully.

The full automatic mode is the highest level of autonomy. The autonomy of an UAV relieves the operator of controlling the UAV and enables him to concentrate just on the main task of the mission. The operator just pickups the desired points to fly, the system calculates a flight path and flies it over. In this mode the control system is enhanced with a supervisory control and decision algorithms with support of various integrated modules like a path planner and collision avoiding algorithms. Although all information, which give a feedback for the operator like an actual position, are still important, this level of autonomy allows the system to make some preprogrammed decision by itself. With the support of a more advanced on-board intelligence, a short communication delays or disconnections between the UAV and base station communication does not necessary lead to any critical situation or damage of the UAV. Such a sophisticated control system enables the UAV to complete also some kind of missions successfully, where the capability of low latency communication is limited or restricted. On the other hand, in this mode various HW and SW modules are coupled together. These modules normally depend on each other and for the correct functionality of the whole system the effective exchange of actual data or the event messages between these collaborative modules is one of the major challenges for the data management architecture.

The second level of the functionality of an UAV system and its data management architecture is to enable the whole system to fulfill the required task of the mission successfully. As we mentioned in the introduction part of this paper, generally the tasks could be focused on monitoring some areas like an observer or any data acquisition tasks. For the data management architecture it usually means that various additional data from third-party sensor systems need to be transferred online to some operation centre or collected to be post-processed and analyzed after the mission. Our department was a member of two research projects: *International mine detection and removal (iMR)* and *International forest fire combat (iWBB)*. In the first project the role of our airship was to be used as an inspection vehicle to ensure the total destruction of mines by a high-energy laser system. Therefore, it was equipped with a remote camera system (Gerke 2009). This camera system is able to stream the video signal online to the base station and even to remote control the camera's viewpoint by the operator.

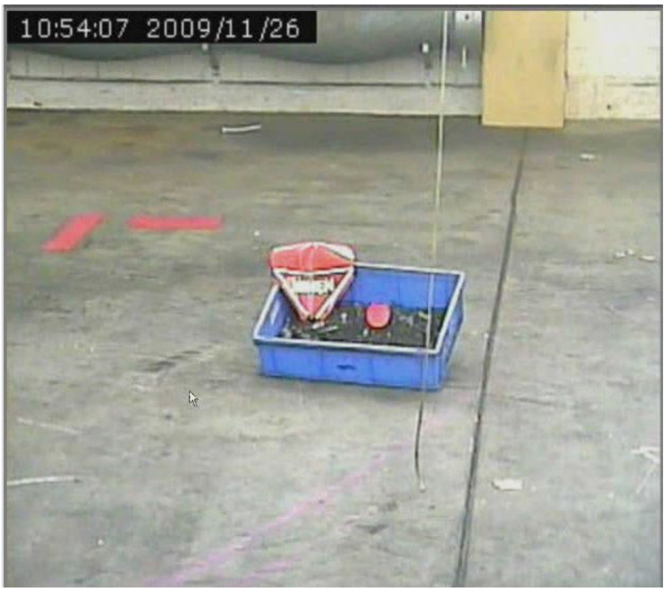


Fig. 2. The view on a destroyed dummy landmine which is captured by the on-board camera system during a test flight in Hemer, Germany, 2009

In the second project our airship was used as a mobile sensor platform for the third-party heat, smoke, gas detection and pollution monitoring sensoric systems. Another important aspect of our research was the establishment of redundant communication links between the airship and the base station, and to create a data interface to the project operation centre (Gerke 2009).

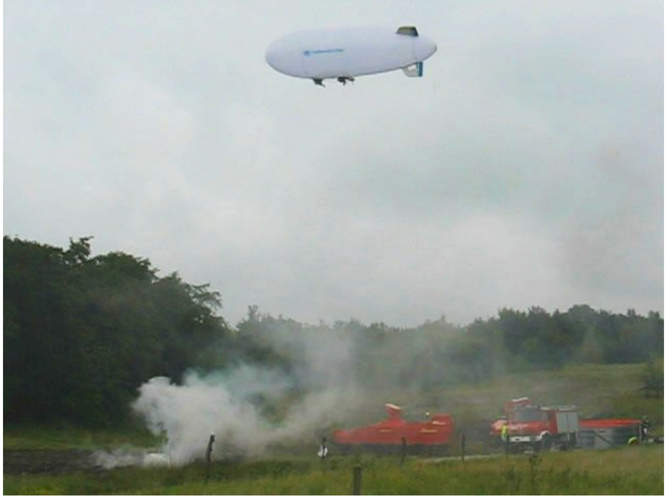


Fig. 3. The detection and monitoring of fire using the third-party sensoric system during the test flight in Hemer, Germany, 2010

The airship equipped with such a sensoric system is a valuable source of information about the actual situation for the firemen. As you can see in Figure 3 the airship was measuring data using these sensors to detect and monitor the fire and these data were online transferred to the project operation centre using the communication link of the airship's data management architecture.

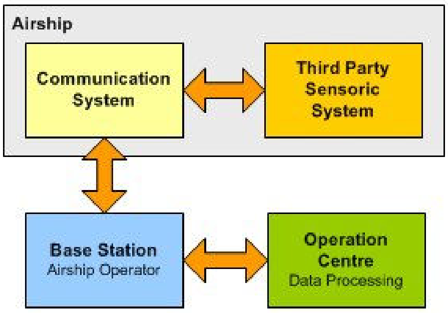


Fig. 4. Schematic of the data flow between the airborne sensoric system and the operation and data processing centre on the ground.

3. SYSTEM ARCHITECTURE AND ENGINEERING TRADEOFFS

During the design of our system architecture some limitations have to be taken to consideration. Our airship is powered by electric energy. The source of this power is just the battery system of the airship. Minimizing the power consumption of all embedded systems is a strict requirement. In order to save the energy that could be used to power the airship actuators and to maximize the time of flight as much as possible. The second very important limitation is the restricted payload of the airship.

The primary components of the hardware platform consist of a few modules. The first module represents an appropriate pre-designed control board with a DSP processor to handle a low level control of the airship in real-time. It has very low power consumption and it is optimized to minimal size and weight. Moreover it enables a rapid prototyping using the Matlab development environment. It includes the navigation system, which couples various sensors like GPS and an inertial measurement unit (IMU), which uses a combination of accelerometers and gyroscopes. This board is the main source of telemetric data for the operator as well as the superior system to process his commands. It is the most important node in the data management architecture.

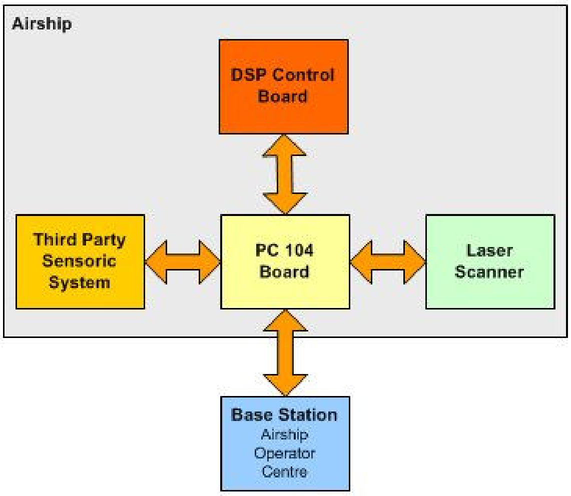


Fig. 5. The primary components of the hardware platform.

The next module is the embedded PC 104 board, which is dedicated to manage the data flow control and reliable distribution of data between all the collaborative modules and the base station on the ground. It is based on the Intel x86 processor architecture and uses a 500 MHz AMD Geode processor with 1 GB of RAM. The system uses a 32 GB compact flash card (CF) for storage and has many HW interfaces like dual Ethernet, RS232 or USB ports, to be able to connect additional third-party sensors or devices. This platform is a balanced trade-off between the needed resources and power consumption. Nowadays mostly expanding boards are based on Intel Atom platform with more resource capability, but also coupled with higher power consumption, which can be in some cases 4 times more like our board. For the purpose of a data flow manager, communication router and data collector we rather took the less powerful variant.

The base station, as an airship operator centre, is located on the ground. This module consists of a powerful mobile workstation PC equipped with pre-developed SW modules to keep the reliable communication between the airship and the base station and a special control panel module for the airship.

Our last tested additional module is the laser scanner UTM-30LX produced by Hokuyo Automatic CO., LTD. It is a relatively small type of a laser scanner. It weights just about 210 grams and its power consumption is less than 8 W, so it is an ideal device for a middle-sized UAV like our airship. It is intented to be used for the purpose of making 3D scans of the environment the airship is flying over and for obstacles detection as an information source for the collision avoiding algorithms.

4. CATEGORIZATION AND DEPLOYMENT OF DATA

By design of data management and communication architecture for tele-operated UAV system it is very important to specify all the data sources. The main source of data is of course the navigation system of the airship which delivers information of the actual position, altitude, orientation and velocities of the airship. As a second source of information and event messages the control panel module is located on the base station. It fires control commands and event messages from the operator to the airship's on-board control system. The next source of valuable information is the battery and actuator system status observer that gives an information of the actual battery capacity, voltage, current and propellers rpm. Other sources of data are also the laser scanner system and third-party sensoric systems, which usually use our communication platform just to transfer or collect measured data.

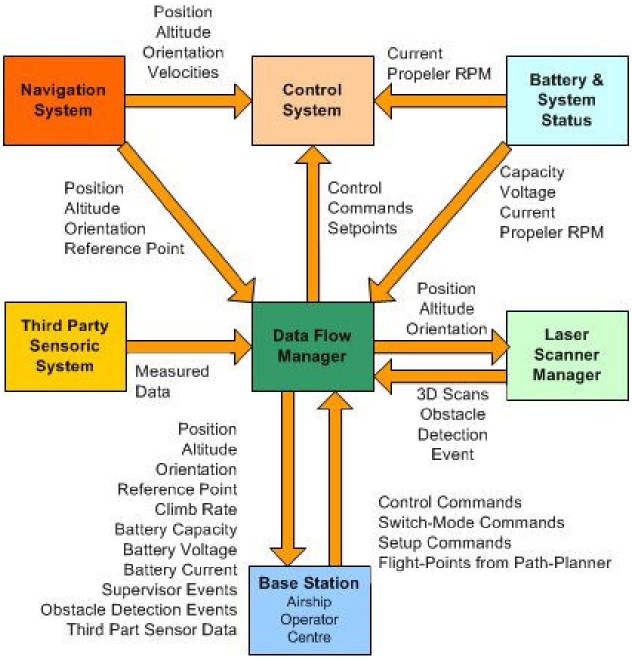


Fig. 6. Schematic of the data relations between the system components.

We can categorize the data to different types based on their utilization. Some applications like navigation and control system need to execute their calculations in real-time. For these applications a continuous exchange of their data in real-time is necessary. This real-time requirement is realized by joining and executing these application modules on the same real-time platform.

Real-time delivery: Some applications require that a message must be delivered within a specified time, otherwise the message becomes useless or its information content is decreasing after the time bound. (Kansal 2010)

Another type of data represents event based messages. These messages can be usually invoked by the operator like a control, setup commands or path-planner recalculated flight-points or any obstacle detection event fired by collision avoiding system. These messages usually don't appear in every real-time sample, but a low latency delivery is necessary to keep their relevancy. Moreover their delivery status has to be checked. This approach is used also for the main exchange of information between airship on-board systems and the base station on the ground.

The next type of data is usually collected and stored on a storage device of the PC 104 board. These data are further used for post processing and analysis. The source of these data is the laser scanner system and any third-party sensoric system. It often handles large amounts of data. In some cases it is required to transfer these data online to the base station or any mission operation centre on the ground. The communication link needs to handle larger amounts of data, but the low latency by data delivery is not critical.

5. FLOW CONTROL AND REPRESENTATION OF DATA

In the previous few sections the primary requirements and constraints of data management architecture, system architecture and data categorization have been discussed. But the most important part of data management architecture is to provide a reliable communication between all of these collaborative modules. This capitol explains the main concept of the communication architecture.

5.1 Communication Architecture

The communication architecture was designed as follows. In the beginning the best concept has been searched. Several communication approaches have been tested to find out the best results. The main criteria of a wireless network structure are reliability of a communication channel, communication range, baud rate and latency by transferring data packets. Moreover all the devices have to be certified by European Regulations.

The ad-hoc network structure was chosen as the first communication approach. In this case the network is decentralized and does not rely on a pre-existing network infrastructure. Communication between the nodes in the network is realized just like a point-to-point data link. This type of wireless network structure fits the requirements of communication architecture for the UAV system, because there are just two nodes which need to communicate with each other. It is the communication node located on the base station on the ground and the on-board communication node of the UAV system. As the first type of wireless network a popular Wi-Fi IEEE 802.11 b/g standard transmitting in the 2.4 GHz frequency band with allowed transmit power of 100 mW has been chosen. Several tests have been realized to prove our main criteria. The best result was the baud rate around 10 Mbit/s and very low latency by transferring data packets, but even in the line of sight the communication range was just about a 100 to 190 meters by using 9 dB Omni-directional antennas. As a second difficulty by using this wireless network type appears a problem with the reliability of a communication channel. After disconnection caused by coming out of the communication range and returning back into it, the Wi-Fi device was not able to establish the communication link before reaching a very near bound of about a 50 meters from base station. It means that a save operation area for a flying UAV to keep reliable communication channel has to be restricted even to a half of the available communication range. As a second type of wireless communication a radio-modem transmitting in ISM 868 frequency band with 250 mW transmit power (by European regulations is allowed up to 500 mW) was applied. During some tests the same criteria have been evaluated. In this case the baud rate was just 28.8 kbit/s with latency about 120 to 250 ms, but the communication range in the line of sight was about 1500 meters by using 5 dB Omni-directional antennas. The reliability of the communication channel was excellent.

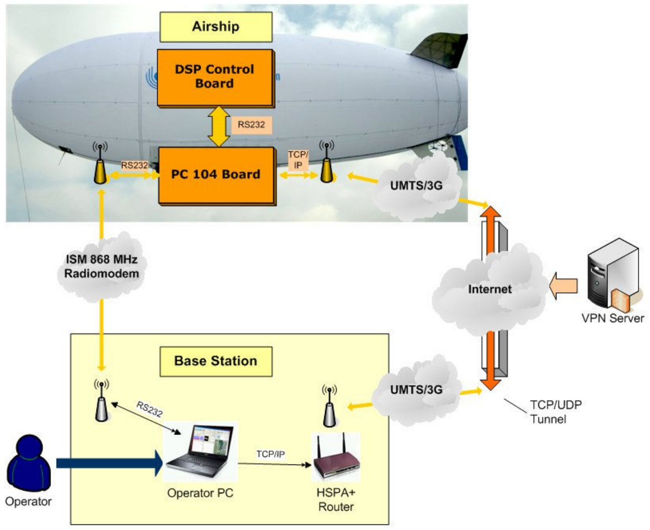


Fig. 7. Schematic of the Communication Architecture.

For the second approach we decided to use an already existing network infrastructure of the Universal Mobile Telecommunications System (UMTS). This type of 3G network infrastructure is nowadays provided by any mobile telecommunication operator. The client is able to connect to internet using resources of this infrastructure. On the other hand this internet connection is provided by the telecommunication operator as a paid service and there is no direct possibility to make a point-to-point data link. To be able to communicate from one UMTS network device on the base station to another device connected to the PC 104 board on the airship a virtual private network (VPN) has to be established. For this purpose a VPN server with a public IP address is used. The same criteria have been evaluated as well as in the case of ad-hoc networks. The baud rate, reliability of communication channel and data delivery latency depends of the actual signal coverage and of network type. The typical values are presented in the table Tab. 1. Especially the data delivery latency varies depending on the actual network link capacity and of course on the delays caused by overhead of the necessary VPN network, which routes the data via VPN server.

	Downlink	Uplink	Latency
HSDPA	7.2 Mbit/s	3.6 Mbit/s	100-300 ms
UMTS	384 kbit/s	128 kbit/s	200-1000 ms
EDGE	236 kbit/s	59.2 kbit/s	300-2500 ms
GPRS	60 kbit/s	40 kbit/s	400-3000 ms

Tab. 1. Typical downlink, uplink and latency values of most used GSM standards.

This type of network connection is in our communication architecture primary used as a redundant communication channel. So the whole communication architecture, illustrated on the figure 7, consists of the communication channel based on point-to-point communication via radio-modem, and as a second redundant channel the communication via UMTS is used.

5.2 Data Flow and Reliability

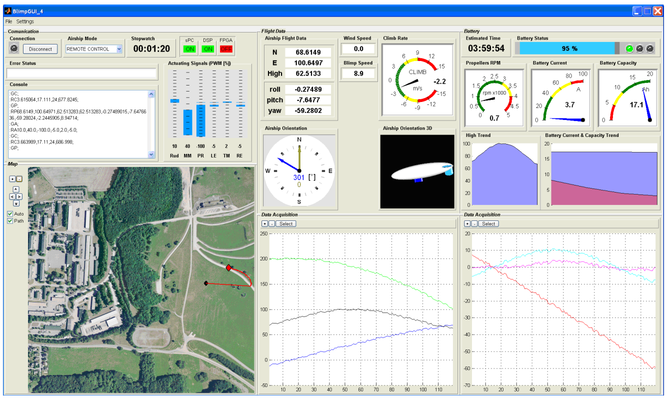
The data management architecture has to manage and to keep reliable data flow with the base station using the pre-designed communication architecture. Various data have to be routed via these two communication channels by specific criteria. The very important telemetric data, control commands from the operator with a higher priority needs to be transferred with low latency. These information and messages are usually smaller data-packets which are transferred via a point-to-point radio-modem communication channel. For the data, which low latency is not such a limiting factor like data measured by third-party sensors or large amounts of collected data, the second UMTS communication channel will be used. The UMTS communication channel will be used also as a backup system in situations by radio-modem communication interruption. The pre-developed data router manager (in testing phase) handles this functionality. The data flow between the components is realized as a service oriented architecture (SOA). It is based on a pre-developed lightweight protocol applied mainly over TCP protocol, so it uses primary a TCP socket interface.

When some component needs a functionality not provided by itself, it asks the system for the required service. If other component of the system has this capability, their location will be provided and finally the client component can consume the service using the common interface in the provider component. The interface of a SOA component must be simple and clear enough to be easily implemented in different platforms both hardware and software. (Pastor 2006)

On the base station is also a pre-developed data hub server as a service provider for multiple clients on the ground like control panel component on the base station or any third-party clients, which use a service to become the data from its sensoric system located on the airship as it is illustrated on the figure 4. The data hub server checks also a delivery status of each message and control the network delay measuring the message delivery with time constraints.

5.3 Data Representation

The next challenge for the data management architecture is the effective and intuitive representation of distributed data for the operator. For this purpose a special control panel module for the airship as a graphical user interface (GUI) has been developed. It is the important module in the system architecture used for interaction between the operator and the airship. The main goal of such an interaction is an effective operation and control of the airship. Moreover, the feedback from the airship sensors aids the operator in making operational decisions.



Airship Mode: REMOTE CONTROL
Stopwatch: 00:01:20
sPC: ON DSP: ON FPGA: OFF

```
:GC;
:RC3.615064,17.111,24,677.8245;
:GP;
:RP68.6149,100.64971,62.513283,62.513283,-0.27489015,-7.64766
36,-59.28024,-2.2445905,8.94714;
:GA;
:RA10.0,40.0,-100.0,-5.0,2.0,-5.0;
:GC;
:RC3.663989,17.11,24,686.998;
:GP;
```

Actuating Signals: Rud 10, MM 40, PR -100, LE -5, TM 2, RE -5

N 68.6149
E 100.6497
High 62.5133
roll -0.27489
pitch -7.6477
yaw -59.2802

Wind Speed 0.0
Blimp Speed 8.9
Climb Rate -2.2
Airship Orientation 301 [°]

Estimated Time 03:59:54
Battery Status 95 %
Propellers RPM 0.7
Battery Current 3.7
Battery Capacity 17.1

Fig. 8. GUI of the airship's control panel module.

The control panel provides the input and output capabilities. The input functionality allows the operator to control the airship. It enables him to select the airship control mode, setup some system options. The output functionality provides the indication of important data like information about the actual position, altitude and orientation of the airship. For a better overview for the operator the airship orientation is visualized with the help of a three-dimensional virtual reality model. The actuating signals are indicated as well as the information of the airship systems and batteries state. There is also a possibility to record the measuring data using the pre-developed data acquisition module.

The actual position is displayed in the two-dimensional (2D) map, which is georeferenced in Universal Transverse Mercator (UTM) geographic coordinate system. It is a very practical grid-based 2D Cartesian coordinate system. The position on the Earth is referenced in the UTM system by the UTM zone, and the easting (E) and northing (N) coordinate pair. (Wikipedia) The navigation system of the airship calculates the actual position in local north, east, down (NED) coordinates as it is illustrated in Figure 9. This local coordinate system needs to be referenced to specify its location on the surface of the Earth. For this purpose the navigation system sets the reference location in Earth-Centered-Earth-Fixed (ECEF) conventional terrestrial coordinate system and this information is sent to the control panel. The reference point is displayed on the map of the control panel. The transformation from ECEF coordinate system to UTM is necessary. As a first step the position needs to be converted from ECEF to geodetic coordinates WGS84 (latitude ϕ, longitude λ, height h) using the Kaplan algorithm.

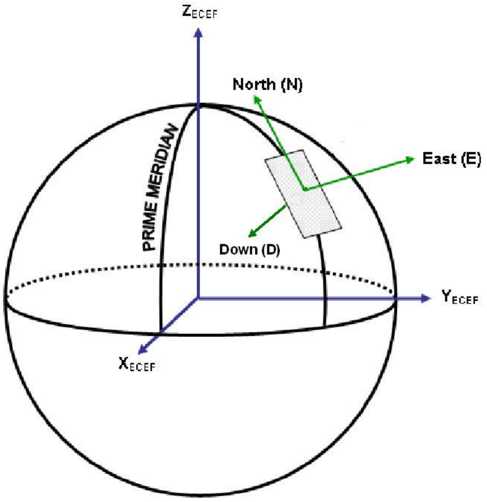


Fig. 9. NED Coordination system.

Afterwards the conversion from geodetic coordinates (latitude ϕ, longitude λ) to UTM coordinates pair (E,N) and the UTM zone is calculated.

6. 3D SCANNING OF THE EARTH SURFACE BY A LASER SCANNER

A typical data acquisition task for our airship could be a scanning of the Earth's surface by a laser scanner. The laser scanner is an additional sensor installed on-board. A basic principle is based on measuring the distance of the scanned points from chosen reference point that is usually placed over the surface. Knowing its exact position and having a possibility to measure its distance precisely to all points of the scanned surface and direction of the measurement, it is possible to calculate exactly their positions.

6.1 Choice Of The Measurement Method

A 3D scanning of the surface based on reflection of waves can be carried out in several ways: By sonar using ultrasonic waves, or by radar working with electromagnetic waves. The distance from a reference point may also be determined using stereoscopy, whereby it is evaluated by comparing two pictures of an object made from two reference points. This leads to different angles of view. Precision of all these systems is given by the used wave lengths and by their propagation. Systems like sonar have a relatively slow response speed and a relatively low resolution not enabling to see details of the scanned surface. A reasonable improvement was brought by laser technology guaranteeing high linearity of the ray movement, its fast speed (above 300 000 m/s) and measurement of details in range of millimeters. After bringing reflectors to Moon by Apollo missions, by using laser technology it was e.g. possible to measure that the Moon is spiraling away from Earth at a rate of 38 mm per year (Espenak 1994), whereby its mean distance is about 384.467 kilometers and the round pulse trip time is about 2½ seconds. So, the laser systems are able to give precise measurement information over a long distance.

By scanning the Earth's surface, the resulting precision depends highly on precision of the reference point localization and orientation of the beam. The final resolution will directly depend on the density of emitting and orientation of the beams.

6.2 Measurement Principle

Assume a laser device emitting laser beams oriented in a plane. That means that this system is able to scan just information about a curve lying in this plane. When wishing to scan the entire space, such scan must be repeated in infinitely many other planes. In order to get stereoscopic information the measurement must be repeated at least from one additional place. Such a measurement can e.g. be based on a translation movement, when the laser moves its reference position and the beam falls to the surface from other reference positions.

During the measurement, the reference position and orientation of the beam must be measured in order to enable the calculation of the points of reflection. This information is usually achieved by a measurement and recording system completing the laser measurement by information about the laser reference position in 3D coordinating system [x,y,z], as well as angular orientation of the beam in all three axes X, Y, Z denoted usually as Roll, Pitch, and Yaw.

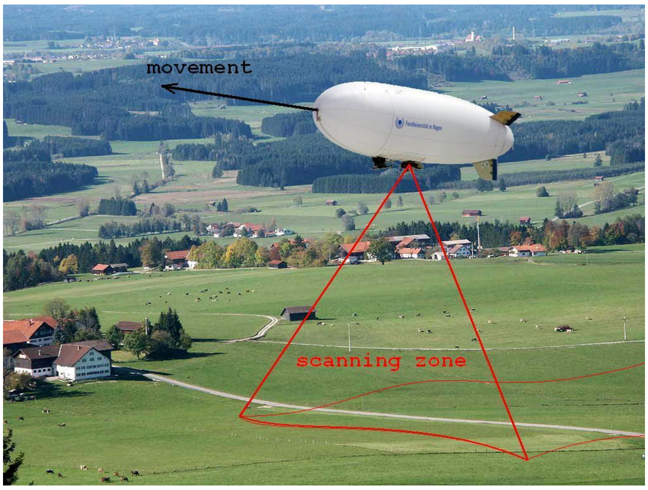


Fig. 9. Earth's surface scanning illustration by a laser scanner.

6.3 Data Identification

For this task two frames are given: navigation frame (FRAME) and body frame (BODY). Data acquisition is considered in body frame. For following representation of measured data, they have to be transformed into a global navigation frame.

The body frame may freely move and rotate in the navigation frame, i.e. it may change the coordinates of its origin in navigation frame, as well as orientation of its axes. The navigation frame is usually considered to be static.

While scanning laser beams are emitted with the origin placed to the origin of body frame [0,0,0].

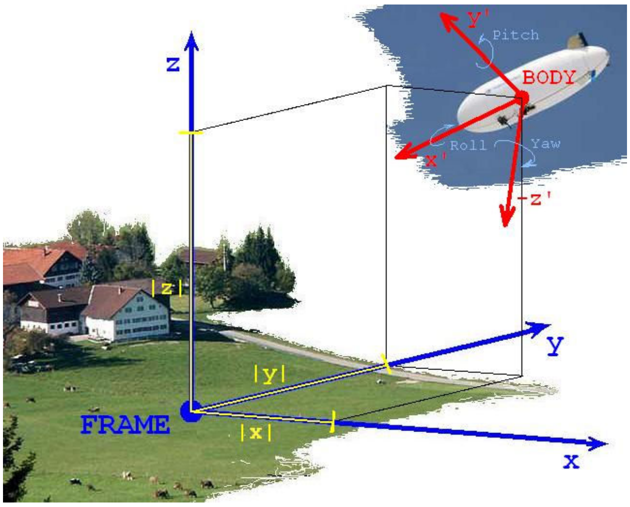


Fig. 10. Illustration of the coordinate systems BODY and FRAME.

These always lie in a plane YZ. After being reflected by the Earth's surface, its points are calculated according to:

$$y1 = d * \sin\alpha \tag{1}$$

$$z1 = d * \cos\alpha \tag{2}$$

d – Measurement of the scanned point from laser.
α – Angle between zero reference position and the emitted beam.

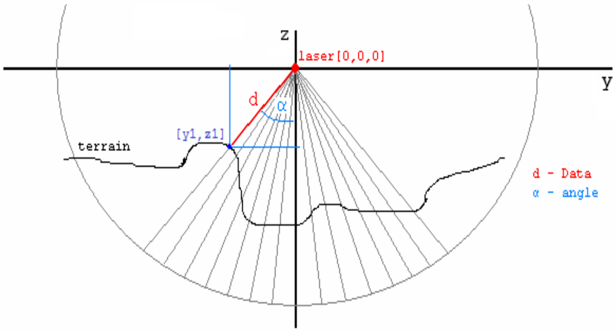


Fig. 11. Scanning principle.

Under zero reference position we will understand beam orientation parallel to axis Z, i.e. vertically to the Earth in direction of –Z. While looking into the X direction, deviation from the zero position do the right means deviation under a negative angle α. Deviation to the left corresponds to positive angles. This sign convention enables an easy calculation of Y coordinates. For such a measurement, the first X-coordinate will be zero, i.e. the measurement will be represented as [0,y,z].

Known coordinates in body frame are necessary to transform into navigation frame coordinates. The coordinate transformation is calculated by means of the direction cosine matrix (DCM). The DCM matrix performs the coordinate transformation of a vector in navigation frame axes (ox0, oy0, oz0) into a vector in body frame axes (ox3, oy3, oz3).

$$\begin{bmatrix} ox_3 \\ oy_3 \\ oz_3 \end{bmatrix}_{BODY} = DCM \begin{bmatrix} ox_0 \\ oy_0 \\ oz_0 \end{bmatrix}_{FRAME} \qquad (3)$$

In the case of 3D space there exist in principle 12 different options how to describe a general body orientation by a sequence of three rotations. It could be demonstrated that the rotation according to X (Roll) axis, then with respect to Y(Pitch) axis and finally with respect to Z(Yaw) axis is different from that one with the same rotation in Roll, Pitch and Yaw carried out in different order. DCM transformation matrix is defined by combination of three axis transformation matrices M(X), M(Y), M(Z).

In our case the rotation angles in body frame are known and we need to carry out the inverse transformation into the navigation frame coordinates. This can be done according to:

$$\begin{bmatrix} ox_0 \\ oy_0 \\ oz_0 \end{bmatrix}_{FRAME} = DCM^{-1} \begin{bmatrix} ox_3 \\ oy_3 \\ oz_3 \end{bmatrix}_{BODY} \qquad (4)$$

By translation movement of the body frame origin, the transformation is calculated as follows:

$$\begin{bmatrix} ox_0 \\ oy_0 \\ oz_0 \end{bmatrix}_{FRAME} = DCM^{-1} \begin{bmatrix} ox_3 \\ oy_3 \\ oz_3 \end{bmatrix}_{BODY} + \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{POSITION} \qquad (5)$$

In the navigation frame the points are recorded in a way that corresponds to their position within the scanned surface.

6.4 System Implementation

The type of used laser scanner was briefly described in the section 3. Up to now a few first tests have been done mostly in laboratory conditions. The laser scanner enables a measurement in the range from 30 to 60 meters (m). The final results depend on the surface reflection capability, whereby precision given by the producer is for measurement up to 10 m in range of 0.03 m, and for measurement up to 30 m in the range up to 0.05 m. The first statement was partially confirmed by a measurement repeated several times in laboratory conditions, whereby the maximal difference of two measured samples was 28 mm. Of course, this does not yet guarantee the absolute measurement precision, but characterizes repeatability of the measurement

During the tests the laser scanner was installed 1.35 m over the surface on a moving platform which enables to control the translation in axis X. The position x values in body frame X axes was incrementally preset in range of -0.9 m up to 0.9 m. Laser beam was emitted in the plane YZ. Pitch was changed in the range from -32° up to 32°. It corresponds to changes of Y value within the range from -0.81 m up to 0.81 m for the given distance of the body frame origin respectively laser scanner over the scanned surface.

These tests were carried out by scanning two boxes and one box with banked surface and decreasing height from 0.245 m, 0.18 m to 0.075 m. The results, displayed in Figure 13, illustrate a relatively correct reproduction of the scanned surface distorted partially by the measurement noise.

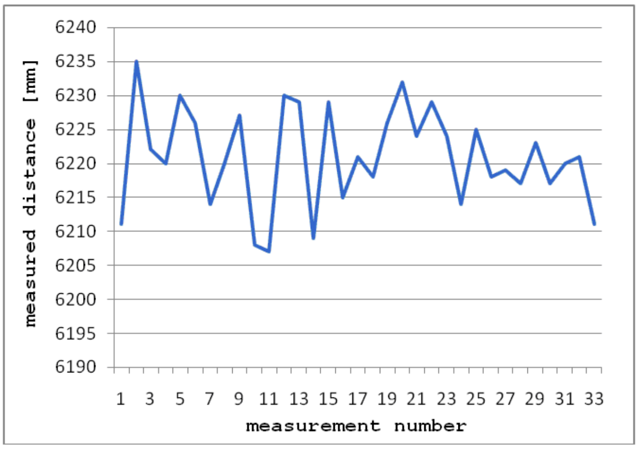


Fig. 12. Results of the measurement precision for 33 realized experiments.

For noise elimination some various filtration procedures may be applied.

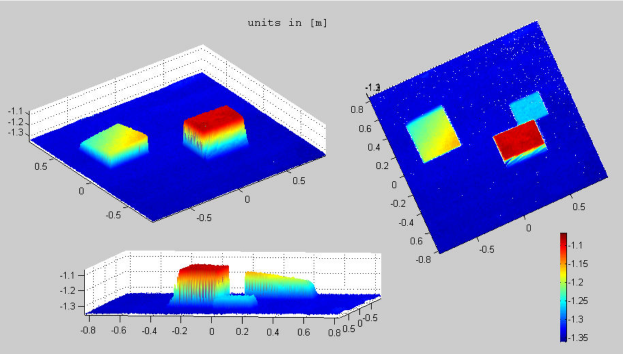


Fig. 13. Results of scanning.

The determination of the laser position respectively the body frame origin and its rotation angles (roll, pitch, yaw) has a dominant influence on the final measurement precision.

7. CONCLUSION

This paper has introduced our approach to design a modular, service based data management architecture for our UAV system, in our case the robotic airship, to provide the data management and routing as well as a reliable distribution of information between all the collaborative system modules. For this purpose the system and communication architecture has been described. As a very important component of the complex system the control panel module for intuitive data representation and control has been developed. Such an architecture enables to establish a robust communication structure, comfortable control capability for the operator and even to provide a service based interface for a mission operation centre with an opportunity of online distribution of important data from the airborne sensoric system as a valuable source of information. The main constraints and experiences by the design and practical experiments have been mentioned as well. The most valuable experiences were gained by fulfilling the major tasks in our two research projects, where design and development of the architecture was formed by the real constraints and problems, which appear during the tests.

As a future work some enhancement of communication data flow and a more robust reliability are planned. The communication structure could be used as a source of information to any leashing algorithms.

Leashing is then performed by having the robot react to low message rates by moving towards the base in order to improve the communication. (Hauert 2010)

The data throughput by a communication channel with low band rate like our radio-modem data link could be enhanced by any real-time data compression methods.

Huffman style compression scheme exploits temporal locality and delta compression to provide better bandwidth utilization, thus reducing latency for real time applications. (Szalapski 2010)

The next interesting mission for our airship is a 3D scanning of the Earth surface by the new additional sensor system, the laser scanner. The last section introduces the measurement method and our first results in this problem.

REFERENCES

Gerke, M. (2009). Mine detection and removal (iMR). In: *Annual Report*, Fakultät für Matematik und Informatik, FernUniversität in Hagen. http://prt.fernuni-hagen.de/forschung/ANNUALS/ET09_PRT.pdf

Gerke, M. (2009). Forest fire prevention and extinction (iWBB). In: *Annual Report*, Fakultät für Matematik und Informatik, FernUniversität in Hagen. http://prt.fernuni-hagen.de/forschung/ANNUALS/ET09_PRT.pdf

Pastor, E. Lopez, J. Royo, P. (2006). A Hardware/Software Architecture for UAV Payload and Mission Control. In: Proceedings of the *25th Digital Avionics Systems Conference,* October 15-19, 2006, Portland, Oregon.

Kansal, P. Kansal, D. Arun, B. (2010). Compression of Various Routing Protocol in Wireless Sensor Network. In: *International Journal of Computer Applications (0975-8887)* Vol. 5 – No.11.

Hauert, S. Leven, S. Zufferey, J.-Ch. Floreano, D. (2010) Communication-based Leashing of Real Flying Robots. In Proceesings of the *2010 IEEE International Conference on Robotics and Automation.* Anchorage Convention District, May 3-8, 2010, Anchorage, Alaska, USA

Szalapski, T. Madria, S. (2010). Real-Time Data Compression in Wireless Sensor Network. In: MDM'10 Proceedings of the 2010 *Eleventh International Conference on Mobile Data Management.* Kansas City, Missouri.

Wikipedia [online]. Universal Transverse Mercator coordinate system. <http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system>. Retrieved 2011-01-15

Espenak, F. [online] (1994). NASA – Measuring the Moon's Distance.<http://eclipse.gsfc.nasa.gov/SEhelp/ApolloLaser.html>. Retrieved 2011-01-26

Mathworks [online]. Direction Cosine Matrix to Wind Angles.< http://www.mathworks.com/help/toolbox/aeroblks/directioncosinematrixtowindangles.html>. Retrieved 2011-01-25.

Gain-scheduled LQR-control for an autonomous airship

I. Masár and E. Stöhr *

* *FernUniversität in Hagen*
Process Control Engineering Group
Universitätsstr. 27
58093 Hagen, Germany
(Tel: +49 2331 9871102; e-mail: ivan.masar@fernuni-hagen.de)

Abstract: In the past two years, an autonomous airship was developed at our department as a flying sensor platform. Our main research areas during this period were navigation, modelling and automatic control of the airship. In this article, we present a gain-scheduled LQR control design for the airship. First, the mathematical model of the system and its linearization will be introduced. After that, we split the linearized system in a lateral and a longitudinal subsystem. With the combined gain-scheduled controlled subsystems, a high-level navigation system allows the airship to follow an appropriate flight trajectory.

Keywords: airship, LQR control, gain scheduling.

1. INTRODUCTION

Unmanned aerial vehicles (UAV's) became very popular in the last years, because of the availability of cheap HW components likes motors, sensors (digital gyroscopes, compass, accelerometers, barometers, etc.), embedded micro-controllers and power units. There exists plenty of designs, including autonomous helicopters and quad- (or more-) copters ([Prior et al. (2009)], [Jaimes et al. (2008)]), but they have a few common problems - typically a very small payload capacity, a short flight time and an intrinsic instability. These properties reduce the number of possible applications to short-time reconnaissance missions - very often with low-cost and therefore low-quality cameras.

As we were confronted with the problem of designing an UAV, which could be used as a carrying platform for a lot of various sensor and camera systems for a couple of hours, which should be safe for ground personal and easy to manoeuvre for the operator, we decided to return back to aviation roots and to design an autonomous airship. Namely, there are some very advantageous properties of such lighter-then-air flying system. In particularly a bigger payload capacity; a longer flight time, because there is no need to actuate the airship all the time; a very good stability in the air and no strong oscillations caused by the motors, that can influence the sensors and cameras.

The developed airship is shown in Fig. 1. The length of the airship is 9 m and the maximal diameter of the hull is 2.5 m. The hull volume of 24 m^3 is filled with helium and could lift a payload of about 5-6 kg. The airship is controlled by two 700W synchronous main motors with propellers, which could be vectored, by the tail thruster and the elevator and rudder control surfaces. The operational speed of the airship is up to 30 km/h.

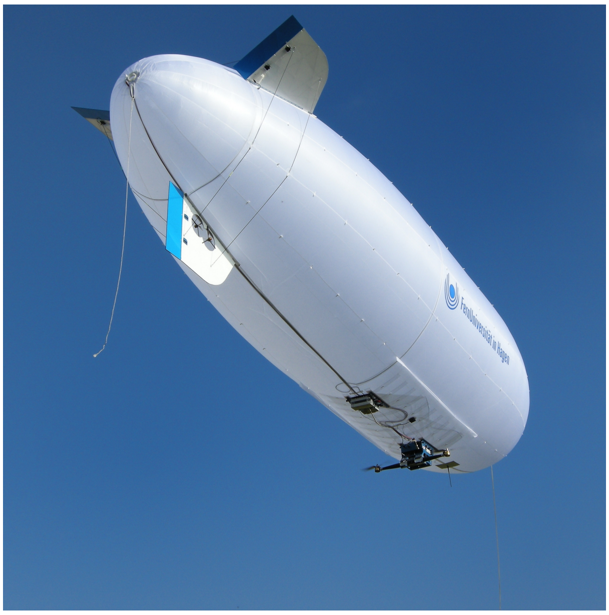


Fig. 1. An autonomous airship of the University of Hagen

2. MATHEMATICAL MODEL OF THE AIRSHIP

For the derivation of the airships mathematical model, two coordinate systems are defined according to Fig. 2. An earth-fixed inertial coordinate system e is used to describe the position and orientation of the airship.

The origin of the body coordinate system b is in the airships centre of volume CV. In this coordinate system, the linear and angular velocities of the airship are described. Moreover, it is used to specify the forces and moments acting on the airships body. The center of gravity CG is the point, in which the total mass of the airship is concentrated (mean location of the gravitational forces).

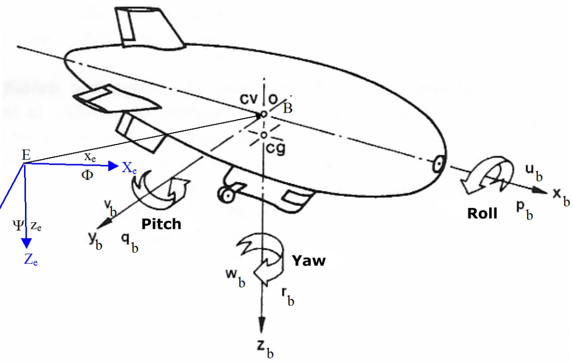


Fig. 2. Coordinate systems of the airship [Khoury (2004)]

2.1 Kinematic model

The kinematic model of the airship involves the position (η_1) and orientation (η_2) vector η, described in the earth-fixed coordinate system e:

$$\eta = {}^e\eta = \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix};\ \eta_1 = \begin{pmatrix} x \\ y \\ z \end{pmatrix};\ \eta_2 = \begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix} \tag{1}$$

Moreover, ν is the vector of linear (ν_1) and angular (ν_2) velocities

$$\nu = {}^b\nu = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix};\ \nu_1 = \begin{pmatrix} u \\ v \\ w \end{pmatrix};\ \nu_2 = \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{2}$$

and τ is the vector of forces (τ_1) and moments (τ_2) acting on the airship.

$$\tau = {}^b\tau = \begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix};\ \tau_1 = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix};\ \tau_2 = \begin{pmatrix} K \\ M \\ N \end{pmatrix}, \tag{3}$$

where X, Y and Z are forces acting in x-, y- and z-direction and K, M and N are moments acting about these axes. Both are defined in the coordinate system b.

Transformation between the coordinate systems For the transformation of our vector quantities between the coordinate systems, we use the Euler-rotation matrices about the x-, y- and z- axis:

$$R_{x,\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\phi & s_\phi \\ 0 & -s_\phi & c_\phi \end{pmatrix} \tag{4}$$

$$R_{y,\theta} = \begin{pmatrix} c_\theta & 0 & -s_\theta \\ 0 & 1 & 0 \\ s_\theta & 0 & c_\theta \end{pmatrix} \tag{5}$$

$$R_{z,\psi} = \begin{pmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6}$$

Transformation of translational velocities Using the xyz-convention, we can define a Jacobi-matrix to transform the translational velocities from coordinate system b to e [Fossen (1991)], [Brockhaus (2001)]:

$$J_1(\eta_2) = R^T_{z,\psi} R^T_{y,\theta} R^T_{x,\phi} \tag{7}$$

$$J_1(\eta_2) = \begin{pmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi c_\phi s_\theta \\ s_\psi c_\theta & c_\psi c_\phi + s_\phi s_\theta s_\psi & -c_\psi s_\phi + s_\theta s_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \tag{8}$$

The translational velocity applies to:

$$\dot{\eta_1} = J_1(\eta_2)\nu_1 \tag{9}$$

The matrix $J_1(\eta_2)$ is ortogonal $(J_1(\eta_2)^T J_1(\eta_2) = I)$. Therefor the inverse velocity transformation can be written as:

$$\nu_1 = J_1(\eta_2)^{-1}\ \dot{\eta_1} = J_1(\eta_2)^T\ \dot{\eta_1} \tag{10}$$

Transformation of rotational velocities A Jacobi-matrix $J_2(\eta_2)$ is used to transform rotational velocities between the coordinate systems b and e [Brockhaus (2001)].

To transform $\dot{\eta_2}$ from e into the vector ν_2, described in the airships body coordinate system b, following equation must be solved:

$$\nu_2 = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + R_{x,\phi} \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + R_{x,\phi} R_{y,\theta} \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \tag{11}$$

$$= \begin{pmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{pmatrix} \cdot \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \tag{12}$$

$$\nu_2 = J_2^{-1}(\eta_2)\ \dot{\eta_2} \tag{13}$$

The matrix $J_2(\eta_2)$ is not orthogonal $(J_2(\eta_2)^{-1} \neq J_2(\eta_2)^T)$. Solving the equation for $\dot{\eta_2}$ yields to:

$$\dot{\eta_2} = J_2(\eta_2)\ \nu_2 = \begin{pmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{pmatrix} \cdot \nu_2 \tag{14}$$

Kinematic equations in vector form The kinematic equations can be expressed in a more compact vector form as:

$$\begin{pmatrix} \dot{\eta_1} \\ \dot{\eta_2} \end{pmatrix} = \begin{pmatrix} J_1(\eta_2) & O_{3\times3} \\ O_{3\times3} & J_2(\eta_2) \end{pmatrix} \cdot \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} \tag{15}$$

$$\dot{\eta} = J(\eta)\ \nu \tag{16}$$

2.2 Dynamic model

The Newton's laws of linear and angular momentum for rigid bodys (RB) describe the airships dynamic behaviour.

$$M_{RB}\dot{\nu} + C_{RB}(\nu)\nu = \tau_b \tag{17}$$

The rigid body inertia matrix M_{RB} can be expressed as:

$$M_{RB} = \begin{pmatrix} mI_{3\times3} & -mS(r_G) \\ mS(r_G) & I_b \end{pmatrix} \tag{18}$$

One possible variant to express the rigid body Coriolis- and centrifugal matrix C_{RB} is:

$$C_{RB}(\nu) = \begin{pmatrix} 0_{3\times3} & -mS(\nu_1) - mS(\nu_2)S(r_G) \\ -mS(\nu_1) + mS(r_G)S(\nu_2) & -S(I_b\nu_2)) \end{pmatrix} \tag{19}$$

With

- the vector cross product $S(k)$,
- the inertia tensor I_b,
- the mass of the airship m and
- the position of the center of mass ${}^b r_G$.

The vector τ_b of external forces and moments can be written as

$$\tau_b = \tau_{add} + \tau_V + \tau_{rest} + \tau_{FK} + \tau_{Fin} + \tau_A \tag{20}$$

where the parts of τ_b are:

- added -mass and -Coriolis and centripedal matrix $\tau_{add} = -M_A\dot{\nu_r} - C_A(\nu_r)\nu_r$
- damping forces of fuselage $\tau_V = -D(\nu_r)\nu_r$
- restoring forces $\tau_{rest} = -g(\eta)$
- Froude-Krylov forces $\tau_{FK} = M_{FK}\dot{\nu}_c$
- fixed fins and control surfaces $\tau_{Fin} = D_{Fin}(\nu_r, \delta) - D_{Fin}(\nu_r)$
- propulsive forces $\tau_p = P_S + P_B + P_{Stern}$

ν_r is the relative airspeed.

Added -mass and -Coriolis and centripedal matrix A body accelerates a certain surrounding air mass with movement. The body behaves thereby simplified, as if an additional mass would be adhere to it. For a completely in a medium submerged body with three planes of symmetry and low velocity, the added mass- M_A and the added Coriolis and centripedal- matrix $C_A(\nu_r)$ can therefore be considered:

$$M_A = diag(X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}) \tag{21}$$

$$C_A(\nu_r) = \begin{pmatrix} 0 & 0 & 0 & 0 & Z_{\dot{w}}w_r & -Y_{\dot{v}}v_r \\ 0 & 0 & 0 & -Z_{\dot{w}}w_r & 0 & X_{\dot{u}}u_r \\ 0 & 0 & 0 & Y_{\dot{v}}v_r & -X_{\dot{u}}u_r & 0 \\ 0 & Z_{\dot{w}}w_r & -Y_{\dot{v}}v_r & 0 & N_{\dot{r}}r_r & -M_{\dot{q}}q_r \\ -Z_{\dot{w}}w_r & 0 & X_{\dot{u}}u_r & -N_{\dot{r}}r_r & 0 & K_{\dot{p}}p_r \\ Y_{\dot{v}}v_r & -X_{\dot{u}}u_r & 0 & M_{\dot{q}}q_r & -K_{\dot{p}}p_r & 0 \end{pmatrix} \tag{22}$$

For example: The force Y_A along the y axis due to an acceleration $\dot{v}$ in y-direction is:

$$Y_A = Y_{\dot{v}}\dot{v} \;\; mit \;\; Y_{\dot{v}} = \frac{\partial Y}{\partial \dot{v}} \tag{23}$$

Damping forces of fuselage The damping effects on the airships fuselage are mainly caused by linear and quadratic surface frictions, due to laminar and turbulent fluid motions. For a completely submerged body, the linear and quadratic damping forces can simplyfied be written:

$$\begin{aligned} D(\nu_r) &= diag(X_u, Y_v, Z_w, K_p, M_q, N_r) \\ &+ diag(X_{u|u|}|u_r|, Y_{v|v|}|v_r|, Z_{w|w|}|w_r|, \\ &K_{p|p|}|p_r|, M_{q|q|}|q_r|, N_{r|r|}|r_r|) \end{aligned} \tag{24}$$

Restoring forces The gravitational force $W = mg$ works against the buoyancy force $B = g\rho V$.

$$f_G = J_1^{-1}(\eta_2)\begin{pmatrix} 0 \\ 0 \\ W \end{pmatrix}; \; f_B = -J_1^{-1}(\eta_2)\begin{pmatrix} 0 \\ 0 \\ B \end{pmatrix} \tag{25}$$

The restoring forces can be therefore represented by:

$$g(\eta) = -\begin{pmatrix} f_G(\eta_2) + f_B(\eta_2) \\ r_G \times f_G(\eta_2) + r_B \times f_B(\eta_2) \end{pmatrix} \tag{26}$$

${}^b r_B$ is the position of the center of buoyancy.

Froude-Krylov forces The Froude-Krylov forces results from differences of pressure, acting on the body surface due to the flow rate v_c of the surrounding air masses. The Froude-Krylov forces can be expressed as:

$$\tau_{FK} = M_{FK}\dot{\nu}_c \tag{27}$$

M_{FK} can be calculated with der inertia tensor and the mass of the displaced air. Assuming a buoyancy neutral airship and homogeneous mass distribution, the inerta matrix M_{FK} could be set equal to M_{RB}.

Fixed fins and control surfaces The airship has three by 120 degrees displaced stabilisation fins with control surfaces. Each fin has its own coordinate system f_i, whose x_i- direction is equal to the bodies x-direction. The control surfaces can be rotated about an angle δ_i around its y_i-axis. [Campa and Innocenti (1999)] The rotation matrix ${}^b R_{fi}$ transforms the fin forces from coordinate system f_i to b.

$${}^b R_{f_i} = R^{-1}_{x,(\pi/2+2\pi k/3)} R^{-1}_{y,\delta_i} \quad (k = 0, 1, 2) \tag{28}$$

δ_i is null for all fixed stabilisation fins. The velocity of fin i in relation to the wind flow, denoted in coordinate system b, can be written:

$${}^b V_{Fi/c} = ({}^b V_{b/c} + {}^b w_{b/c} \times {}^b P_{Fi}) \tag{29}$$

${}^b P_{Fi}$ is the position of the i-te fin. ${}^b w_{b/c}$ is the angular- and ${}^b V_{b/c}$ the translational- velocity part of the relativ airspeed ν_r.

${}^b V_{Fi/c}$ can be transformed into the fin coordinate system f_i as follows:

$${}^{fi} V_{Fi/c} = {}^{fi} R_b {}^b V_{Fi/c} \tag{30}$$

The wind in y-direction of f_i leads to no application of force and can therefore be neglected. The angle of attack α_{fi} between the wind- and the fin coordinate system is given through:

$$\alpha_{fi} = atan2({}^{fi} V_{Fi/c}(z), {}^{fi} V_{Fi/c}(x)) \tag{31}$$

The positive x- axis of the wind w coordinate system shows toward the relative velocity between fin and wind-current. A transformation from the wind into the fin coordinate system can be achieved with the matrix:

$$ {}^{fi}R_w = \begin{pmatrix} c_{\alpha_{fi}} & 0 & -s_{\alpha_{fi}} \\ 0 & 1 & 0 \\ s_{\alpha_{fi}} & 0 & c_{\alpha_{fi}} \end{pmatrix} \tag{32} $$

Damping forces of fins are given in the wind coordinate system:

$$ {}^{w}F_{Fi} = -\frac{1}{2}\rho A \begin{pmatrix} C_D \\ C_C \\ C_L \end{pmatrix} {}^{fi}V_{Fi/c}^{T}\,{}^{fi}V_{Fi/c} \tag{33} $$

C_D, C_C and C_L are aerodynamic damping coefficients.

The forces are transformed from w into the b coordinate system:

$$ {}^{b}F_{Fi} = {}^{b}R_{fi}{}^{fi}R_w{}^{w}F_{Fi} \tag{34} $$

and leads to moments via the lever arm ${}^{b}P_{Fi}$:

$$ {}^{b}M_{Fi/b} = {}^{b}P_{Fi} \times {}^{b}F_{Fi} \tag{35} $$

τ_{Fin} can be devided into forces and moments caused by the stabilisation fins $D_{Fin}(\nu_r)$ and the control surfaces $-D_{Fin}(\nu_r,\delta)$.

$$ \begin{aligned} \tau_{Fin} &= -D_{Fin}(v_r,\delta) + D_{Fin}(\nu_r) \\ &= \sum_{i=1}^{4}\begin{pmatrix} {}^{b}F_{Fi}(v_r,\delta) \\ {}^{b}M_{Fi/b}(v_r,\delta) \end{pmatrix} + \sum_{i=1}^{4}\begin{pmatrix} {}^{b}F_{Fi}(v_r) \\ {}^{b}M_{Fi/b}(v_r) \end{pmatrix} \end{aligned} \tag{36} $$

Propulsive forces The airship is actuated by two main gondola motors with propellers, witch could be vectored (rotated about an angle α around the bodys y-axis). The motor forces ${}^{a}F_S$ and ${}^{a}F_B$ points in x-direction from coordinate system a. The rotation matrix ${}^{b}R_a$ transforms the motor forces from coordinate system a to b.

$$ {}^{b}R_a = R_{y,\alpha} \tag{37} $$

$$ {}^{b}F_S = {}^{b}R_a{}^{a}F_S; \quad {}^{b}F_B = {}^{b}R_a{}^{a}F_B \tag{38} $$

The main gondula motor forces and moments can be written as:

$$ P_S = \begin{pmatrix} {}^{b}F_S \\ {}^{b}r_S \times {}^{b}F_S \end{pmatrix}; \; P_B = \begin{pmatrix} {}^{b}F_B \\ {}^{b}r_B \times {}^{b}F_B \end{pmatrix} \tag{39} $$

${}^{b}r_S$ and ${}^{b}r_B$ are the positions of the main gondula propellers.

The tail motor forces and moments acting upon the airship can be written as:

$$ P_{Stern} = \begin{pmatrix} F_{Stern} \\ r_{Stern} \times F_{Stern} \end{pmatrix} \tag{40} $$

F_{Stern} and ${}^{b}r_{Stern}$ are the tail motor force and its position.

Wind current The wind current points towards the x-axis of w. One can transform the wind current from w into the e coordinate system, by rotating about the angle of attack α_c and the angle of sideslip β_c:

$$ {}^{e}V_c = R_{y,\alpha_c}R_{z,-\beta_c} \cdot {}^{w}V_c \tag{41} $$

The transformation in the body coordinate system can be achieved with $J(\eta)$. The relative speed $\nu_r = \nu - \nu_c$ is the speed between the airship (ν) and the wind velocity (ν_c).

2.3 The overall airchip model

The overall non-linear 6-DOF (degrees of freedom) model of the airship has the following form:

$$ \begin{aligned} &(M_{RB} + M_A)\dot{\nu_r} + C_{RB}(\nu)\nu + C_A(\nu_r)\nu_r \\ &+D(\nu_r)\nu_r + D_{Fin}(\nu_r) + g(\eta) = \tau_P + D_{Fin}(\nu_r,\delta) \\ &\dot{\eta} = J(\eta)\;\nu \end{aligned} \tag{42} $$

3. DESING OF THE AIRSHIP CONTROL

A very good solution to handle all input/output variables is to use a state-space controller. The airship is a nonlinear system. Therefore, we decided to use a nonlinear gain-scheduling controller. A gain-scheduling controller optimizes the linear state-space controller parameters to various operating points of the airship. There exists many design methods for linear state controllers. We choose an LQR-based controller design, because of its relatively easy implementation. However, LQR-controller designs are only applicable for linear systems. Therefore, in a first step, the linearization of the airship model is necessary. It is reasonable to split the linearized model of the whole airship into a lateral and a longitudinal subsystem. Each of them are used to control some particularly motions. The control-loop structures for lateral and longitudinal motions are explained bellow. Moreover, the airship is a nonholonomic 6-DOF system. So, it is not possible to control it in every direction with arbitrary orientation. The following control system is at this time only designed without implementing the tail fin control surfaces as controller inputs.

3.1 Linearisation of the airship model

Equilibrium points The nonlinear airship system can be writtes as:

$$ \dot{x}(t) = f(x(t),u(t)) \tag{43} $$

An equilibrium point corresponds to a condition, at which the dynamical system is in steady state.

$$ 0 = \dot{x}_0 = f(x_0,u_0) \tag{44} $$

Due to the complex aerodynamic data, the equilibrium points cannot be found analytically. We use a numeric optimization algorithm to find the equilibrium points x_0, u_0 over the flight envelope. Therefor, a convex cost function is minimized.

$$ \begin{aligned} F = \min = {}& \dot{u_0}^2 + \dot{v_0}^2 + \dot{w_0}^2 + \dot{p_0}^2 + \dot{q_0}^2 + \dot{r_0}^2 + \\ & (\dot{x_0} - V_{GS})^2 + \dot{y_0}^2 + \dot{z_0}^2 + \dot{\phi_0}^2 + \dot{\theta_0}^2 + \dot{\psi_0}^2 + \\ & {v_0}^2 + {p_0}^2 + {q_0}^2 + {r_0}^2 + {x_0}^2 + {y_0}^2 + {z_0}^2 + {\phi_0}^2 + {\psi_0}^2 \end{aligned} \tag{45} $$

The scheduling-variable V_{GS} specifies various speeds of the airship in $\dot{x_0}$ direction and so the flight envelope.

Linaerization With the derivations

$$ \widetilde{x} = x - x_0 \quad \widetilde{u} = u - u_0 \tag{46} $$

the system can be lineaized by using a multivariable Taylor serie.

$$\dot{\tilde{x}}(t) \approx \left.\frac{\partial f}{\partial x}\right|_{x=x_0;u=u_0} \widetilde{x} + \left.\frac{\partial f}{\partial u}\right|_{x=x_0;u=u_0} \widetilde{u} \qquad (47)$$

Were $A = \left.\frac{\partial f}{\partial x}\right|_{x=x_0;u=u_0}$ and $B = \left.\frac{\partial f}{\partial u}\right|_{x=x_0;u=u_0}$ are the $n \times n$ and $n \times m$ Jacobi-matrices.

Because of the complex nonlinear data (aerodynamic data), the linearization cannot be done analytically. The numerical linearization can be done by perturbing each state or input signal slightly from its equilibrium points. The matrices A (and B) can be approximated as:

$$\tilde{\dot{x}}_1 - \tilde{\dot{x}}_{-1} = f(x_0) + h\left.\frac{\partial f}{\partial x}\right|_{x=x_0} - f(x_0) + h\left.\frac{\partial f}{\partial x}\right|_{x=x_0} \qquad (48)$$

$$A = \left.\frac{\partial f}{\partial x}\right|_{x=x_0} \approx \frac{\tilde{\dot{x}}_1 - \tilde{\dot{x}}_{-1}}{2h} \qquad (49)$$

The linear model of the airship can be expressed in a well-known general state-space form:

$$\dot{\widetilde{x}} = A\widetilde{x} + B\widetilde{u} \qquad (50)$$

$$y = C\widetilde{x} + D\widetilde{u} \qquad (51)$$

$$C = I_{(12\times12)} \qquad (52)$$

$$D = 0_{(12\times3)} \qquad (53)$$

3.2 Lateral controller

The heading system is nonholonomic, because the yaw angle can not be controlled without changing the airships y position. The main purpose of the lateral system is to control the yaw angle ψ. For this reason, the y position is not included in the state vector: $xH = v, p, r, \phi, \psi$. The input variable is $u = F_{Stern}$. [Metelo and Campos (2003)]

The state and input matrices for the lateral sub-system are extracted from the linearised model by using the new state vector x_H:

$$A_H = \begin{array}{c} \begin{array}{ccccc} v & p & r & \phi & \psi \end{array} \\ \begin{pmatrix} a_{2,2} & a_{2,4} & a_{2,6} & a_{2,10} & a_{2,12} \\ a_{4,2} & a_{4,4} & a_{4,6} & a_{4,10} & a_{4,12} \\ a_{6,2} & a_{6,4} & a_{6,6} & a_{6,10} & a_{6,12} \\ a_{10,2} & a_{10,4} & a_{10,6} & a_{10,10} & a_{10,12} \\ a_{12,2} & a_{12,4} & a_{12,6} & a_{12,10} & a_{12,12} \end{pmatrix} \end{array} \begin{array}{c} \\ v \\ p \\ r \\ \phi \\ \psi \end{array} \qquad (54)$$

$$B_H = \begin{array}{c} F_{Stern} \\ \begin{pmatrix} b_{2,1} \\ b_{4,1} \\ b_{6,1} \\ b_{10,1} \\ b_{12,1} \end{pmatrix} \end{array} \begin{array}{c} \\ v \\ p \\ r \\ \phi \\ \psi \end{array} \qquad (55)$$

The state vector deviation from the operating point is given by:

$$\tilde{x}_H = x_H - x_{H_{d(0)}} = \begin{pmatrix} v - v_0 \\ p - p_0 \\ r - r_0 \\ \phi - \phi_0 \\ \psi - \psi_0 - (\psi_d - \psi_0) \end{pmatrix} \qquad (56)$$

The controlled input variable

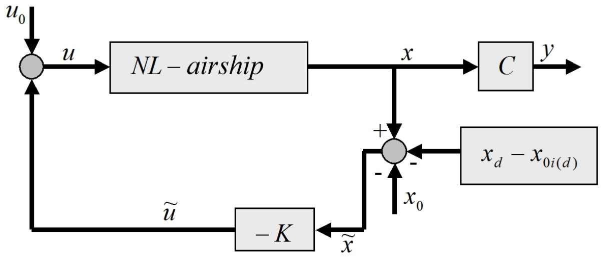


Fig. 3. Closed-loop lateral system

$$\tilde{u}_H = -K_H\tilde{x}_H \qquad (57)$$

stabilizes the state space equation of the lateral system:

$$\dot{x}_H = A_H\tilde{x}_H + B_H\tilde{u}_H \qquad (58)$$

In order to find the optimal controller parameters, the quadratic cost criterion should be minimized:

$$J = \frac{1}{2}\int_0^\infty \left[x^T(t)Qx(t) + u^T(t)Ru(t)\right] dt = \min_K \qquad (59)$$

Therefore, the optimal controller parameters K can be calculated by solving the Riccati equation for the matrix P:

$$PA + A^TP - PBR^{-1}BP + Q = 0 \qquad (60)$$

$$K = R^{-1}B^TP \qquad (61)$$

The parameters $q_i > 0$ from the weight matrix $Q = diag(q_1, q_2, \ldots, q_n)$ are weighting the associated states. The parameters $r_i > 0$ from the weight matrix $R = diag(r_1, r_2, \ldots, r_m)$ are weighting the associated inputs.

The closed-loop lateral system with state controller is shown in Fig. 3.

All poles of the open-loop lateral system are stable. However, one of them is on the imaginary axis. The pole location of the open-loop lateral system is shown is Fig. 4.

The weighting matrices $Q = Q^T \geq 0$ with $n = 5$ and $R = R^T \geq 0$ with $m = 1$ elements on the main diagonal should be selected in such a way, to ensure, that the output variable F_{Stern} does not saturate. Moreover, the state variable $(\psi - \psi_d)$ should converges faster to zero as the others. This request can be realised through a larger value for q_5 in the weighting matrix Q.

Resulting poles of the closed-loop heading system with LQR-controller are shown in Fig. 5. The step response for the yaw angle is shown in Fig. 6.

3.3 Longitudinal controller

The longitudinal (XZ) control is dedicated to control the state variables u and z. The gondola drives with propellers are used as input variables. Similar to the lateral system, the longitudinal system is not independent controllable in all DOF. The yaw angle ψ affects the controllability of

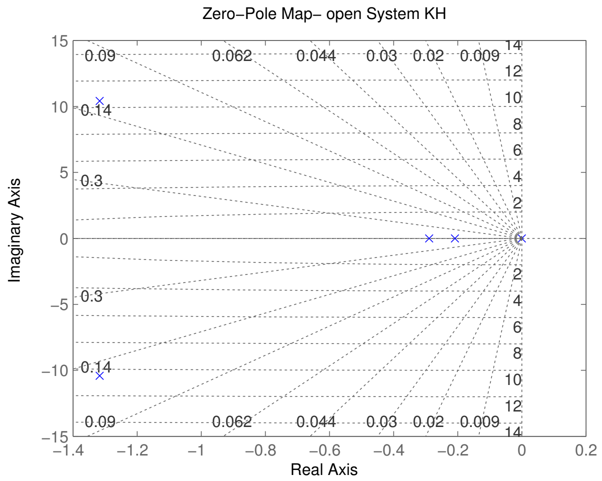


Fig. 4. Poles of the open-loop lateral system

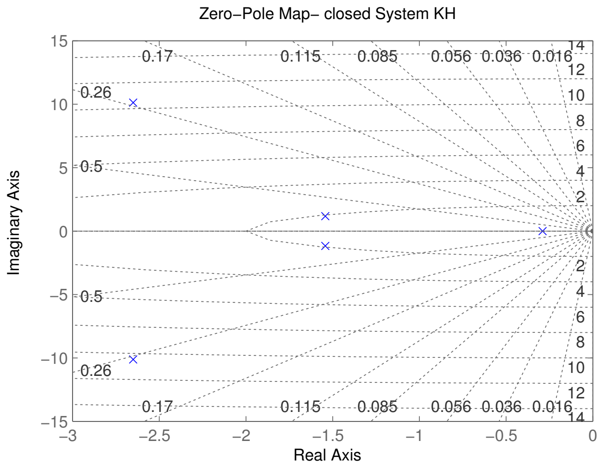


Fig. 5. Poles of the closed-loop lateral system

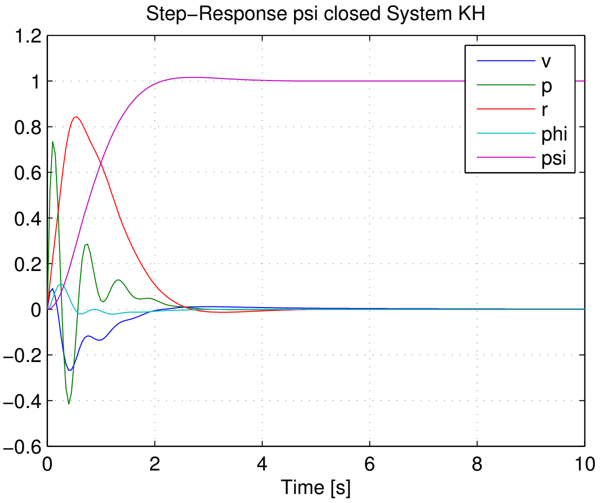


Fig. 6. Step response (yaw angle) of the lateral system

the x−position of the airship. Therefore, the state variable x is not included in state vector for longitudinal control. Moreover, the pitch angle θ is at the equilibrium points not equal to zero. Thus, it is not reasonable to control this angle to zero. Therefore, this variable is also not included in the resulting state vector $x_{XZ} = u, w, q, z$. The input vector includes the forces, generated by the main propellers. The input forces $F(\alpha)$ are splitted for control purpose in an x and z direction $u = F_x, F_z$. [Metelo and Campos (2003)]

$$F = FS + FB \tag{62}$$

$$F_x = Fcos(\alpha) \tag{63}$$

$$F_z = Fsin(\alpha) \tag{64}$$

The linearized longitudinal state and input matrices are:

$$A_{XZ} = \begin{array}{cc} \begin{array}{cccc} u & w & q & z \end{array} & \\ \begin{pmatrix} a_{1,1} & a_{1,3} & a_{1,5} & a_{1,9} \\ a_{3,1} & a_{3,3} & a_{3,5} & a_{3,9} \\ a_{5,1} & a_{5,3} & a_{5,5} & a_{5,9} \\ a_{9,1} & a_{9,3} & a_{9,5} & a_{9,9} \end{pmatrix} & \begin{array}{c} u \\ w \\ q \\ z \end{array} \end{array} \tag{65}$$

and

$$B_{XZ} = \begin{array}{cc} \begin{array}{cc} F_x & F_z \end{array} & \\ \begin{pmatrix} b_{1,2} & b_{1,3} \\ b_{3,2} & b_{3,3} \\ b_{5,2} & b_{5,3} \\ b_{9,2} & b_{9,3} \end{pmatrix} & \begin{array}{c} u \\ w \\ q \\ z \end{array} \end{array} \tag{66}$$

The primary controlled variables u and v are coupled, because the flight speed u can affect the altitude z. Since the LQR-controller has only proportional feedback, there will result a not negligible error. By introducing an I-type feedback, this error can be eliminated. For this purpose, new state variables are defined:

$$x_{\int} = (x_{\int u}, x_{\int z}) = \left(\int_0^t u(\tau)d\tau, \int_0^t z(\tau)d\tau \right) \tag{67}$$

The new state variables can be added to the existing longitudinal state-space variables,

$$\tilde{x}_{XZ} = x_{XZ} - x_{XZ_{d(0)}} = \begin{pmatrix} u - u_0 - (u_d - u_0) \\ w - w_0 \\ q - q_0 \\ z - z_0 - (z_d - z_0) \end{pmatrix} \tag{68}$$

witch results in the following controlled longitudinal overall system:

$$\begin{pmatrix} \dot{\tilde{x}}_{XZ} \\ \dot{\tilde{x}}_{\int u} \\ \dot{\tilde{x}}_{\int z} \end{pmatrix} = \begin{pmatrix} A_{XZ} & 0_{4\times2} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & 0_{2\times2} \end{pmatrix} \begin{pmatrix} \tilde{x}_{XZ} \\ \tilde{x}_{\int u} \\ \tilde{x}_{\int z} \end{pmatrix} + \begin{pmatrix} B_{XZ} \\ 0_{1\times2} \\ 0_{1\times2} \end{pmatrix} \tilde{u}_{XZ} \tag{69}$$

The input signals are calculated as:

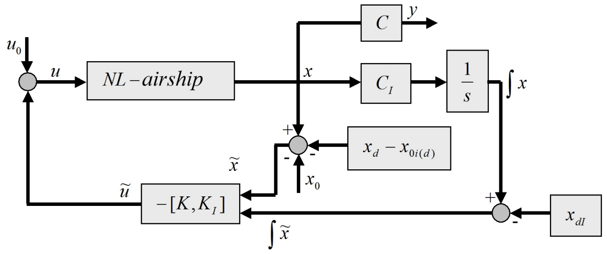


Fig. 7. Closed-loop longitudinal system with additive I-controller

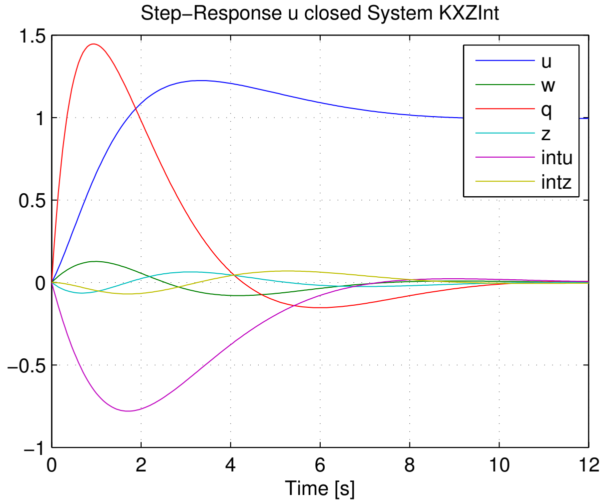


Fig. 8. Step response of the longitudinal system

$$\tilde{u}_{XZ} = -\left(K \;\; K_{\int u} \;\; K_{\int z} \right) \begin{pmatrix} \tilde{x}_{XZ} \\ \tilde{x}_{\int u} \\ \tilde{x}_{\int z} \end{pmatrix} \tag{70}$$

Fig. 7 shows the closed-loop longitudinal system.

The procedure for calculating the controller parameters is similar to the lateral system. In this case, the state variables $u - ud, z - zd, x_{\int u}$ and $x_{\int z}$ should be controlled faster than the others. Therefore, the weighting coefficients $q1, q4, q5$ and $q6$ of the matrix Q must be greater then the values of $q2$ and $q3$, respectively. The resulting LQR-controller is a 2×6 matrix.

The values of integral-feedback gains can reach very large values, if the airship has an unilateral error from the desired flight path. Therefore, it is necessary to implement an anti-wind-up saturation of these signals.

The step response of the longitudinal closed-loop system for the output variable u is shown in Fig. 8.

3.4 Gain-scheduling controller

A linear controller can only guarantee stability in a short range about the operating point, for which it was designed. However, the airship should be controlled under all possible flight conditions. This can be realised by using a nonlinear controller.

A nonlinear gain-scheduling controller acts as a switch between a lot of linear controllers. Each of them are designed for a specific operating point. Operating points are equilibrium points for a certain value of the gain-scheduling variable s [Moutinho (2007)].

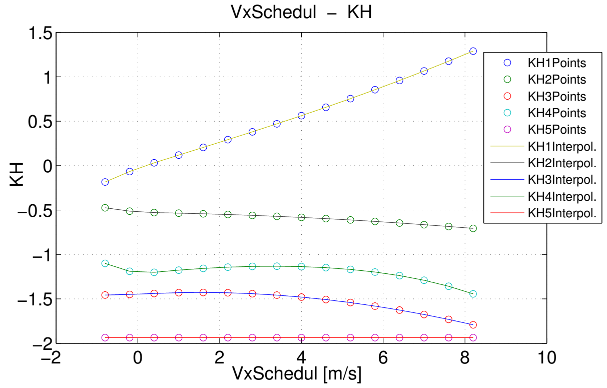


Fig. 9. Parameters of the LQR-controller for lateral system

$$0 = f(x_0(s), u_0(s)) \tag{71}$$

The gain-scheduling variable s for the airship is defined by its speed. This speed corresponds to the x-velocity ${}^{e_{xy}}v_{rx}$ of the relative speed v_r, transformed into XY plane of the earth coordinate system:

$${}^{e_{xy}}v_r = \begin{pmatrix} {}^{e_{xy}}v_{rx} \\ {}^{e_{xy}}v_{ry} \\ {}^{e_{xy}}v_{rz} \end{pmatrix} = J_1(\phi, \theta, 0) \cdot v_r \tag{72}$$

The nonlinear airship model $\dot{x} = f(x, u)$ is linearized at all equilibrium points (Eq. 71.), given by various values $s(1), s(2), ...s(n)$ of the gain-scheduling variable s.

In this way, n linear systems are developed:

$$\dot{\tilde{x}} = A(s)\tilde{x} + B(s)\tilde{u} \tag{73}$$

$$\tilde{y} = C(s)\tilde{x} \tag{74}$$

For each of the n linear systems, an LQR controllers $K(s)$ will be calculated:

$$\tilde{u} = -K(s)\tilde{x} \tag{75}$$

The controller parameters $K(s)$ are then interpolated between all equilibrium points. The interpolated controller parameters $K(s)$ are depending on the value of the gain-scheduling variable s.

The graphs in Fig. 9. shows the calculated and interpolated parameters of the LQR-controller for the lateral control system.

It is easy to detect that there are some step changes in the parameter values. These can lead to an instability of the whole closed-loop system, if it is controlled between two operating points with big changes in controller parameters. As a solution we designed a checking algorithm, which allows only to change the coefficients of controller parameters with limited range and so without step changes in the scheduled controller coefficients. The step changes of the controller coefficients are detected by using the derivation of the controller coefficients curve. The following figure

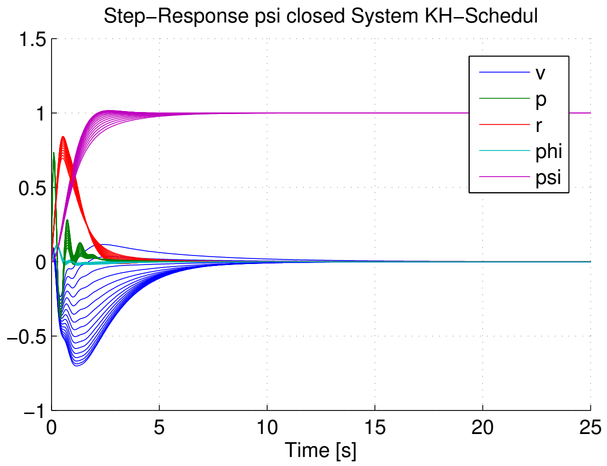


Fig. 10. Step response of the gain-scheduled controlled lateral system

Fig. 10. shows the step responses of the designed lateral gain-scheduling controller.

3.5 Navigation

The above described lateral and longitudinal control systems are not able to control the airship in all 6DOF, as it is a nonholonomic and so a not fully controllable system. Besides the yaw angle ψ, it must be also controlled the altitude Z_e and the speed u in X_b-direction. For flying on an appropriate flight trajectory, the desired set values must be proper generated for the controllers.

For this purpose we designed a navigation module, which switches between the two operating modes of the airship - steady state control and point-to-point flight. The point-to-point control is used to flight to a new target point. In this mode, the lateral control system is used for adjusting the airship orientation towards the desired set position. The longitudinal system controls the desired speed u and desired Z_e position of the airship. The desired speed u is increased, if the control deviation of the orientation decreases. If the airship is nearby the target point, the steady state control mode is activated. In this mode, the maximal speed of the airship is limited and decreases with decreasing distance to the target point. This strategy prevents the airship to do very ineffective maneuvers in the neighbourhood of the set position. Moreover, the airship can flight backward to the target position in this mode [Metelo and Campos (2003)].

Combining these strategies yields to a complex motion control for the airship. Fig. 11. shows simulation results of a typical airship flight with take-off, movement on a linear and circular composed trajectory with constant altitude and landing [Moutinho (2007)].

4. CONCLUSION

In this paper we presented the design procedure for a gain-scheduled LQR controller for an autonomous airship. Two types of control sub-systems (lateral, longitudinal) have been designed from the nonlinear 6DOF airship model to fulfill different goals (yaw as well as speed and position control). The navigation system generates appropriate inputs for the controllers of both sub-systems and combines them to a complex control unit, witch allows the airship to follow an adequate flight trajectory. Simulations shows a good performance of the designed control system, which will be tested on the real airship in near future.

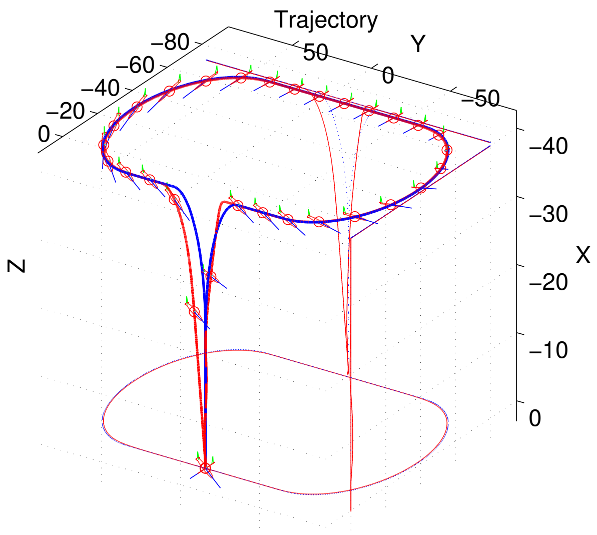


Fig. 11. Spline trajectory with gain-scheduled LQR-controller

REFERENCES

Brockhaus (2001). *Flugregelung*. Springer, Berlin, 2 edition.

Campa, G. and Innocenti, M. (1999). Model of an underwater vehicle (shark.pdf). University of Pisa, 56126 Pisa, Italy, Department of Electrical Systems and Automation. URL `http://www.mathworks.com/matlabcentral/fileexchange/1207`.

Fossen, T.I. (1991). *Nonlinear Modelling and Control of Underwater Vehicles*. Ph.D. thesis, Norwegian Institute of Technology, Department of Marine Technology.

Jaimes, A., Kota, S., and Gomez, J. (2008). An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles uav(s). In *IEEE International Conference on System of Systems Engineering*.

Khoury, G. (2004). *Airship Technology (Cambridge Aerospace)*. Cambridge Univ Press, 1 edition.

Metelo, F.M.S. and Campos, L.R.G. (2003). Vision based control of an autonomous blimp (videoblimp). URL `http://www.isr.ist.utl.pt/vislab/thesis/03-videoblimp-tfc.pdf`.

Moutinho, A.B. (2007). *Modelling and Nonlinear Control for Airship Autonomous Flight*. Ph.D. thesis, UNIVERSIDADE TÉCNICA DE LISBOA, INSTITUTO SUPERIOR TÉCNICO.

Prior, S.D., Shen, S.T., Karamanoglu, M., Odedra, S., Erbil, M.A., Barlow, C., and Lewis, D. (2009). The future of battlefield micro air vehicle systems. In *International Conference on Manufacturing and Engineering Systems, Huwei, Taiwan*.

Experience in Designing Autonomous Airplane

Róbert Schochmann* Jakub Pilka Martin Suchý*** Ivan Pestún**** Mikuláš Huba*******

Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology
(e-mails: *robo@schochmann.com
**xpilkaj@stuba.sk
***13chaser.@gmail.com
****ivan.pestun@stuba.sk
*****mikulas.huba@stuba.sk)

(website: www.animatechnika.com)

Abstract: In this paper we present experience gathered in our work on UAV design. UAVs (Unmanned Autonomous, or Aerial Vehicles) are radio controlled flying vehicles, e.g. airplanes equipped with RF communication systems, cameras and sensors (e.g. accelerometers, gyros, temperature and pressure sensors, a GPS module etc.) enabling to be capable of autonomous flight. Currently UAVs are mostly used in the military area to safely watch areas of potential risk without endangering the lives of people otherwise needed to provide information on such areas. Lately UAVs have begun to make their way into civil applications as well. Possible uses include monitoring of forest fires, following fleeing suspects by the police, scanning of ground surface to create maps on a local level and so on, or simply for fun.

1. INTRODUCTION

In this paper we present our achievements gain so far in approximately 3 year experience. Our goal is to design hardware and software architecture for a fully autonomous UAV capable of flying to its destination to carry out some task (e.g. to gather visual information) and safely returning without the need for user interference.

2. CONSTRUCTION AND ELECTRONICS

We chose a standard model RC (Remotely Controlled) plane made of balsa and plywood. This means that it is light but also strong enough to survive vibrations generated by the 4 horsepower, 2-stroke combustion engine. The total weight is around 4.5 kilograms (including electronics) and it is possible to add another 0.75 kilogram load and still maintain stable flight conditions.

The plane electronics is powered by two Li-Pol batteries (one for the main system and the other for the servomotors). The star architecture is centred around the main 32-bit PIC (Peripheral Interface Controller) microcontroller. Other 8-bit PICs whose purpose is to collect and process data from sensors are connected to it. The reason for this architecture is distribute the A/D conversion time, whereas the main μC can be used for other necessary calculations. The main advantage of this architecture is that the main μC can work without having to wait for its peripheral circuits to finish a conversion from the analogue sensors. Another advantage is the ease possibility of adding additional system components since the computing power of the main μC is never fully used. We prepared our PCB (Printed circuit board) designs to be able to add more electronic equipment if needed. XBee RF communication modules compliant with IEEE 802.11 standards are used for wireless communication with the model. The control station is composed of a laptop, joystick to control the aircraft and of the XBee RF communication module.

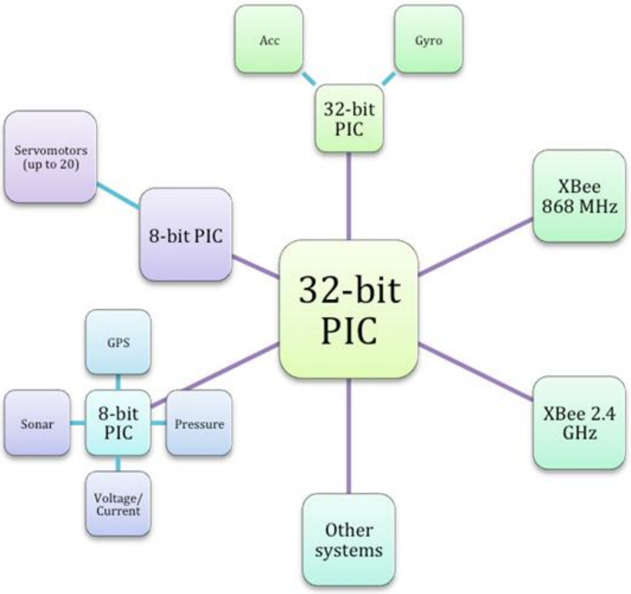


Fig. 1. Hardware architecture

3. UTILITY SOFTWARE OF THE GROUND STATION

The utility software gives the pilot all the necessary information he needs to pilot the plane. This includes a video feed from the on-board camera, acceleration in X, Y and Z axes, roll and pitch indication is also present. Also the joystick calibration will be available, as well as the fuel level indication and battery charge status. Our goal is to make GUI as simple and intuitive as possible so that the pilot can fully concentrate on controlling the aircraft.

4. COMMUNICATION

Communication between ground station and the plane is performed via XBee RF modules at a 115 200 baud per sec. data rate. Communication runs on 868 MHz frequency. Each of the modules has its role in the network. One acts as coordinator (ground station), the other as end-device (plane).The coordinator works in a broadcast mode so it does not require acknowledgement messages from the end device since they are not needed because minor data dropouts are acceptable. This further increases communication speed. Before the network is established, the coordinator finds the most suitable channel and sets a PAN ID (Personal Area Network) specific only to this network. The end device then scans for coordinator networks and associates only with the one with a specific 64-bit address. This ensures no other coordinators interfere with our communication.

5. PRIMITIVE AUTOPILOT

From the GPS module we get the following data:

- Latitude
- Longitude

True course in degrees is calculated using the following formula:

$$c = \operatorname{acos}\left(\frac{\sin(Lat2) - \sin(Lat1) * \cos(d)}{\cos(Lat1) * \sin(d)}\right)$$

Where ‘Lat1’ and ‘Lat2’ are gathered directly from the GPS module (‘Lat1’ being the last known latitude and ‘Lat2’ the current latitude value) and ‘d’ is the great circle distance. This is the shortest path between two points on a sphere, in our case the globe.

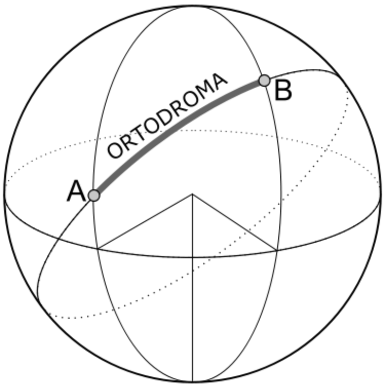


Fig. 2. Great circle distance (‘d’)

From the gathered data we can calculate out true heading. The waypoint heading is a constant telling the autopilot where to steer the aircraft.

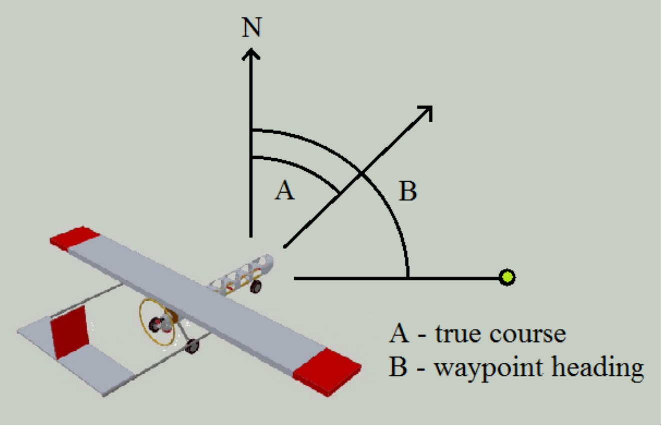


Fig. 3. Waypoint heading calculation

The flight direction deviation is counted as:

Error = waypointHeading – trueHeading.
if (error >= 180) error = error - 360;
if (error <= -180) error = 360 + error;
PolohaServa = stred - error;

The variable PolohaServa sets the turning of the direction flaps on the tail of the plane, to turn the whole plane to the desired course.

6. SENSORS

6.1 GPS module

Our GPS module has an RS-232 output with 4800 baud per sec. communication speed. The data refresh rate is 1 Hz. GPGGA is the protocol we use. The output data consists of 15 blocks, proceeded by a synchronization symbol ‘$’, and a comma separating each data block. The data structure is:

$GPGGA,hhmmss.ss,llll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,
M,x.x,xxxx*hh

1 = World time
2 = Latitude
3 = North/South
4 = Longitude
5 = East/West
6 = GPS signal quality indication
7 = Number of satellites
8 = Horizontal spacing
9 = Altitude (above sea level)
10 = Unit (meter)
11 = Altitude (geoid)
12 = Unit (meter)
13 = Time of last refresh
14 = ID#
15 = Checksum

6.2 Accelerometers and gyroscopes

To get values for pitch, roll and acceleration we use a CHAR6D module.

Fig. 4. CHR-6d module

The CHR-6d is a 6-axis IMU, combining three accelerometer axes and three rate gyro axes in a .8" by .7" footprint. An onboard ARM Cortex™ processor samples and filters gyro and accelerometer outputs and sends results over a TTL (3.3V) UART.

Data Outputs

- 3-axis roll rates (+/- 400 deg/s)
- 3-axis acceleration (+/- 3 g)
- Pitch and roll angles
- 16-bit effective measurement resolution (after oversampling and decimation)

Features

- Configurable digital filter (windowed Parks-McClellan FIR)
- Onboard EKF for pitch and roll angle estimation
- Automatic bias calibration
- High bias stability over temperature
- Adjustable output rates (20 Hz - 300 Hz)
- TTL (3.3V) UART interface

6.3 Ultrasound sonar

We added sonar oriented towards the ground to assist while landing because the camera is mounted on top of the plane making it hard to perform a visual landing since you cannot always see the ground. The range is circa 8 meters and refresh rate is around 14 Hz.

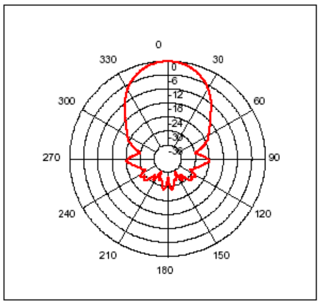


Fig. 2. Sonar radiation

6.4 Prandtle probe

To measure flight speed of aircraft Prandtl probes are used enabling to measure flow velocity of liquids, in our case plane velocity with respect to surrounding air. It is based on measuring difference between the total pressure and the static pressure of flow in a point of flow streamline. The velocity meter in central part of Prandtl probe flow velocity is brought to zero isentropically. With velocity equal to zero measured pressure equals to the total pressure. On outer part of probe where the flow streamlines are tangent the flow velocity is not changed and the pressure measured on that part is called the static pressure. The difference between the total and the static pressure in one point of flow streamline is called dynamic pressure. The dynamic pressure represents kinetic energy of the flow.

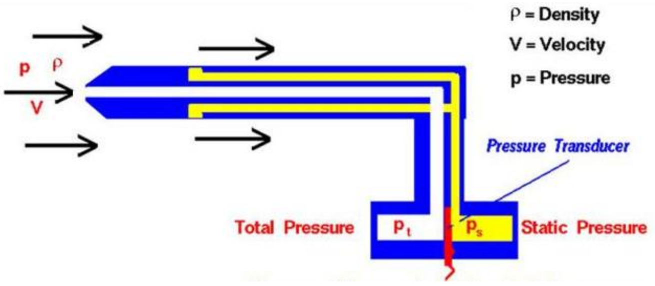


Fig. 3. Measurement principle

Flow velocity can then be calculated.

For ideal gas: $v = \sqrt{\frac{2(p_1 - p_2)}{\rho}}$

For real airflow:

$$v = \sqrt{\frac{2k}{k-1} . g . R . T . \left(1 - \left(\frac{p_2}{p_1}\right)^{\frac{k-1}{k}}\right)}$$

These equations work only for speeds lower than the speed of sound.

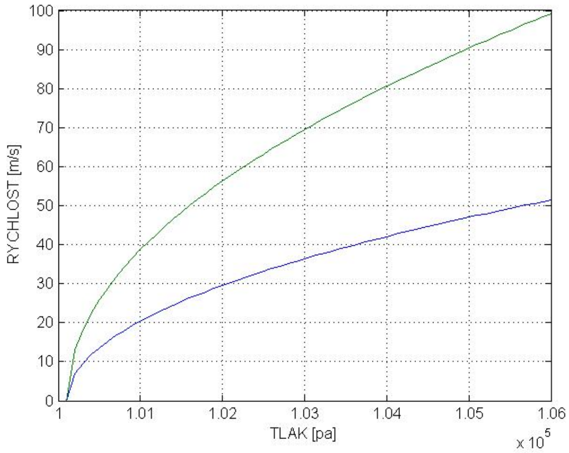


Fig. 4. Speed measurement characteristic for ideal gas (green) and real gas (blue)

The Prandtl probe is capable of measuring speeds up to 100 m/s. This means dynamic pressure up to 6 kPa. The most suitable pressure sensor to measure such values has proven to be MPXV4006G. This is a piesoresistive pressure sensor with integrated temperature compensation.

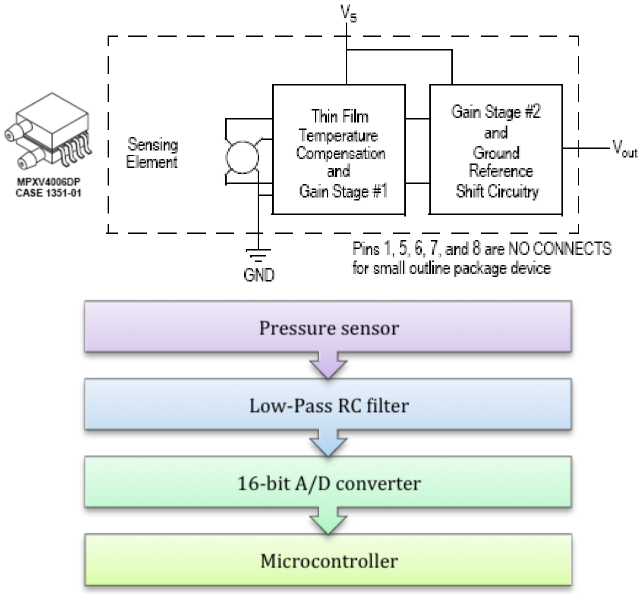

V_S
MPXV4006DP
CASE 1351-01
Sensing Element
Thin Film Temperature Compensation and Gain Stage #1
Gain Stage #2 and Ground Reference Shift Circuitry
V_out
GND
Pins 1, 5, 6, 7, and 8 are NO CONNECTS for small outline package device
Pressure sensor
Low-Pass RC filter
16-bit A/D converter
Microcontroller

Fig. 5. MPXV4006DP pressure sensor diagram and case and module architecture

Sensor output is an analogue signal from 0.2 to 4.7 Volts. Thus, the measurement sensitivity is 766 mV/kPa. For a proper pressure measurement we needed to design appropriate filtering and A/D conversion.

7. PROBLEMS AND SOLUTIONS

7.1 Parachute

Since in the development of the flight control system one has to consider potential problems that could lead to serious malfunctions, the aeroplane should be equipped with a parachute safety system. In the case of uncontrolled movement it should automatically start its activity and to assure sufficiently damped landing.

Before choosing the appropriate type of parachute we need to put together some basic data. This includes:

- Max falling speed
- Weight of the plane + weight of the parachute itself
- Density of air
- Max limit for oscillation
- Deployment speed

There are two possible choices for the parachute shape. One is a more stable cross-chute and the other a cone-shaped chute that is more prone to oscillations but has a higher air resistance constant.

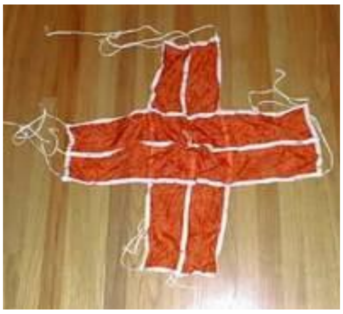


Fig. 8. Cross chute

Air resistance constant = (0.6 – 0.85)
Deployment force constant = (1.1 – 1.2)
Oscillation = (0° - ± 3°)

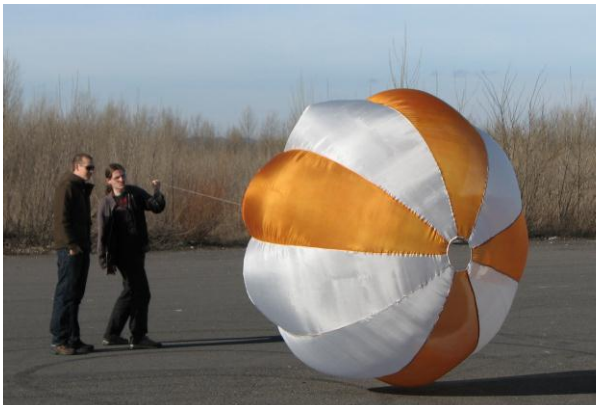


Fig. 9. Our parachute

Air resistance constant = (0.75 – 0.9)
Deployment force constant = (1.8)
Oscillation = (±10° - ± 35°)

Equations needed:

- Brake force
- G-force
  $F_t = m * g = 44{,}15\ N$
- Falling speed
  $$v_D = \sqrt{\frac{2 * m * g}{A * C * \rho}} = 2{,}12\ ms^{-1}$$
- Surface area
  $A = \pi * r^2 = 5{,}3\ m^2$
- Diameter after deployment
  $$D = \sqrt{\frac{8 * m * g}{v_D^2 * \rho * C * \pi}} = 2{,}6\ m$$

Where:

- C – resistance constant
- ρ – air density
- g – gravitational force constant

The parachute test proved that our calculations were correct, at 5 m/s windspeed we measured a force of roughly about 50 Newtons. This means it is possible to use this parachute as a backup landing system in case of emergencies or lack of landing space.

7.2 Communication fault protection

In case of interrupt of communication, the plane could become a hazard for surroundings. To prevent such situations, if communication is interrupted, the flaps are extended to the maximum possible position so the plane is forced in a corkscrew motion sending it directly to the ground instead of flying off to a populated area. Also to prevent damage to the plane and/or injuries to anyone standing beneath the falling plane the safety parachute deploys after entering communication fault mode. This ensures a softer landing.

7.3 Vibrations problems

The most serious problem solved so far is caused by vibrations generated by the engine. These are detected by accelerometers and; in fact, they are so strong that it does not produce any relevant data. The problems are caused by a number of factors, which include:

- Combustion of fuel in the engine cylinder. Since the piston and everything connected to it have some momentum, combustion of fuel causes a force, which forces the engine in the opposite direction of piston movement. This is the cause of Z-axis vibrations.

- Rotation axis vibrations are caused by compression and decompression of gasses in the engine cylinder. This effect causes mild deceleration and acceleration of the propeller thus transferring force into the body of the plane.

The CHR6d module (described in section 6.2) oversamples and decimates the ADC data on all channels to reduce quantization noise and increase the effective ADC resolution to 16 bits. After decimation, sensor data is processed using a configurable Parks-McClellan window FIR (Finite Impulse Response) low-pass filter. The corner frequency of the filter is independently adjustable for each channel from 10 Hz to 140 Hz in 10 Hz increments.
FIR (Finite Impulse Response) filters represent one of two primary types of digital filters used in Digital Signal Processing (DSP) applications. FIR filters sometimes have the disadvantage that they require more memory and/or calculation to achieve a given filter response characteristic. Also, certain responses are not practical to implement with FIR filters.

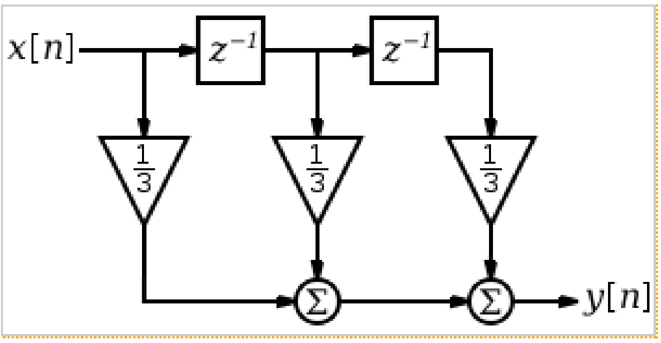


Fig. 6. Block diagram of a simple FIR filter

Tap - A FIR "tap" is simply a coefficient/delay pair. The number of FIR taps is an indication of:

1) The amount of memory required to implement the filter
2) The number of calculations required
3) The amount of "filtering" the filter can do

In effect, more taps means more stop band attenuation, less ripple, narrower filters, etc.

The measurement was done for idle thrust of the engine (25 rps/ 25 Hz). The body of the plane was placed on a fixed, rigid base. During measurement we adjusted values of the low-pass filter and tap value. These values were set to 140Hz for the filter and 8 taps during the first part of measurements and 10Hz 64 taps for the second part.
There is a very noticeable reduction of interference after the second setting on both the accelerometers and gyros.

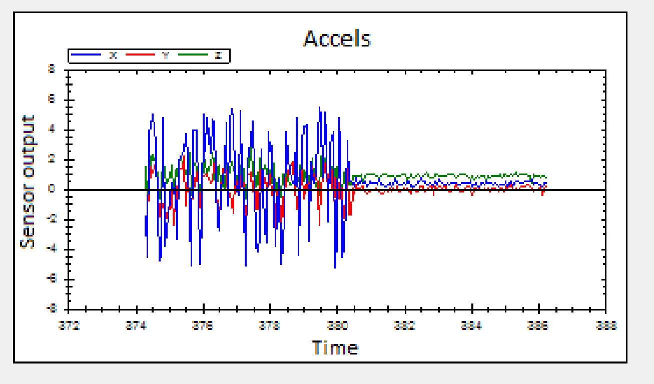


Fig. 7. Accelerometer measurement

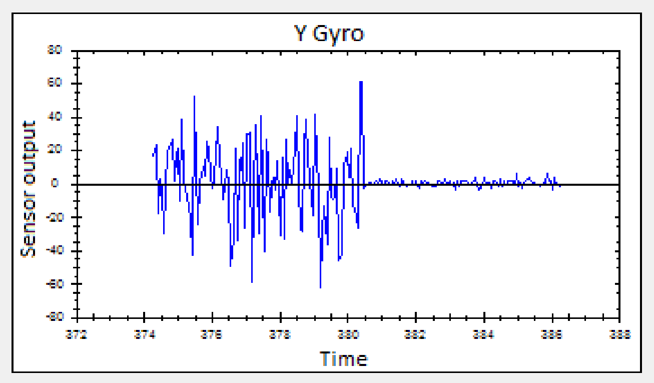


Fig. 8. Gyro measurement

Pitch and roll angles are calculated using an extended Kalman filter, which is a comoponent of the sensor module. Even after filtration there is some interference. This can be caused by imperfection of the whole system (accumulated vibrations in the hull, irregular operation of the engine etc.).
The basic theory behind the filter is this: if the sensor is not accelerating, then the onboard accelerometers can be used to detect gravity and (hence) estimate pitch and roll angles. In general, however, the sensor might be moving around and vibrating, so that in the short term, the accelerometers can't be trusted. This is where the rate gyros come in. MEMS (Microelectromechanical systems) gyro output is angular velocity. In order to get pitch and roll angles we need to integrate this velocity. Since rate gyros are less sensitive to acceleration, they can be used to estimate changes in pitch and roll in the short term. But angle estimates produced by rate gyros tend to drift over time. The onboard Extended Kalman Filter is used to combine accelerometer and rate gyro measurements in a way that removes long-term drift, and that removes the negative effect of transient vibrations. In practice, a tradeoff must be made between trusting the rate gyros and trusting the accelerometers. When the filter trusts rate gyros more, it is less sensitive to acceleration and more sensitive to nonzero gyro biases. On the other hand, if the filter trusts accelerometers more, it is more sensitive to bad acceleration, and less sensitive to nonzero gyro biases.
To get the plane’s absolute position we need to avoid long-term double integration of acceleration values (leading to increasing relative position value errors) by using acquired GPS position data in our calculations. By fusing this data together we get all the information about the plane’s velocity and absolute and relative position we need.
On the CHR6d, the Extended Kalman Filter is tuned by adjusting the process variance. If the sensor were mounted on a rotorcraft, for example, the process variance could be set very low to reduce the effect of vibrations.

8. CONCLUSION

UAV development is directly tied with multiple disciples; mathematics, physics, engineering, telecommunication, IT and many others. The project is conducted in order to achieve fully autonomous flight equipment with no input from humans. In proposing solutions, we note the aircraft hazard to people in case of errors.
We emphasize on the criterion of real-time, safety and transfer the event of failure on backup safety systems.

ACKNOWLEDGMENTS

REFERENCES

[1]http://www.pipeflowcalculations.com/prandtl/

[2]http://www.grc.nasa.gov/WWW/K-12/airplane/pitot.html

[3] http://www.dspguru.com/dsp/faqs/fir/basics

[4] http://en.wikipedia.org/wiki/Finite_impulse_response

[5] http://www.pololu.com/file/0J342/chr6d_datasheet.pdf

Bioproduction of mcl-PHAs Biopolymers – Process States and Control

P. Hrnčiřík* F. Černý* J. Náhlík* J. Vovsík* J. Mareš* P. Lovecká**

** Department of Computing and Control Engineering,
Institute of Chemical Technology Prague, Technická 5, 166 28 Praha 6, Czech Republic
(Tel: +420 220 444 174; e-mail: Pavel.Hrncirik@vscht.cz)
** Department of Biochemistry and Microbiology,
Institute of Chemical Technology Prague, Technická 5, 166 28 Praha 6, Czech Republic
(e-mail: Petra.Lovecka@vscht.cz)*

Abstract: The presented paper deals with the experimental and theoretical work related to mcl-PHAs biopolymer production process using fed-batch cultivations of the bacterium *Pseudomonas putida* KT2442. The focus is on the definition and identification of process states in the form of physiological situations relevant to intracellular biopolymer production as well as the design of appropriate process control strategies.

1. INTRODUCTION

Polyhydroxyalkanoates (PHAs) are microbial polyester polymers synthesized by numerous bacteria as carbon and energy storage material. These polymers are deposited intracellularly in the form of inclusion bodies and can amount up to 90% of dry weight of cells. PHAs belong to promising candidates for biodegradable plastics and elastomers of industrial interest and of environmental value. Potential fields of their application include *e.g.* drug delivery, bone replacement applications, etc.

PHAs are in general classified in two major classes: short-chain-length PHAs (scl-PHAs) with C4-C5 monomers and medium-chain-length PHAs (mcl-PHAs) with C6-C14 monomers. At present, only few PHAs, typically from the structurally simpler class of scl-PHAs are industrially produced. Especially in the case of the more complex mcl-PHAs the disadvantage of the high production cost has prevented its wide use despite their promising properties.

Even though fermentation processes were developed employing different kinds of bacteria to improve PHA productivity there is still considerable room for further improvements, including the area of process control. The work done in this field by ICTP Prague has been concentrated on the development of a physiological state control approach for the production of mcl-PHAs biopolymers by cultivation of the *Pseudomonas putida* KT2442 strain.

2. MATERIALS AND METHODS

The inoculum (bacterium *Pseudomonas putida* KT2442) for fed-batch cultivations was prepared at 30 °C in shaking flasks in a rotary incubator (incubation duration: 16-18 h). The fed-batch cultivation conditions were as follows: temperature at 30 °C, pH = 7, stirrer speed 900 min^{-1}, air flow 9.5 l.min^{-1}. Base (14% NH_4OH) and acid (17% H_3PO_4) solutions were added to the cultivation medium to control pH. Following the initial batch phase octanoic acid was continually supplied as carbon source using feeding strategies described further in the text.

All cultivations were carried out in a 7-litre laboratory bioreactor (newMBR). The bioreactor was equipped with an IMCS 2000 analogue control unit, a programmable logic controller and the proprietary Biogenes II control system. The dissolved oxygen tension was measured by an oxygen probe (Mettler Toledo); the oxygen and carbon dioxide concentrations in the off-gas were measured by SERVOMEX 1100 and 1440 analysers, respectively. For the substrate supply to the bioreactor a DP200 peristaltic pump (New Brunswick) was used. Control variables feeding rate, acid and base were also recorded.

The capacitance measurement reflecting the amount of live biomass in the bioreactor was carried out by an Aber Biomass Monitor 210 (Aber Instruments Ltd., UK) operating in scanning mode and equipped with a highly sensitive four-annular ring probe AberProbe. The measuring device measured both the capacitance spectrum and the capacitance difference ($\Delta C = C(0.47\ \mathrm{MHz}) - C(15.65\ \mathrm{MHz})$). Biomass concentration in the bioreactor was determined off-line gravimetrically as dry cell weight.

The intracellular PHA content was also determined gravimetrically. Biomass for PHA gravimetric determinations was prepared by centrifuging samples (50 ml). The pellet was washed three times with deionized water and lyophilised. PHA was extracted from the lyophilised cells by Soxhlet extraction with hot chloroform (150 ml) for 24 h. Excessive chloroform was then distilled off to obtain cca 5 ml residue. PHA was subsequently precipitated in 10 volumes of cold methanol. The precipitated polymer was separated by decantation, the solvent evaporated to dryness and the purified PHA was then weighed.

In order to characterize quantitatively the process of biopolymer accumulation in the cultivated bacterial cells, several new variables computed from off-line analyses of PHA and biomass concentrations were introduced. Besides standard cumulative productivities and yields of biomass, residual biomass and PHA, there are specifically:

the specific PHA synthesis rate based on the amount of residual biomass - qPHA(Xr), defined as (dPHA/dt)/Xr (g PHA . g Xr^{-1} . h^{-1}),

the specific PHA synthesis rate based on the amount of PHA - qPHA(PHA), defined as (dPHA/dt)/PHA (g PHA . g PHA^{-1} . h^{-1}).

The specific PHA synthesis rate based on the amount of PHA has been found to be particularly useful for the evaluation of the PHA production process and the definition of individual physiological situations (see Figures 1 and 2).

3. RESULTS AND DISCUSSION

3.1 Definition of physiological situations

From the standpoint of industrial operation it is important to note that the type of metabolism used by the cultivated microorganism for the processing of substrates has decisive impact on process performance measured by indicators like productivity and yield. Therefore the design of bioprocess control strategies has to be focused not only on the issue of cell environment control but should ideally also aim at the control of the cell physiology itself. This issue has been addressed by the introduction of a control concept referred to as physiological state control by Konstantinov and Yoshida (1989). In contrast to conventional control strategies operating in closed loop in respect to the cell environment, the physiological state control scheme creates a closed loop in respect to the cell state. Consequently the environment is not a goal but a tool for manipulating cell physiology.

First step in the design of a control scheme based on the concept of the physiological state control is thus the definition and subsequent classification of bioprocess states related to the physiology of the cultivated microorganism – referred to as physiological situations. In the presented case this approach is demonstrated using the experimental data from two cultivations carried out in the Bioprocess Control Laboratory at ICT Prague (named Experiment 100210 and Experiment 100513).

Although the final classification was done for the process of fed-batch cultivations of the strain *Pseudomonas putida* KT2442 grown on octanoic acid for production of mcl-PHAs biopolymers, it is also valid for all fed-batch processes which use toxic substrates and where precise carbon source feeding is necessary.

The physiological situations are defined according to the feeding quality with respect to the state of the cells themselves (see Table 1).

Name	Description	Definition
BATCH	Batch or Return from Overfeeding	Fm = 0, DO ↓, DO spectrum I
OPT	Optimal feeding (optimal growth with optimal substrate utilization)	Fm > 0, DO ↓, DO spectrum II
OF	Overfeeding (growth inhibited owing to excessive substrate concentration)	Fm > 0, DO ↑, DO spectrum I
UF	Underfeeding (growth limited owing to lower substrate concentration)	Fm > 0, DO ↑, DO spectrum II
DOLIM	DO limitation	2 % < DO < 10 %, CO2 ↑
DOZER	DO zero or DO near zero value	DO < 2 %, CO2 ↓

Tab. 1. Physiological situations definitions (Fm – carbon source feedrate, CO2 – carbon dioxide offgas concentration, DO – dissolved oxygen concentration, DO spectrum I, II – different spectra of the dissolved oxygen concentration signal)

Two different DO limitation situations were introduced, because different DO limitations were observed to have different effect on cell growth and PHA production respectively. The situations can be estimated on-line from the carbon source feeding rate Fm and the trends of DO and CO2 variables.

In the two above mentioned experiments 100210 and 100513 the physiological situations were identified as follows from Table 2 and Table 3 and depicted in Fig. 1 and Fig. 2.

Cultivation Time (min)	Physiological Situation
0-250	BATCH
255-405	OPT
405-565	OF
565-750	BATCH
750-790	UF
790-1180	OPT
1180-1380	UF
1380-1440	OPT
1440-1580	UF
1580-1660	OPT
1660-1850	DOLIM
1850-2040	DOZER

Tab. 2. Physiological situations - Experiment 100210

Cultivation Time (min)	Physiological Situation
0-320	BATCH
320-420	OPT
420-520	OF
520-600	BATCH
600-650	UF
650-1560	OPT
1560-2070	DOZER

Tab. 3. Physiological situations - Experiment 100513

From Figures 1 and 2 it can be seen that optimal situation for mcl-PHA production according to the specific PHA synthesis rate based on the amount of PHA - variable qPHA(PHA) – is Optimal feeding, during which this variable is increasing. On the other hand this variable is decreasing in Overfeeding and also Underfeeding situations, not to mention DO limitations.

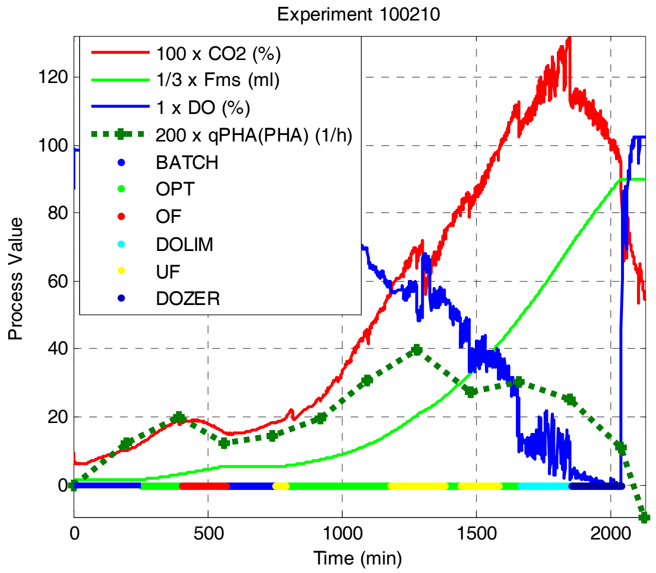


Fig. 1. Physiological situations during Experiment 100210

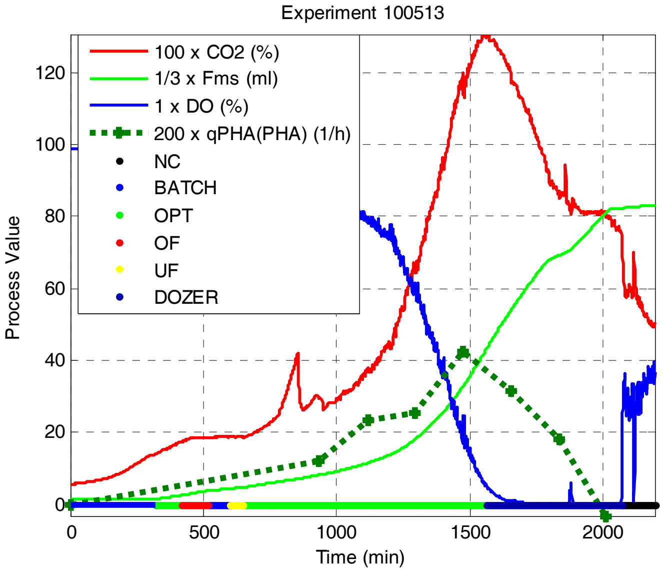


Fig. 2. Physiological situations during Experiment 100513

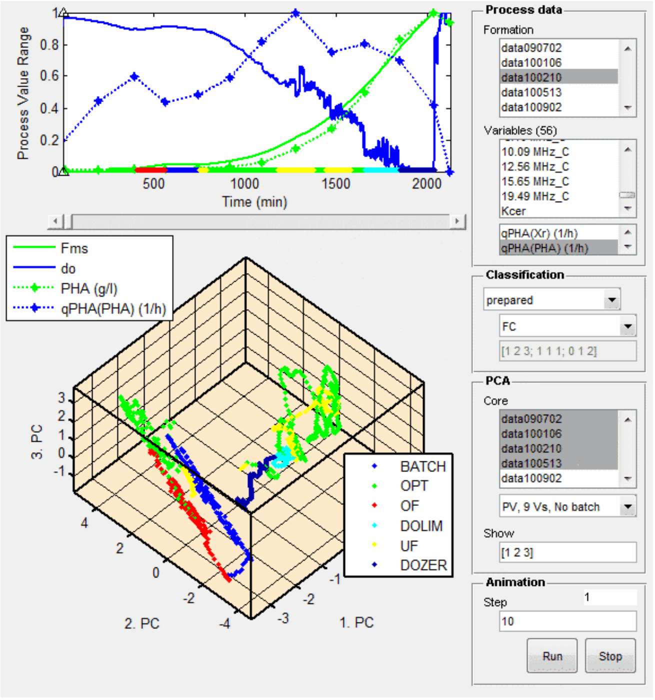
Both experiments were also visualized and analysed in the 3D space of the first three principal components (see Figures 3 and 4). These components were computed using Principal Component Analysis (PCA) from nine selected physiological variables (specific uptake/production rates and rate ratios), without significant loss of information about the process. The selected variables are all of the ratio type and were chosen from the complete set of computed physiological variables to eliminate the time trend effect on the fermentation trajectory depiction in the PCA space.

It can be observed that during fed-batch cultivations the fermentation goes through different physiological states (points in the space) even if it remains in the same physiological situation, as e.g. Optimal feeding.

Summary of the results of both experiments is presented in Table 4.

Experiment:	100210	100513
Cell concentration $(g \cdot l^{-1})$	64.53	42.34
PHA concentration $(g \cdot l^{-1})$	33.26	25.50
PHA content (%)	54.94	65.19
Overall PHA yield $(g \cdot g^{-1})$	0.48	0.56
Overall PHA productivity $(g \cdot l^{-1} \cdot h^{-1})$	3.36	3.67

Tab. 4. Experimental Results – Summary of attained maxima



Process data
Formation
data090702
data100106
data100210
data100513
data100902

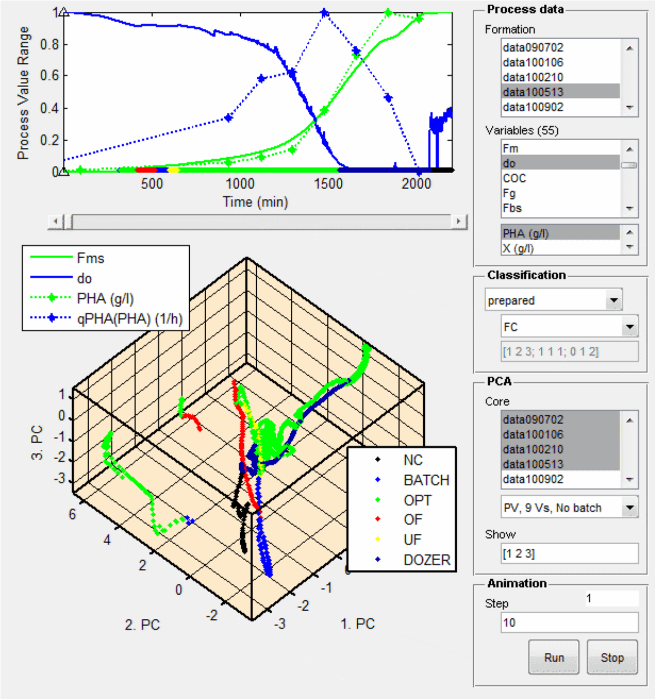
Variables (56)
10.09 MHz_C
12.56 MHz_C
15.65 MHz_C
19.49 MHz_C
Kcer
qPHA(Xr) (1/h)
qPHA(PHA) (1/h)

Classification
prepared
FC
[1 2 3; 1 1 1; 0 1 2]

PCA
Core
data090702
data100106
data100210
data100513
data100902
PV, 9 Vs, No batch
Show
[1 2 3]

Animation
Step 1
10

Fig. 3. Physiological situations during Experiment 100210 (PCA)



Formation: data090702, data100106, data100210, data100513, data100902

Variables (55): Fm, do, COC, Fg, Fbs; PHA (g/l), X (g/l)

Classification: prepared; FC; [1 2 3; 1 1 1; 0 1 2]

PCA Core: data090702, data100106, data100210, data100513, data100902; PV, 9 Vs, No batch

Show: [1 2 3]

Step: 1; 10

Fig. 4. Physiological situations during Experiment 100513 (PCA)

3.2 Process control strategies

Microbial bioproduction processes are traditionally mostly operated in fed-batch mode capable of achieving high-cell density cultures in an efficient way. The key issue related to the design of such processes is the choice of suitable substrate feeding strategies to control key nutrient concentrations, being in most cases the carbon source. The ideal solution, i.e. control strategies based on a direct measurement of the particular key nutrient concentration, is in many cases not feasible due to the limited availability of the necessary analytical devices. The carbon source used in the mcl-PHAs fed-batch bioproduction - octanoic acid - is a typical example of such a case, where sensor system suitable for routine on-line measurement of its concentration is not readily available. A common solution to this problem is often the use of predetermined exponential feeding profiles, i.e. feeding recipes calculated apriori from historical data with the aim to achieve a prescribed cell growth rate. However, the main drawback of this approach is its lack of flexibility vis-a-vis unexpected process events, since it cannot compensate for process disturbances that may severely affect the cell physiology. More appropriate are therefore control strategies where the key nutrient feeding is based on cell physiology, typically via one or several physiological variables that are calculable or even directly measurable on-line.

Specifically for the mcl-PHAs fed-batch bioproduction process, two such new process control strategies for the phases of optimal cell growth (CER agent) and oxygen limitation (DOPID) respectively have been designed and implemented.

CER Agent - control strategy description

- carbon-source (octanoic acid) feeding strategy based on CO2 evolution (production) rate CER (Eq. 1).
- $$C-source\_\ feeding\_\ rate = k_{CER}\cdot CER \qquad (1)$$
  $$\text{where } k_{CER} \approx \frac{1}{Y_{CO_2/C-source}}$$
- based on similar strategy applied by Sun et al. (2006) in high-cell-density fed-batch cultivation of *P. putida* KT2440 with glucose as the carbon source
- contrary to common feeding strategies based on predetermined exponential feeding profile this strategy is able to adjust the feeding profile in response to the variability of individual bacterial cultures
- using this strategy both high mcl-PHAs concentration and content have been achieved
- the strategy can be tuned by just one parameter - kCER, which is adjusted by process operator using a process state classification scheme (based on CO2, DO online measurements and carbon-source feeding rate), automation of this adjustment strategy is envisaged. This new classification scheme turned out to be better (more robust and easier to implement on-line) than the classification scheme based on the so called “obesity quotient” studied previously (Hrnčiřík et al. 2010).

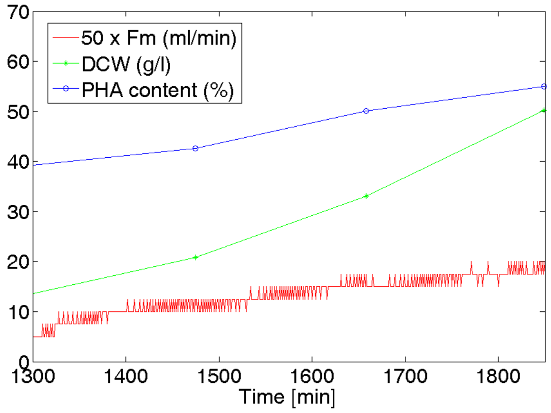


Fig. 5. Cell growth controlled by the CER agent strategy (Experiment 100210, Fm = substrate feeding rate, DCW = dry cell weight, PHA content = intracellular biopolymer content)

DOPID - control strategy description

- carbon-source (octanoic acid) feeding strategy based on dissolved oxygen (DO) concentration
- aiming to stabilize the DO level above critical values to avoid excessive carbon-source overfeeding that might lead to inhibiting accumulation of the carbon source (octanoic acid) in the cultivation medium
- switched to from the CER agent strategy after oxygen limitation sets in (at approx. DO=15%)
- implementation by a standard PID control loop
- oscillatory behaviour of the controlled variable (DO) serves as an immediate indicator of an adequate feeding level

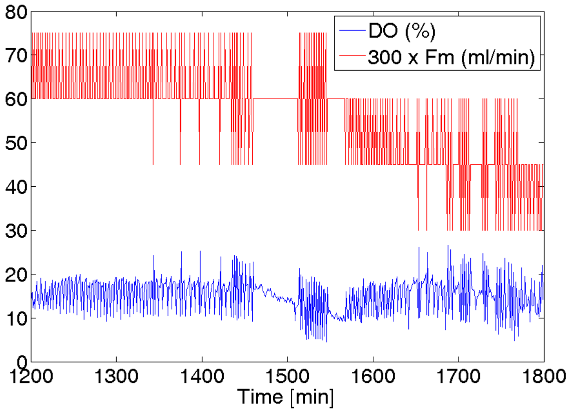


Fig. 6. Process control during the oxygen limitation phase - DOPID strategy (Experiment 110317, DO = dissolved oxygen concentration, Fm = substrate feeding rate)

3.3 Conclusion

Using these strategies in combination with the classification of physiological situations it was possible to maintain the process in the optimal feeding regime just on the border between underfeeding and overfeeding situations, thereby bringing about the optimal exploitation of cell growth potential and PHA production capacity.

ACKNOWLEDGMENTS

This work was supported by the 6th Framework Programme of the European Community under contract No. NMP2-CT-2007-026515 "Bioproduction project - Sustainble Microbial and Biocatalytic Production of Advanced Functional Materials" and by the fund No. MSM 6046137306 of the Czech Ministry of Education. This support is very gratefully acknowledged.

REFERENCES

Hrnčiřík P., Vovsík J., Náhlík J. (2010). A New On-Line Indicator of Biopolymer Content in Bacterial Cultures. In: *Proceedings of the 11th IFAC Symposium on Computer Applications in Biotechnology*. pp. 192-196, IFAC. Leuven.

Konstantinov, K.B., Yoshida, T. (1989). Physiological state control of fermentation processes. *Biotechnology and Bioengineering* **33**, 1145-1156

Sun Z., Ramsay J., Guay M., Ramsay B. (2006). Automated feeding strategies for high-cell-density fed-batch cultivation of Pseudomonas putida KT2440. *Applied Microbiology and Biotechnology* **71**, 4, 423-431.

Structure and Parametric Definition of Linear Dynamic Object via Identification Based on Real Interpolation method

A.S. Alexeev* S.V. Zamyatin M.I. Pushkarev*****

** Tomsk Polytechnic University, Russia, Tomsk, Lenina street 30*
(Tel: +7-3822-701837; Fax: +7-3822-420-588; e-mail: alekseyev@sibmail.com)
*** Tomsk Polytechnic University, Russia, Tomsk, Lenina street 30*
(e-mail: zamsv@tpu.ru)
**** Tomsk Polytechnic University, Russia, Tomsk, Lenina street 30*
(e-mail: maxebbc@mail.ru)

Abstract: The paper considers identification problem of linear dynamic object based on real interpolation method. It is provided the solution of raised problem that gives formalized algorithm for structure and parameters definition using as input data step-response function. Suggested approach gives possibility to build transfer function of analyzed object with pre-determinate level of relevant error in time domain. Numerical example is provided.

Keywords: Identification, transfer function, real interpolation method.

1. INTRODUCTION

Identification problem plays significant role in automation control system design, operation and diagnosis. Mathematical model type and plant parameters determination enables to design and adjust controllers with a high accuracy, to create self-tuning systems etc.

It is convenient to use transfer functions for describing mathematical models of linear object. In this case identification problem consists of two subtasks. The first one is called structure identification and includes mathematical model structure determination, in particular, polynomial degrees determination of transfer function numerator and denominator. The second subtask is parametric identification. It includes transfer function coefficients definition of given structure.

For the case, when the dynamics of an analyzed object could be described with the help of linear differential equations, there exist quite a large number of solutions for identification problem (Hildebrand et al. 2007), (Agüero et al. 2010) and (Liu et al. 2010).

Real interpolation method showed high efficiency in parametric identification problem solution. However there is no formalized and algorithmically efficient solution of structure identification problem within this method. For example, in the work (Rudnitsky et al. 2008), the single structure according to Ishlinsky rule is chosen for all plants. Such approach introduces excessive redundancy in case of objects identification, which is described by low order differential equations, and do not allow reproducing specific characteristics of 4th and higher order objects behavior. In the paper (Antropov et al. 2004) it is offered to identify transfer function structure by enumerative technique, starting from the first order object with the further increase of denominator and numerator polynomial orders. The order is increasing until error in a time domain reaches specified level. Such approach is rather effective, but it is desirable to reduce number of considered variants of transfer function structures.

The main goal of this paper is to suggest more efficient algorithm for structure identification of transfer function such as degree orders of numerator and denominator using step response function as input information. The raised problem requires decreasing number of iterations in case of enumeration of structure parameters such as degree orders of numerator and denominator.

In the paper is considered new approach for structure and parametric identification, which is based on real interpolation method.

2. BASIS OF REAL INTERPOLATION METHOD

It is known that signal images could be obtained on the basis of appropriate real transformation, based on the direct Laplace transform formula

$$F(p)=\int_0^\infty f(t)e^{-pt}dt, \qquad (1)$$

where $f(t)$ is an original function, $F(p)$ – Laplace representation of given function, $p=\delta+j\omega$ – complex transformation variable. There are several important things from point of mathematical description and automation control system design. Functions $F(p)$ are images and therefore their usage is more preferable in contrast with time functions $f(t)$. For instance, time function differentiation operation $f(t)$ in the Laplace domain, corresponds to function multiplication operation in case of zero initial conditions $F(p)=L\{f(t)\}$ by variable *p*. Integration in the original domain corresponds to the division of function $F(p)$ by variable p.

Taking into consideration the fact that real interpolation method operates in a real domain, the formula of such transformation could be obtained by replacement of transformation complex variable p for real δ in the expression (1)

$$F(\delta)=\int_0^\infty f(t)e^{-\delta t}dt. \qquad (2)$$

The most important moment when describing automation control system from mathematical point of view while obtaining the result of expression (2) is the lack of imaginary unit. Conditions of function $F(\delta)$ existence and uniqueness are defined by integral convergence (2). Therefore variable δ is constrained:

$$F(\delta)=\int_0^\infty f(t)e^{-\delta t}dt,\ \delta\in[C,\infty),\ C\geq 0. \qquad (3)$$

Expression (3) is a formula of direct integral real transformation or δ-transformation.

In application to linear automation control system, when $f(t)$ represents its dynamic characteristic of plant, convergence (3) is provided by choosing an appropriate parameter *C* value. For stable system with an impulse step response *C* may be assumed as 0.

In literature (Goncharov 1995) this method is known as real interpolation method (RIM)

3. PARAMETRIC IDENTIFICATION OF DYNAMIC OBJECT

To obtain image functions $F(\delta)$ by time functions $f(t)$ it is possible to use the method, which is based on direct δ-transformation formula (3). This method of obtaining mathematical description of signals is convenient to use in case when original function $f(t)$ in analytical form is known or such function could be obtained by means of experimental tabular data interpolation.

In the general case function $W(\delta)$ can be obtained on the basis of transfer function definition, as output signal image $W(\delta)$ to input signal image $X(\delta)$ ration (in case of zero initial conditions)

$$W(\delta)=\frac{Y(\delta)}{X(\delta)}=\int_0^\infty y(t)e^{-\delta t}dt \Big/ \int_0^\infty x(t)e^{-\delta t}dt\,. \qquad (4)$$

The «input-output» correlation takes on a standard form $Y(\delta)=W(\delta)X(\delta)$.

The most suitable form of function approximation on infinite intervals of an argument changing is fractionally rational representation of the following type

$$W(\delta)=\frac{b_m\delta^m+b_{m-1}\delta^{m-1}+...+b_1\delta+b_0}{a_n\delta^n+a_{n-1}\delta^{n-1}+...+a_1\delta+1},\ n\geq m\,. \qquad (5)$$

The possibility of obtaining real function, based on expression (4) creates favorable preconditions for solving practical and research problems while designing automation control systems via RIM. Realization of these facilities using computers demands transition from analytical expressions to numerical sequences, with further transition from one form of representation to another.

For this aims it is offered in RIM to implement continuous functions $F(\delta)$ discretization, performing their following restoration in continuous form with the help of interpolation. For real function $F(\delta)$, $\delta\in[0,\infty)$, in a node system δ_i, $i=1,2,3...$ a set of values $F(\delta_i)$, $i=\overline{1,\eta}$ is formed, where η – numerical characteristic dimension:

$$\{F(\delta_i)\}_\eta=\{F(\delta_1),F(\delta_2),...,F(\delta_\eta)\}\,. \qquad (6)$$

It is necessary to define the disposition interval and distribution law, when choosing the interpolation nodes. In general case this task do not have any exact solution, therefore uniform distribution law of nodes is used very frequently. Coverage of the domain where the most essential changes for function $F(\delta)$ happen is a requirement made to the interval $\delta\in[\delta_1,\delta_\eta]$.

In the general case, when the inequality $m\leq n$ is hold for the polynomials order of a numerator m and denominator n of function (5), then for δ_η node estimation the following equation is considered:

$$F(\delta_\eta)=(0{,}1\div 0{,}2)[F(0)-F(\infty)]+F(\infty)\,. \qquad (7)$$

The solution of equation (7) is found numerically with the iterative procedures.

Nodes position inside the $[\delta_1,\delta_\eta]$ interval for analytical grid is determined in a following way:

$$\delta_i=\delta_1+\frac{\delta_\eta-\delta_1}{\eta-1}(i-1),\ i=\overline{1,\eta}\,. \qquad (8)$$

Then «input-output» equation for automatic control system design with the real images involvement, written in a form

$$Y(\delta)=W(\delta)X(\delta)\,, \qquad (9)$$

could be represented as a correlation between numerical characteristic elements $\{X(\delta_i)\}_\eta$, $\{Y(\delta_i)\}_\eta$, $\{W(\delta_i)\}_\eta$ of input $x(t)$ and output $y(t)$ signals with a transfer function $W(\delta)$:

$$\{Y(\delta_i)\}_\eta = \{W(\delta_i)\}_\eta \{X(\delta_i)\}_\eta .$$

The usage of a numerical representation for the system signal in aggregate with low operation number turns to be a positive characteristic of the latter equation in contrast to analytical representation (9).

Connection between the model in form of numeric characteristic and real transfer function in form (5) is established by means of linear algebraic equations system

$$W(\delta_i) = \frac{b_m\delta_i^m + b_{m-1}\delta_i^{m-1} + ... + b_0}{a_n\delta_i^n + a_{n-1}\delta_i^{n-1} + ... + a_1\delta_i + 1}, \; i = \overline{1,\eta} . \qquad (10)$$

The solution of (10) is found in the following form

$$\begin{cases} b_m\delta_1^m + b_{m-1}\delta_1^{m-1} ... + b_0 - a_n\delta_1^n W(\delta_1) - ... - a_1\delta_1 W(\delta_1) = W(\delta_1), \\ b_m\delta_2^m + b_{m-1}\delta_2^{m-1} ... + b_0 - a_n\delta_2^n W(\delta_2) - ... - a_1\delta_2 W(\delta_2) = W(\delta_2), \\ ... \\ b_m\delta_\eta^m + b_{m-1}\delta_\eta^{m-1} ... + b_0 - a_n\delta_\eta^n W(\delta_\eta) - ... - a_1\delta_\eta W(\delta_\eta) = W(\delta_\eta). \end{cases}$$

As a result of linear algebraic equations system solution, unknown transfer function $W(\delta)$ coefficients will be found. And their number will be equal to parameter η. For the case of representation a transfer function in form (5) the value of this parameter is $\eta = n + m + 1$, that provides linear algebraic equation system (10) solution unicity. The transition from real transfer function $W(\delta)$ to Laplace representation is realized by formal substitution of complex transformation variable $p \to \delta$ in transfer function expression.

The principal unsolved problem at this stage is a choice of the transfer function structure. That is numerical value n and m determination in expression (5). Let's consider this problem deeper.

4. IDENTIFICATION STRUCTURE OF TRANSFER FUNCTION

Transfer function structure problem, that is numerator m and denominator n polynomial order, can be solved by the method offered in paper (Shalaev 2005). Let's use equation (5) and extreme correlation

$$\lim_{\delta\to\infty} \frac{W(\delta)}{W(g\cdot\delta)} = g^{n-m}, \qquad (11)$$

Where $g > 1$ is a real number. From the resulting correlation (11) structure parameters estimation is obtained

$$\gamma = \frac{\ln(g^{n-m})}{\ln(g)} = \widehat{n-m} . \qquad (12)$$

Real number is obtained with a help of formula (12) that contains integer and fractional parts. Let's assume fractional part as one unit and add to integer part.

Unfortunately, it is rather difficult to define the limit (11) analytically, because the expressions for $W(\delta)$ and $W(g\cdot\delta)$ are defined according to the formula (4). In connection with this it is offered to restrict an interval $W(\delta)$ of essential function changes and consider not extreme correlation (11), but the expression

$$\frac{W(\delta_\eta)}{W(g\cdot\delta_\eta)} \cong g^{n-m} , \qquad (13)$$

where node δ_η is defined by solution of the equation (7) with substitution $\delta \to g\cdot\delta_\eta$, since denominator transfer function (12) changes g times faster, than in numerator. Then magnitude δ_η specifies right interval boundary $[\delta_1, \delta_\eta]$ of interpolation nodes distribution δ_i, $i = \overline{1,\eta}$. Having a single step response and taking into account extreme correlations $\lim_{\delta\to 0} W(\delta) = \lim_{t\to\infty} h(t)$ and $\lim_{\delta\to\infty} W(\delta) = \lim_{t\to 0} h(t)$, it is possible to write in the right part of expression (7)

$$W(g\delta_\eta) = (0{,}1 \div 0{,}2)[h(\infty) - h(0)] + h(0) , \qquad (14)$$

where $h(\infty)$, $h(0)$ – final and initial step response values $h(t)$ correspondingly. Then γ evaluation is found as a result of equation (14) and further calculations for expressions (13) and (12). With the help of obtained structure parameters estimation γ it is possible to form the transfer function structure identification algorithm. To make this, it is necessary to express numerator m order through denominator n order and estimation value γ

$$m = n - \gamma , \; n = \begin{cases} 1, 2, ... \text{ if } \gamma = 0, \\ \gamma, \gamma+1, ... \text{ if } \gamma \neq 0. \end{cases} \qquad (15)$$

Value n is a free argument in the last equation. The existence of the fixed parameter γ, allows to get rid of examination a set of transfer function structures, which do not meet the requirements of equation (15). Parameter n value enumeration should be continued until relative identification error satisfies specified criterion in the time domain.

5. NUMERICAL EXAMPLE

Let's examine structure and parametric identification approach. As a result of the experiment, the signal $x(t) = 1(t)$ (with zero initial conditions) was feed to the object input. Object response $y(t) = h_{obj}(t)$ is represented on figure 1.

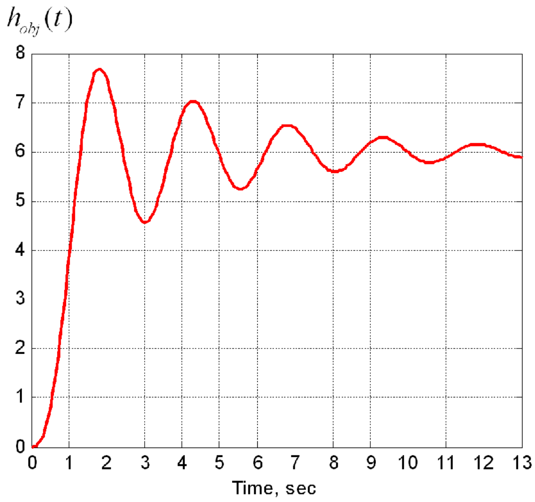


Fig. 1. Step-response of given object

For calculating $W_{obj}(\delta)$ in expression (4) the numerical integration formula is used (Collins 2003)

$$W_{obj}(\delta)=\frac{\left(\dfrac{h_{obj}(t_0)e^{-\delta t_0}+h_{obj}(t_k)e^{-\delta t_k}}{2}+\sum_{j=1}^{k-1}h_{obj}(t_j)e^{-\delta t_j}\right)\cdot\Delta T}{\left(\dfrac{1(t_0)e^{-\delta t_0}+1(t_k)e^{-\delta t_k}}{2}+\sum_{j=1}^{k-1}1(t_j)e^{-\delta t_j}\right)\cdot\Delta T},$$

where ΔT – input and output signal sampling period accordingly, t_j = $j\cdot\Delta T$ – current moment of time, $j=\overline{0,k}$ $(k=59)$. Let's assume ΔT = 0.22 sec. Notice that experimental data were obtained on the interval $t\geq 0$, so that function 1(t) corresponds to the constant value 1(t) = 1, in this example. Taking into account simplifications, the latter expression could be represented in the form of

$$W_{obj}(\delta)=\frac{\dfrac{h_{obj}(t_0)e^{-\delta t_0}+h_{obj}(t_k)e^{-\delta t_k}}{2}+\sum_{j=1}^{k-1}h_{obj}(t_j)e^{-\delta t_j}}{\dfrac{e^{-\delta t_0}+e^{-\delta t_k}}{2}+\sum_{j=1}^{k-1}e^{-\delta t_j}}.$$

Further let's define interval boundaries, inside which interpolation nodes will be placed. Since object is stable, it's possible to assume δ_1=0 for the left bound. Right bound will be defined by node δ_η placement. Its value will be found from equation (14)

$$W_{obj}(g\cdot\delta_\eta)=0.2\cdot[h_{obj}(t_k)-h_{obj}(t_0)]+h_{obj}(t_0)\ .$$

Let's place real transfer function $W_{obj}(\delta)$ expression in a left side of equation, and then turn to equation examination

$$\frac{\dfrac{h_{obj}(t_0)e^{-g\delta_\eta t_0}+h_{obj}(t_k)e^{-g\delta_\eta t_k}}{2}+\sum_{j=1}^{k-1}h_{obj}(t_j)e^{-g\delta_\eta t_j}}{\dfrac{e^{-g\delta_\eta t_0}+e^{-g\delta_\eta t_k}}{2}+\sum_{j=1}^{k-1}e^{-g\delta_\eta t_j}}= \qquad (16)$$
$$=0.2\cdot[h_{obj}(t_k)-h_{obj}(t_0)]+h_{obj}(t_0).$$

After accepting a parameter value as $g=2$ (Shalaev 2005) and using numeric values for the last equation, it is possible to find the solution of the equation (16). In order to solve equation (16) iterative procedure is used. For given numeric values the solution is $\delta_\eta=2.15$ with an accuracy of $|\,\varepsilon\,|\leq 0{,}01$.

Let's implement substitution (13) to (12) and find structure estimation

$$\gamma=\ln\left(\frac{W_{obj}(\delta_\eta)}{W_{obj}(g\cdot\delta_\eta)}\right)\frac{1}{\ln(g)}=\ln\left(\frac{W_{obj}(2.15)}{W_{obj}(2\cdot 2.15)}\right)\frac{1}{\ln(2)}=1.88.$$

Finally according to recommendation given earlier let's assume $\gamma=2$. Then according to (15) let's write

$$m=n-2,\ \ n=2,3,\ldots\,. \qquad (17)$$

The last expression provides information about relative degree of transfer function model $W_m(\delta)$.

Nodes system is formed according to expression (8). Then while taking into account structure parameter estimation linear algebraic equation system (10) is composed. The solution gives possibility for defining transfer function model $W_m(p)$ parameters. As proximity criterion for identified transfer function $W_m(p)$ and object transfer function $W_{obj}(p)$ the estimation in time domain is used in the following form

$$I=\max_{t\in[0,T]}\left(100\frac{|h_{obj}(t)-h_m(t)|}{h_{obj}(t)}\right)\leq E\ ,$$

where $h_m(t)$ – step response of transfer function $W_m(p)$, $T=13\,\text{sec}$ – observation time for given object, E = 5 % – maximum identification error.

Using expression (16) it is possible to make a conclusion that transfer function identification procedure should be started from the second order model with m = 0 degree of numerator and n = 2 degree of denominator. If the identification error isn't satisfy proximity criterion then more complex structure of transfer function $W_m(\delta)$ should be used. According to expression (17) denominator degree is n = 3 and numerator degree is $m=1$. This iterative process is continuing until identification error will reach a pre-determinate level.

Let's demonstrate how to compose equation system (10) for the transfer function $W_m(\delta)$ with with m = 0 degree of numerator and n = 2 degree of denominator:

$$\begin{cases} W_{obj}(\delta_1) = \dfrac{b_0}{a_2\delta_1^2 + a_1\delta_1 + 1}, \\ W_{obj}(\delta_2) = \dfrac{b_0}{a_2\delta_2^2 + a_1\delta_2 + 1}, \\ W_{obj}(\delta_3) = \dfrac{b_0}{a_2\delta_3^2 + a_1\delta_3 + 1}, \end{cases}$$

where interpolation nodes are performed on the base of formula (8): $\delta_1 = 0$, $\delta_2 = 1.07$, $\delta_3 = 2.15$. The solution is: $b_0 = 5.9$, $a_2 = 0.63$, $a_1 = 0.47$. The value of proximity criterion is $I = 72.3$ %. Given solution doesn't satisfy pre-determinate level of error E. It means that structure of transfer function $W_m(\delta)$ should be more complex.

The results of identification for different structures of transfer function $W_m(\delta)$ are presented in table 1.

Table 1. Identification results

Degree	Transfer function $W_m(p)$	I, %
$m = 0$ $n = 2$	$\dfrac{5.986}{0.629p^2 + 0.469p + 1}$	72.3
$m = 1$ $n = 3$	$\dfrac{-0.547p + 5.986}{0.036p^3 + 0.279p^2 + 0.597p + 1}$	27.7
$m = 2$ $n = 4$	$\dfrac{0.068p^2 + 4.695p + 5.986}{0.107p^4 + 0.292p^3 + 0.909p^2 + 1.543p + 1}$	4.93

After analyzing the results we could see that appropriate approximation was reached with the proximity criterion value $I = 4.93$ %, for transfer function

$$W_m(p) = \frac{0.068p^2 + 4.695p + 5.986}{0.107p^4 + 0.292p^3 + 0.909p^2 + 1.543p + 1}.$$

On the figure 2 step-response functions for this model $h_m(t)$ and object $h_{obj}(t)$ are represented.

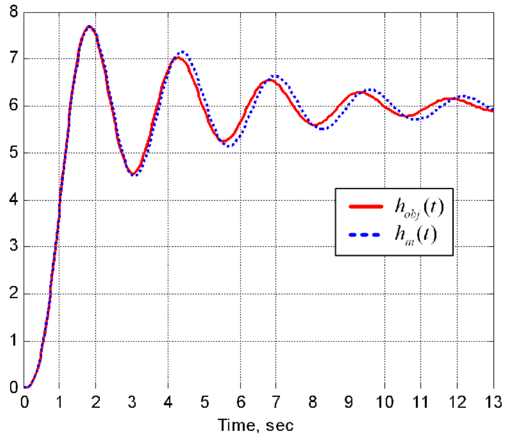


Fig. 2. Step-responses of object and identified model

Thus in case of simple enumeration we have to consider transfer functions with following combinations of numerator and denominator degrees: (m = 0, n = 2), (m = 1, n = 2), (m = 2, n = 2), (m = 0, n = 3), (m = 1, n = 3), (m = 2, n = 3), (m = 3, n = 3), (m = 0, n = 4), (m = 1, n = 4) and (m = 2, n = 4). Instead of considering ten different variant of transfer functions for parameters estimation it was used only three structures, which were determinate according to expression (16). This obviously demonstrates that proposed identification procedure provides rather efficient way of model estimation for linear dynamic object.

6. CONCLUSION

This paper presents a new method of structure and parametric identification of linear dynamic systems based on RIM which was improved with more efficient algorithm of structure definition. The efficiency of the method was shown on numerical example. The proposed identification procedure operates with signals in time domain and uses step-response function as input information for further processing. Whereas traditional identification methods (as recursive least squares method) provides only parametric approximation for given function template, the new identification procedure in addition to parameters definition gives possibility of model structure estimation.

ACKNOWLEDGMENTS

The work has been supported by the Ministry of Education and Science of the Russian Federation under FTP “Scientific, research and educational cadres of innovative Russia” for 2009-2013.

REFERENCES

Agüero, J.C., Yuz, J.I., Goodwin, G.C., Delgado R.A. (2010). On the equivalence of time and frequency domain maximum likelihood estimation. In: *Automatica*. **Vol. 46.** pp. 260-270.

Antropov, A.T., Udod, A.S. (2004). Plant identification via real interpolation method. In: *Proc. of the 2nd Conf. “Engineering design and scientific application in MATLAB”*. IPU RAN Press. Moscow.

Collins, G.W. (2003). *Fundamental numeric methods and data analysis*. Case Western Reserve University.

Goncharov, V.I. (1995). *Real interpolation method in problems of automatic control*. TPU Press. Tomsk.

Hildebrand, R., Solari G. (2007). Identification for control: Optimal input intended to identify a minimum variance controller. In: *Automatica*. **Vol. 43.** pp. 758-767.

Liu, T., Gao, F. (2010). Closed-loop identification of integrating and unstable processes. In: *Chemical Engineering Science*. **Vol. 65.** pp. 2884-2895.

Rudnitsky, V.A., Goncharov ,V.I. (2008). Tools of Control Object Identification. In: *System Science*. **Vol. 34**. pp. 43-47.

Shalaev, Y.N. (2005). The Estimation of Parameters of Dynamic system by a Method of Image Vectors. In: *9th Korean-Russian International Symposium on Science & Technology (KORUS)*. **Vol. 1.** pp. 694-695. NSTU Press. Novosibirsk.

Mathematical model of differentially steered mobile robot

F. Dušek* D. Honc* P. Rozsíval**

** Department of Process Control, Faculty of Electrical Engineering and Informatics, University of Pardubice
nám. Čs. legií 565, 53210 Pardubice, Czech Republic
(Tel: +420 466 037 125; e-mail: frantisek.dusek@upce.cz)
** Department of Electrical Engineering, Faculty of Electrical Engineering and Informatics, University of Pardubice
(e-mail: pavel.rozsival@upce.cz)*

Abstract: Paper deals with dynamic mathematical model of an ideal differentially steered drive system (mobile robot) planar motion. The aim is to create model that describes trajectory of a robot's arbitrary point. The trajectory depends on supply voltage of both drive motors. Selected point trajectory recomputation to trajectories of wheels contact points with plane of motion is a part of the model, too. The dynamic behaviour of engines and chassis, form of coupling between engines and wheels and basic geometric dimensions are taken into account. The dynamic model will be used for design and verification of a robot's motion control in MATLAB / SIMULINK simulation environment.

1. INTRODUCTION

Paper deals with dynamic model of an ideal mobile robot with differentially steered drive system and planar motion. Single-axle chassis or caterpillar chassis is mostly used in case of small mobile robots (Novák 2005). Caster wheel is added to single-axle to ensure stability. This solution together with independent wheel actuation allows excellent mobility on the contrary to a classic chassis – see commercially available robot in Fig. 1. Derived mathematical model comes from lay-out, nominal geometric dimensions and other features of that robot with view of ideal behaviour of individual components and some simplifying assumptions. The aim is to create model based on forces caused by engine moments of independent wheel drives. Model will consist of dynamic behaviour description of chassis and in series connected DC motors. Presented motion model based on centre of mass (primary element) dynamics is different from models reflecting kinematics only and commonly used in literature – published e.g. in (Stengel 2010) or (Lucas 2010). Standard models describe robot's trajectory time evaluation depending up to known wheel speed (information from wheel speed sensors) and chassis geometry - odometry – published e.g. in (Winkler 2010). Our model extends standard model with dynamic part describing wheel speed dependency on motor supply voltage by respecting dynamics, construction, geometry and other parameters of chassis and motors.

Motor supply voltage actuating the wheel causes driving torque and thereby wheel rotation. Inertial and resistance forces act against driving torque. Both driving torques influence each other because of these forces. Planar curvilinear motion of the robot is result of various time variant wheel rotation speeds.

Planar curvilinear motion can be decomposed to a sum of linear motion (translation) and rotation motion. Forces balance is starting point for the derivation of motion equations. If F is actual force acting to a mass point with weight m and with distance r from axis of rotation then it holds for general curvilinear motion that vector sum of all forces acting to a selected point is zero - see literature (Horák et al. 1976)

$$\vec{F} + \underbrace{m\frac{d\vec{v}}{dt}}_{inertial\ force} + \underbrace{m\frac{d\vec{\omega}}{dt}\times r}_{Euler's\ force} + \underbrace{2m\vec{\omega}\times\frac{d\vec{r}}{dt}}_{Coriolis\ force} + \underbrace{m\vec{\omega}\times\left(\vec{\omega}\times\vec{r}\right)}_{centrifugal\ force} = 0 \qquad (1)$$

Application of this general equation requires specification of individual forces according to actual conditions and/or eventually implementing other acting forces. We will consider forces originated by motion of real body – induced with resistances (losses) in addition to curvilinear motion forces.

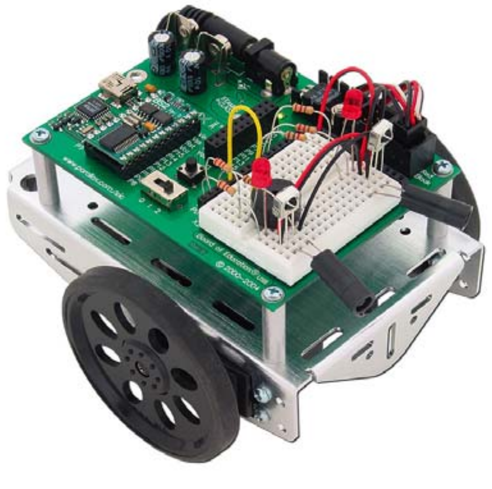


Fig. 1 Differentially steered mobile robot

We will approximate these forces in simplest manner to be proportional to a speed. Equations describing dependences of translation and rotation speed of selected chassis point to actual wheel motor voltages will be result of the dynamic part.

Selection of the point where actual translation and rotation speed will be evaluated influences significantly initial equations and hence complexity of the resulting model. If the selected point is centre of gravity then initial equations of dynamic part are simplest but equations describing dependencies between wheel speeds and translation and rotation speed are more complicated. Centre of the join between wheels is used as the selected chassis point in common literature. Such a choice leads to simplest recalculation of actual wheel speeds to motion equations of that point. Trade-off between these two approaches is chosen in our paper – point as centre of gravity projection to join between wheels is selected. Trajectory (time course) computation of another chassis points (points where wheels meet the ground) supplements dynamic part of the model.

2. MATHEMATICAL MODEL

Described mobile robot is driven by two DC motors with common voltage source and independent control of each motor. Motors are connected with driving wheels through gear-box with constant gear ratio. Ideal gear-box means that it reduces linearly angular speed and boosts the moment (nonlinearities are not considered). Loses in motor and also in gear-box are proportional to rotation speed. Chassis is equipped with caster wheel with no influence to chassis motion (its influence is included in resistance coefficients acting against motion).

Model of the robot consists from three relatively independent parts. Description of the ideal DC motors connected in series is given in chapter 2.1. Two equations describe dependency of the motor rotation speed and current on power supply voltage and loading moment related to chassis dynamics. Motion equations are presented in chapter 2.2 – dependency between translation and rotation speed of the selected point on moments acting to driving wheels. Chapter 2.3 is dedicated to equations describing how motor speed influences translation and rotation speed of the selected point and to complete model formulation. In last chapter 2.4 the model is transformed to simpler form which is more suitable for next using and for trajectory of arbitrary point calculation. Equations describing trajectory corresponding to contact points of the driving and caster wheels with the ground are formulated.

2.1 DC motor in series connection dynamics

Equivalent circuit of ideal DC motor in series connection (Poliak et al. 1987) is in Fig. 2. It consists from resistivity R, inductance L and magnetic field of the motor *M*. Commutator is not considered. Rotor produces electrical voltage with reverse polarity than source voltage – electromotive voltage U_M, which is proportional to rotor angular velocity ω. Twisting moment of the rotor M_M is proportional to current *i*.

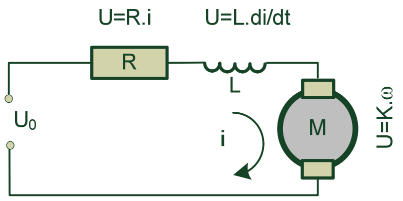


Fig. 2 Equivalent circuit of motor

Ideal behaviour means that whole electric energy used to magnetic field creating is transformed without any loses to mechanical energy – moment of the motor. We do not consider loses in magnetic field but only electric loses in winding and mechanical loses proportional to rotor speed.

Firs equation describes motor behaviour through balancing of voltages (Kirhoff's laws)

$$U_R + U_L = U_0 - U_M\,, \quad Ri + L\frac{di}{dt} = U_0 - K\omega \tag{2}$$

where

R	[Ω]	is motor winding resistivity,
L	[H]	is motor inductance,
K	[kg.m^2.s^{-2}.A^{-1}]	is electromotoric constant,
U_0	[V]	is source voltage,
ω	[rad.s^{-1}]	is rotor angular velocity and
i	[A]	is current flowing through winding.

Second equation is balance of moments (electric energy) – moment of inertia M_s, rotation resistance proportional to rotation speed (mechanical loses) M_o, load moment of the motor M_x and moment M_M caused by magnetic field which is proportional to current

$$\begin{gathered} M_s + M_o + M_x = M_M \\ J\frac{d\omega}{dt} + k_r.\omega + M_x = Ki \end{gathered} \tag{3}$$

where

J	[kg.m^2]	is moment of inertia,
k_r	[kg.m^2.s^{-1}]	is coefficient of rotation resistance and
M_x	[kg.m^2.s^{-2}]	is load moment.

2.2 Chassis dynamics

Chassis dynamics is defined with vector of translation speed v_B acting in selected chassis point and with rotation of this vector with angular velocity ω_B (constant for all chassis points). It is possible to calculate trajectory of arbitrary chassis point from these variables. Point B for which the equations are derived is centre of gravity normal projection to join between wheels – see Fig. 3. This leads according to authors to simplest set of equation for whole model. We consider general centre of gravity T position – usually it is placed to centre of the join between wheels.

We consider forces balances as starting equations. It is possible to replace two forces F_L and F_P acting to chassis in left (L) and right (P) wheel ground contact points with one force F_B and torsion moment M_B acting in point B. Chassis is characterized with semi-diameter of the driving wheels *r*, total weight *m*, moment of inertia J_T with respect to centre of gravity located with parameters l_T, l_L, l_P.

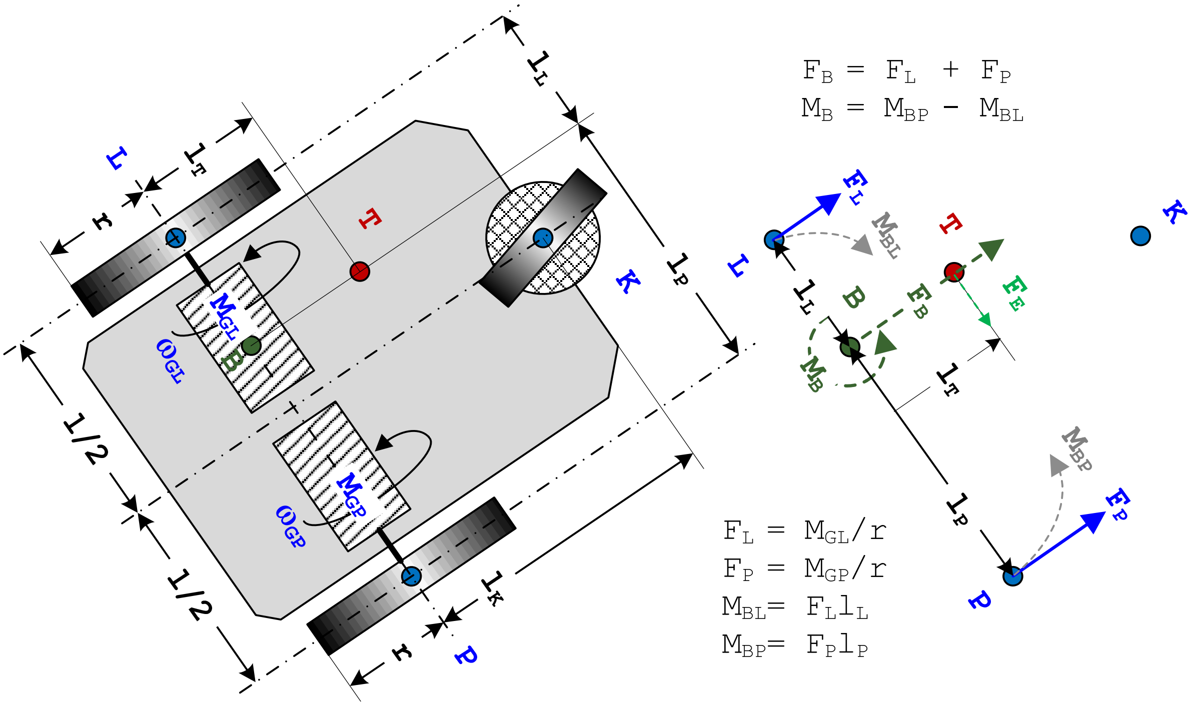


Fig. 3 Chassis scheme and forces

Let us specify equation (1) for our case. Position of the centre of gravity is constant with respect to axis of rotation so we do not need to consider Coriolis force. We have to consider Coriolis force for example if the chassis moves on rotating surface.

Similarly we do not consider centrifugal force – chassis is supposed to be solid body represented as mass point (centre of gravity). Because force vector causing the movement acts in point B and goes through centre of gravity it is enough if we consider inertial force by linear motion. By rotational motion it is necessary to consider moment caused with Euler's force because the axis of rotation does not go through the centre of gravity.

By the balance of forces causing linear motion we will consider except of forces F_L, F_P caused by drives and inertial force F_S also resistance force F_O proportional to speed v_B. The balance of forces influencing linear motion is

$$F_L + F_P + F_O + F_S = 0$$
$$\frac{M_{GL}}{r} + \frac{M_{GP}}{r} - k_v v_B - m\frac{dv_B}{dt} = 0 \qquad (4)$$

where

m [kg] is robot mass,
k_v [kg.s^{-1}] is resistance coefficient against linear motion
M_{GL} [kg.m^2.s^{-2}] is moment of the left drive,
M_{GP} [kg.m^2.s^{-2}] is moment of the right drive,
v_B [m.s^{-1}] is linear motion speed and
r [m] is semi-diameter of the wheels.

Balance of moments is slightly more complicated because the rotation axis does not lie in centre of gravity. That's why it is necessary to take into account except chassis momentum M_T also moment $M_E = l_T F_E$ caused by Euler's force F_E. Similarly as by linear motion we will consider moment M_O caused with resistance against rotation to be proportional to angular velocity ω_B.

$$M_{BL} + M_{BP} + M_O + M_T + M_E = 0$$
$$-\frac{M_{GL}}{r} l_L + \frac{M_{GP}}{r} l_P - k_\omega \omega_B - J_T \frac{d\omega_B}{dt} - l_T m \frac{d\omega_B}{dt} l_T = 0 \qquad (5)$$

where

l_P [m] is distance of the right wheel from point B,
l_L [m] is distance of the left wheel from point B,
l_T [m] is distance of the centre of gravity from point B,
k_ω [kg.m^2.s^{-1}] is resistance coefficient against rotary motion
J_T [kg.m^2] is moment of inertia with respect to rotation axis in centre of gravity and
ω_B [s^{-1}] is angular speed in point B.

Resulting moment of inertia J_B with respect to rotation axis in point B is given by eq. (6) which is parallel axis theorem or Huygens-Steiner theorem - see e.g. (Horák et al. 1976)

$$J_B = J_T + m l_T^2 \qquad (6)$$

where

J_T [kg.m^2] is moment of inertia with respect to centre of gravity and
l_T [m] is distance between centre of gravity and point B.

2.3 Relationship between rotation speed of the motor and centre of gravity chassis movement (kinematics)

The equation describing the behaviour of the two motors (currents and angular velocity) and the behaviour of the chassis (the speed of the linear movement and speed of the rotation) are connected only through moments of engines.

Equations express the law of conservation of energy which is conversion of electrical energy to mechanical including one type of losses but represent only one relationship between the speed of the two motors (peripheral speed of the drive wheels) and rates of movement and rotation of the chassis. Additional relation is given by design of the drive and chassis. We expect that both drive wheels are firmly linked to rotors of relevant engines over ideal gearbox with gear ratio p_G - without nonlinearities and any flexible members.

Gearbox decreases output angular velocity ω_{Gx} with relation to the input angular speed ω_x according to the transmission ratio p_G and simultaneously in the same proportion increases output torque M_{Gx} with relation to the input torque M_x.

$$\omega_{GL} = \frac{\omega_L}{p_G} \qquad \omega_{GP} = \frac{\omega_P}{p_G} \tag{7a}$$

$$M_{GL} = p_G M_L \qquad M_{GP} = p_G M_P \tag{7b}$$

Further we assume that both drive wheels have the same radius r and their peripheral speeds v_L, v_P depend on the angle speeds of gearbox output ω_{GL}, ω_{GP} according to relations

$$\begin{aligned} v_L &= r\omega_{GL} = r\frac{\omega_L}{p_G} \\ v_P &= r\omega_{GP} = r\frac{\omega_P}{p_G} \end{aligned} \tag{7c}$$

To determine the value of the linear speed in point B and the angular velocity of rotation let us start from Figure 4. We expect that both drive wheels have the same axis of rotation and therefore their peripheral speeds are always parallel. The illustration shows the positioning where the peripheral speeds v_L and v_P actually operate (driving wheels L and P) and the point B. We want to specify such a linear v_B and angular ω_B speeds that have the same effect as the action of the peripheral speed of the driving wheels. By using the similarity of triangles depicted in Figure 4 we can recalculate the peripheral speeds of the wheels v_L, v_P to the speed v_B in point B according to relation (8a) and the angular velocity of rotation ω_B according to the relation (8b)

$$v_B = \frac{v_L l_P + v_P l_L}{l_L + l_P} = \frac{r}{p_G(l_L + l_P)}(l_P\omega_L + l_L\omega_P) \tag{8a}$$

$$\omega_B = \frac{v_B}{x + l_L} = \frac{v_P - v_L}{l_L + l_P} = \frac{r}{p_G(l_L + l_P)}(-\omega_L + \omega_P) \tag{8b}$$

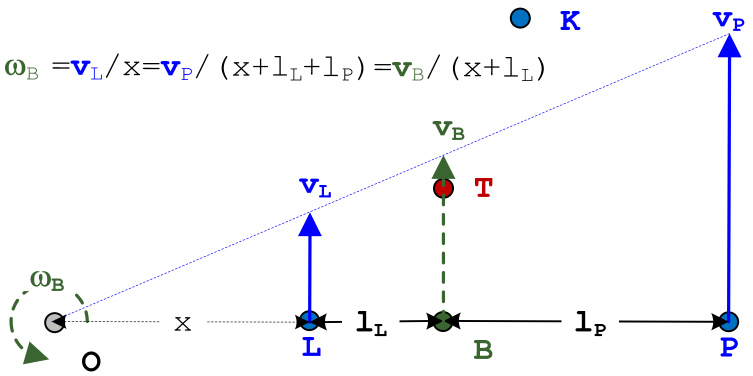


Fig. 4 Linear and angular speeds recalculations

2.4 Trajectory calculation of the arbitrary chassis points

We can determine from linear speed of v_B and angular speed ω_B (motion equations) current rotation angle α of the chassis and the current position (the coordinates x_B, y_B) of point B (Šrejtr 1954) according to relations

$$\frac{d\alpha}{dt} = \omega_B \tag{9a}$$

$$\frac{dx_B}{dt} = v_B \cos(\alpha) \tag{9b}$$

$$\frac{dy_B}{dt} = v_B \sin(\alpha) \tag{9c}$$

To determine the current position of the contacts of all three chassis wheels (points L, P and K) with ground we need to know the location of these points in relation to point B. This location is shown in Figure 5. From geometric dimensions we determine equation describing the relative position of these points in relation to point B depending on the angle of rotation.

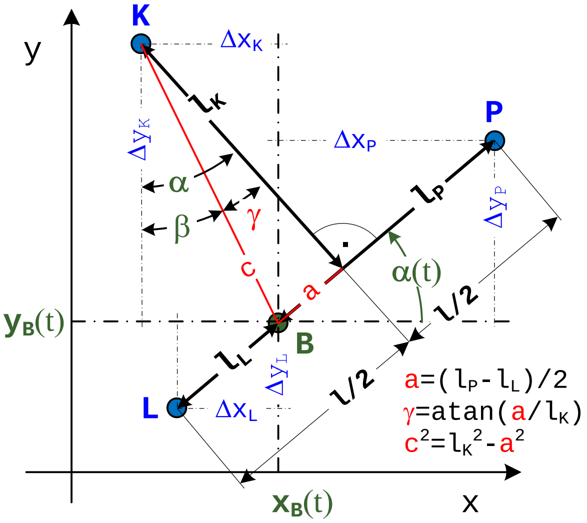


Fig. 5 Arbitrary chassis point recalculation

Relative positions Δx_L, Δy_L of the point L and Δx_P, Δy_P of the point P depending on angle of rotation α are given by

$$\Delta x_L = -l_L \sin(\alpha) \qquad \Delta y_L = -l_L \cos(\alpha) \tag{10a}$$

$$\Delta x_P = +l_P \sin(\alpha) \qquad \Delta y_P = +l_P \cos(\alpha) \tag{10b}$$

To determine the relative position Δx_K, Δy_K of the point K we use an auxiliary right triangle specified by cathetus c and hypotenuses a and l_K (see Figure 5). Then the equations for relative coordinates of the point K calculating are

$$\begin{aligned} &a = \tfrac{1}{2}(l_P - l_L) \\ &\gamma = \arctan\left(\frac{a}{l_K}\right) \qquad c = \sqrt{a^2 + l_K^2} \\ &\Delta x_K = -c\sin(\alpha - \gamma) \qquad \Delta y_K = c\cos(\alpha - \gamma) \end{aligned} \tag{10c}$$

2.5 Overall model and steady-state

The dynamic part of the model consists from four differential equations describing the behaviour of both motors, two differential equations describing the dynamics of the chassis and two algebraic equations with dependency of linear and angular chassis speed on the peripheral speeds of the driving wheels. We can find in these equations eight state variables describing the current state of the left motor (current i_L, angular velocity of the rotor ω_L, loading moment M_L) and the right motor (current i_P, angular velocity of the rotor ω_P, loading moment M_P) and the movement of the chassis (linear speed v_B and angular velocity of rotation ω_B). All the state variables are dependent on the time courses of the power of the left U_L and right U_P motor.

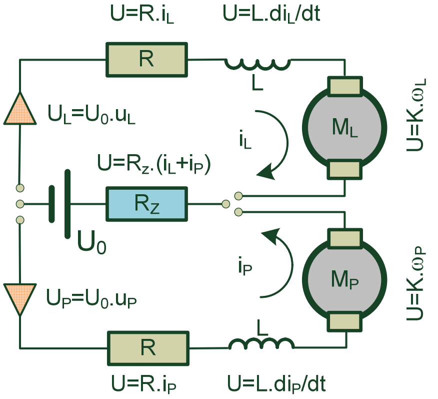


Fig. 6 Motors wiring

Each motor has its own power supply voltage (U_L, U_P) disbranched from the common source of voltage U_0. Control of the supply voltage of both motors using amplifier with control signal u_x is shown in Figure 6. Because both engines are powered from the common source it will be taken into account also effect of the internal resistance R_z. Both motors are considered with the same parameters. We can write with using the equations (2) and (3) and Figure 6 four differential equations describing the behaviour of both engines as

$$Ri_L + R_z(i_L + i_P) + L\frac{di_L}{dt} = u_L U_0 - K\omega_L \tag{11a}$$

$$Ri_P + R_z(i_L + i_P) + L\frac{di_P}{dt} = u_P U_0 - K\omega_P \tag{11b}$$

$$J\frac{d\omega_L}{dt} + k_r\omega_L + M_L = Ki_L \tag{12a}$$

$$J\frac{d\omega_P}{dt} + k_r\omega_P + M_P = Ki_P \tag{12b}$$

Differential equations (4) and (5) describing the behaviour of the chassis complete the dynamic model. We can rewrite these equations with respect to the equations (7) and introduction of the "reduced" radius of the wheel r_G and total moment of inertia J_B (13a) as

$$r_G = \frac{r}{p_G} \qquad J_B = J_T + ml_T^2 \tag{13a}$$

$$\begin{gathered}\frac{p_G}{r}M_L + \frac{p_G}{r}M_P - k_v v_B - m\frac{dv_B}{dt} = 0 \\ M_L + M_P - r_G k_v v_B - r_G m\frac{dv_B}{dt} = 0\end{gathered} \tag{13b}$$

$$\begin{gathered}-l_L\frac{p_G}{r}M_L + l_P\frac{p_G}{r}M_P - k_\omega\omega_B - \left(J_T + ml_T^2\right)\frac{d\omega_B}{dt} = 0 \\ -l_L M_L + l_P M_P - k_\omega\omega_B - r_G J_B\frac{d\omega_B}{dt} = 0\end{gathered} \tag{13c}$$

It is possible to rewrite the last two algebraic equations (8a) a (8b) describing the dependence between rotations speed of both motors and chassis movement with using the substitution (13a) as

$$v_B = \frac{r_G}{l_L + l_P}\left(l_P\omega_L + l_L\omega_P\right) \tag{14a}$$

$$\omega_B = \frac{r_G}{l_L + l_P}\left(-\omega_L + \omega_P\right) \tag{14b}$$

These six differential equations (11a,b), (12a,b), (13b,c) and two algebraic equations (14a,b) containing eight state variables representing a mathematical description of dynamic behaviour of ideal differentially steered mobile robot with losses linearly dependent on the revolutions or speed. Control signals u_L and u_P that control the supply voltages of the motors are input variables and the speed of the movement v_B and speed of rotation ω_B are output variables. From them with using the equations (9a) – (9c) we can determine the current coordinates of a point B and the angle of rotation of the chassis.

In the following calculation of steady-state values for constant engine power voltages is given. Calculation of steady-state is useful both for the checking of derived equations and secondly for the experimental determination of the values of the unknown parameters. Because the equation (11)-(14) are linear with respect to state variables the calculation of steady-state leads to a system of eight linear equations which we can write in matrix form as

$$\begin{bmatrix} R+R_z & R_z & K & 0 & 0 & 0 & 0 & 0 \\ R_z & R+R_z & 0 & K & 0 & 0 & 0 & 0 \\ K & 0 & -k_r & 0 & -1 & 0 & 0 & 0 \\ 0 & K & 0 & -k_r & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -r_G k_v & 0 \\ 0 & 0 & 0 & 0 & -l_L & l_P & 0 & -r_G k_\omega \\ 0 & 0 & l_P & l_L & 0 & 0 & -\frac{l_P + l_L}{r_G} & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & -\frac{l_P + l_L}{r_G} \end{bmatrix}\begin{bmatrix} i_L \\ i_P \\ \omega_L \\ \omega_P \\ M_L \\ M_P \\ v_B \\ \omega_B \end{bmatrix} = \begin{bmatrix} U_L \\ U_P \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

2.6 Computational form of the model

A mathematical model will be used in particular for the design and simulation validation of control movement of the robot. Model can be divided into three series-involved parts as shown in Figure 7. From the control point of view action variables are signals u_L a u_P that control the supply voltage of the motors. Momentary speed v_B and speed of rotation ω_B are output variables from linear part of the model. These variables are the inputs to the consequential non-linear part of the model, whose outputs are controlled variables - the coordinates of selected point position x_B, y_B and the rotation angle of the chassis α. The last part is the calculation of coordinates of the position of arbitrary points of the chassis.

We can modify linear part of the model into simpler form for control design purposes – to reduce number of differential equations from six to four. If we substitute equations (14a,b) into (13b,c) and eliminate moments M_L a M_P by substitution of (12a,b) to (13b,c) we are able to reduce four differential equations (12a,b) a (13b,c) into two (17c,d).

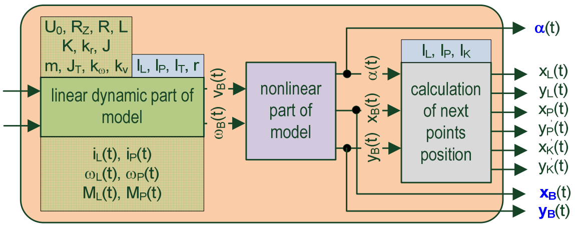


Fig. 7 Model partitioning into linear and nonlinear part

If we introduce substitution of the parameters according to following formulas

$$a_L = k_r + \frac{k_v l_P r_G^2}{l_L + l_P} \qquad a_P = k_r + \frac{k_v l_L r_G^2}{l_L + l_P} \tag{16a}$$

$$b_L = J + \frac{m l_P r_G^2}{l_L + l_P} \qquad b_P = J + \frac{m l_L r_G^2}{l_L + l_P} \tag{16b}$$

$$c_L = k_r l_L + \frac{k_\omega r_G^2}{l_L + l_P} \qquad c_P = k_r l_P + \frac{k_\omega r_G^2}{l_L + l_P} \tag{16c}$$

$$d_L = J l_L + \frac{J_B r_G^2}{l_L + l_P} \qquad d_P = J l_P + \frac{J_B r_G^2}{l_L + l_P} \tag{16d}$$

The reduced linear part of the model consists from set of equations

$$\frac{di_L}{dt} = \frac{u_L U_0 - K.\omega_L - (R + R_z).i_L - R_z i_P}{L} \tag{17a}$$

$$\frac{di_P}{dt} = \frac{u_P U_0 - K.\omega_P - (R + R_z).i_P - R_z i_L}{L} \tag{17b}$$

$$\begin{gathered}\frac{d\omega_L}{dt} = \frac{1}{b_L d_P + b_P d_L}\big(d_P\big[K(i_L + i_P) - a_L\omega_L - a_P\omega_P\big] - \\ - b_P\big[K(-l_L i_L + l_P i_P) + c_L\omega_L - c_P\omega_P\big]\big)\end{gathered} \tag{17c}$$

$$\begin{gathered}\frac{d\omega_P}{dt} = \frac{1}{b_L d_P + b_P d_L}\big(d_L\big[K(i_L + i_P) - a_L\omega_L - a_P\omega_P\big] + \\ + b_L\big[K(-l_L i_L + l_P i_P) + c_L\omega_L - c_P\omega_P\big]\big)\end{gathered} \tag{17d}$$

and output variables are given by algebraic equations (14a,b).

It is possible to write reduced linear part of the model as standard state-space model in matrix form as

$$\begin{gathered}\frac{d\mathbf{x}}{dt} = \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} = \mathbf{Cx}\end{gathered} \qquad \mathbf{x} = \begin{bmatrix} i_L \\ i_P \\ \omega_L \\ \omega_P \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} u_L \\ u_P \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} v_B \\ \omega_B \end{bmatrix} \tag{18a}$$

with constant matrices **A**, **B** a **C**

$$\mathbf{A} = \begin{bmatrix} -\frac{R+R_z}{L} & -\frac{R_z}{L} & -\frac{K}{L} & 0 \\ -\frac{R_z}{L} & -\frac{R+R_z}{L} & 0 & -\frac{K}{L} \\ \frac{K(d_P + b_P l_L)}{b_L d_P + b_P d_L} & \frac{K(d_P - b_P l_P)}{b_L d_P + b_P d_L} & -\frac{d_P a_L + b_P c_L}{b_L d_P + b_P d_L} & -\frac{d_P a_P - b_P c_P}{b_L d_P + b_P d_L} \\ \frac{K(d_L - b_L l_L)}{b_L d_P + b_P d_L} & \frac{K(d_L + b_L l_P)}{b_L d_P + b_P d_L} & -\frac{d_L a_L - b_L c_L}{b_L d_P + b_P d_L} & -\frac{d_L a_P + b_L c_P}{b_L d_P + b_P d_L} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} \frac{U_0}{L} & 0 \\ 0 & \frac{U_0}{L} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & \frac{l_P r_G}{l_L + l_P} & \frac{l_L r_G}{l_L + l_P} \\ 0 & 0 & -\frac{r_G}{l_L + l_P} & \frac{r_G}{l_L + l_P} \end{bmatrix}$$

3. EXAMPLE OF THE BEHAVIOUR

Basic verification of the above derived model was made by calculation for situations where we can guess the behaviour of the real device. First value of the state variables in steady states will be given for some combinations of parameters and motor supply voltages. Further time courses of the robot trajectory will be determined for some combinations of time courses of supply voltages when robot is starting from zero speed.

Values of the parameters listed in the following tables are used in all of the calculations. These values are chosen so that they at least roughly correspond to the values estimated for the robot in Figure 1. The values of the geometrical and other parameters of the chassis are listed in Table 1.

Table 1 Chassis parameters

Notation	Value	Dimension	Meaning
l_L	0.040	m	distance of the left wheel from point B
l_P	0.060	m	distance of the right wheel from point B
l_T	0.020	m	distance of centre of gravity from join between wheels
l_K	0.040	m	distance of caster wheel from join between wheels
r	0.050	m	semi-diameter of driving wheel
m	1.250	kg	total weight of the robot
k_v	0.100	kg.s^{-1}	coefficient of the resistance against robot linear motion
J_T	0.550	kg.m^2	moment of inertia of robot with respect to centre of gravity
k_ω	1.350	kg.m^2.s^{-1}	coefficient of the resistance against robot rotating

Necessary parameters for DC motors with common voltage source description are given in Table 2. We consider identical motors with identical parameters.

Table 2 DC motors parameters

Notation	Value	Dimension	Meaning
R	2.000	Ω	motor winding resistivity
L	0.050	H	motor inductance
K	0.100	kg.m^2.s^{-2}.A^{-1}	electromotoric constant
R_Z	0.200	Ω	source resistance
U_0	10.00	V	source voltage
J	0.025	kg.m^2	total moment of inertia of rotor and gearbox
k_r	0.002	kg.m^2.s^{-1}	coefficient of the resistance against rotating of rotor and gearbox
p_G	25	---	gearbox transmission ratio

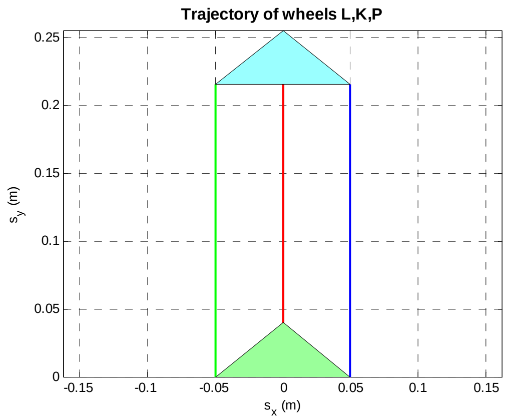
3.1 Steady state for different positions of point B and motors voltages

The steady states are calculated as a solution of the system of eight equations in matrix form (15). Traces of the wheels are shown during the first 20 seconds of motion from zero initial conditions - calculated from state-space model (18) and from the equations for the trajectories calculation (9,10). Trajectories are plotted for the situation that the origin of the coordinate system is in the centre between the wheels, which is on the x-axis and the default orientation of the robot is in the direction of the y axis. Starting and final position of the robot is displayed using the triangle that connects all three wheels. Trajectory of the centre of gravity is displayed in addition to the traces of the wheels.

Steady-state A (Table 3a) corresponds to the geometric arrangement - point (B) is midway between the wheels and both engines have the same supply voltage. The result is that the robot moves only linearly.

Table 3a Steady state A

	left wheel	right wheel	
U	1.000	1.000	V
l	0.050	0.050	m
i	0.13514	0.13514	A
ω	1.07534	1.07534	Hz
M	0.000001	0.000001	N.m
v_B	0.0013513		m.s^{-1}
ω_B	0.0000000		Hz



The following three experiments show the influence of centre of gravity position. Steady-state B (table 3b) holds again for the symmetric geometric arrangement but only one motor is powered. Steady state C (table 3c) shows the situation in the case that point B is in the extreme position above the left wheel and only the left motor is powered. Steady-state D (table 3d) corresponds to the same position of the point B above the left wheel but is only right motor is powered. In all three cases the robot rotates and at the same time the point B has some linear speed. Both wheels produce translation because of the interactions.

Table 3b Steady state B

	left wheel	right wheel	
U	0.000	1.000	V
l	0.050	0.050	m
i	-0.02772	0.16287	A
ω	0.04523	1.07935	Hz
M	-0.001176	1.03010	N.m
v_B	0.006757		m.s^{-1}
ω_B	0.123762		Hz

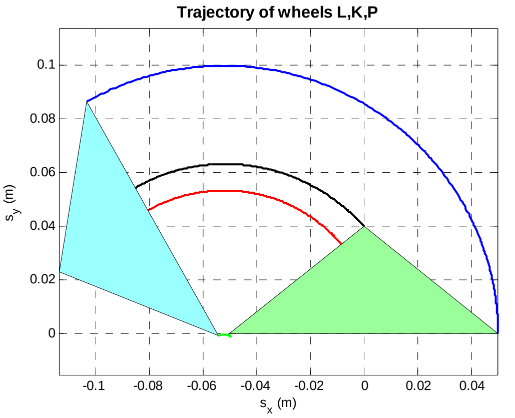


Table 3c Steady state C

	left wheel	right wheel	
U	1.000	0.000	V
l	0.000	0.100	m
i	0.16288	-0.02773	A
ω	1.030062	0.045243	Hz
M	0.00334400	-0.00334141	N.m
v_B	0.0129441		m.s^{-1}
ω_B	-0.1237559		Hz

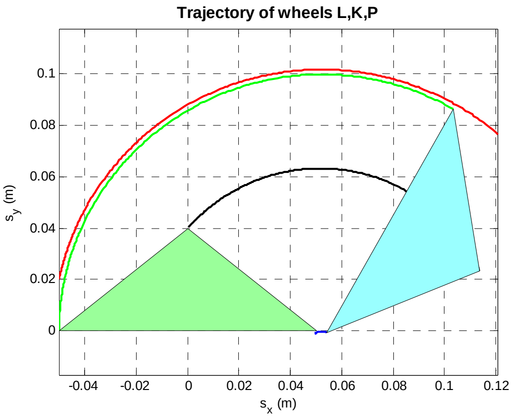
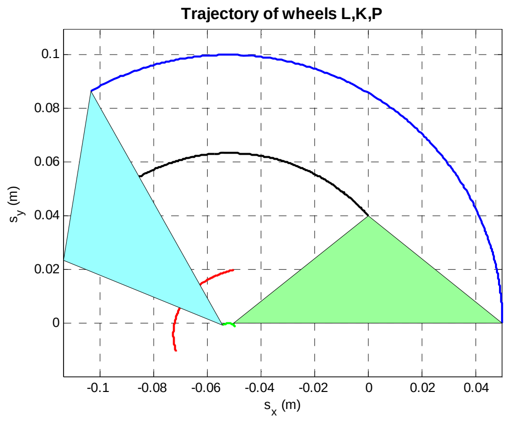


Table 3d Steady state D

	left wheel	right wheel	
U	0.000	1.000	V
l	0.000	0.100	m
i	-0.02773	0.16286	A
ω	0.045248	1.030120	Hz
M	-0.00334148	0.00334159	N.m
v_B	0.0005686		$\mathrm{m.s^{-1}}$
ω_B	0.1237626		Hz



3.2 Dynamic behaviour for particular cases

Dynamic behaviour is demonstrated on the time courses of currents and angular speeds of the motors starting from zero initial conditions. Graphs in Figure 8 show courses of supply voltages, currents and angular speeds for the case that the point B is in the middle between both motors with the same constant voltage 1 V. Situation corresponds to experiment with the parameters in Table 3a.

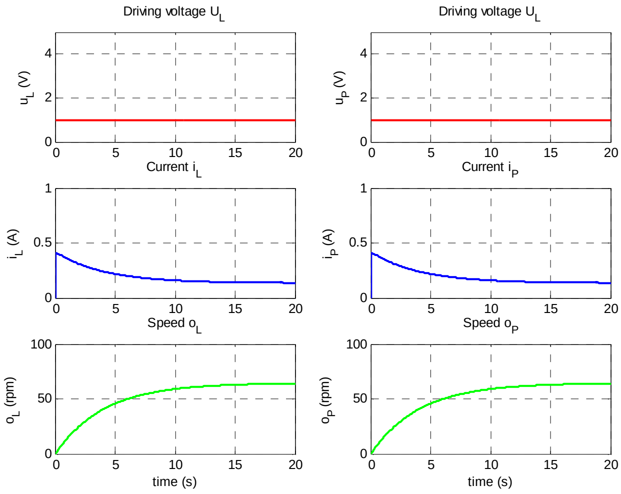


Fig. 8 Dynamic behaviour - constant supply voltage 1 V for both motors

Situation where point B is in the middle between both motors with the right motor voltage 1 V only corresponds to experiment with the parameters in Table 3b.

Illustrative example of behaviour in the situation when both voltages are periodic and with different amplitudes is in figures 10 and 11. On the left motor is a rectangular voltage of period 20 s, phase offset 10 s and amplitude 3. On the right motor is a rectangular voltage of doubled period 40 s and amplitude 4.

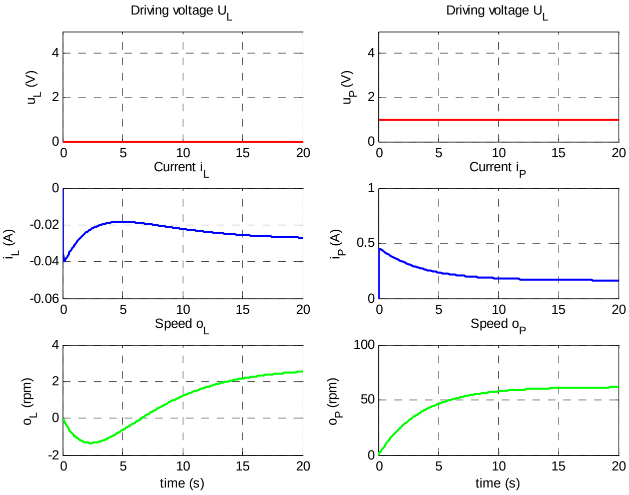


Fig. 9 Dynamic behaviour - constant supply voltage 1 V for right motor

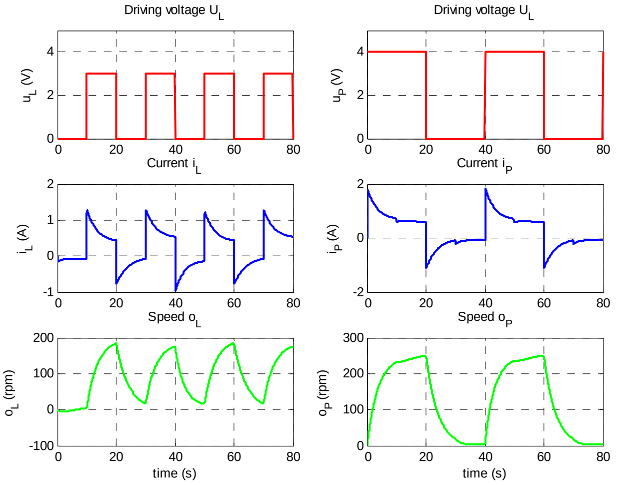


Fig. 10 Dynamic behaviour - periodic voltages

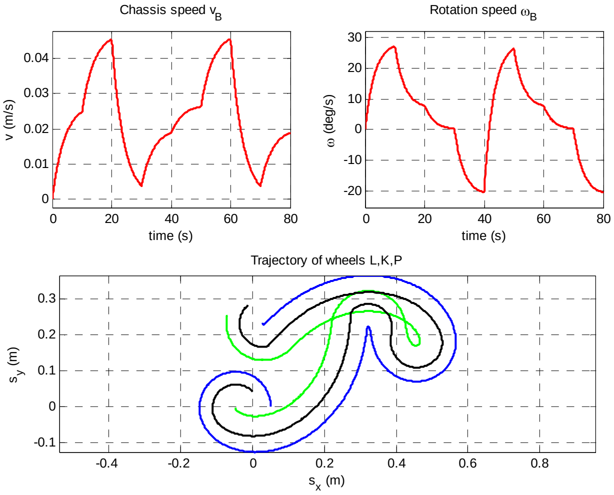


Fig. 11 Dynamic behaviour - periodic voltages – speeds and trajectories

4. CONCLUSION

The behaviour of the dynamic model in simulated situations agrees with the expected behaviour of a real device. Position of centre of gravity does not affect the behaviour in steady state. Immediate linear speed in point B depends on its position but the trajectories of the wheels are independent on the position of the point B.

Interaction of the two drives was confirmed. Because of the forces of inertia and the forces of resistance also wheel without supply voltage rotates by the chassis movement. Even change of the meaning of the rotation occurs in the transient state. This situation is seen in Figure 9.

Motor dynamics is negligible compared to the expected dynamics of the chassis for estimated motor parameters. Because the parameters of the model have physical meaning it will be possible to measure directly some parameters on real device. Identification of additional parameters will be possible experimentally from measured time courses of power voltages and the corresponding courses of angular speed of the wheels.

ACKNOWLEDGMENTS

The work has been supported in the framework of the research project MSM 0021627505 in part "Management, optimization and diagnostics of complex systems". This support is very gratefully acknowledged.

REFERENCES

Horák, Z., Krupka, F. (1976). *Fyzika*, vol. 1. SNTL Praha, 1976, 422 p.

Lucas, G.W. (2010). *A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators*, http://rossum.sourceforge.net/papers/DiffSteer/DiffSteer.html [cited 11.11.2010]

Novák, P. (2005). *Mobilní roboty (pohony, senzory, řízení).* BEN 2005, ISBN 80-7300-141-1

Poliak, F., Fedák, V., Zboray, L. (1987). *Elektrické pohony*, Bratislava: Alfa, 1987

Stengel, R.F. (2010). *Robotics and Intelligent Systems;A Virtual textbook*, http://www.princeton.edu/~stengel/RISVirText.html [cited 11.11.2010]

Šrejtr, J. (1954). *Technická mechanika II. Kinematika 1. část.* SNTL Praha, 1954, 256 p.

Winkler, Z. (2010). *Odometrie*; http://robotika.cz/guide/odometry/cs [cited 11.11.2010]

Identification of nonlinear systems with general output backlash

J. Vörös*

** Slovak Technical University, Faculty of Electrical Engineering and Information Technology,
Institute of Control and Industrial Informatics, Ilkovicova 3, 812 19 Bratislava, Slovakia
e-mail : jvoros@elf.stuba.sk*

Abstract: The paper deals with identification of nonlinear cascade systems with general output backlash, where instead of the straight lines determining the upward and downward parts of backlash characteristic general curves are considered. A new form of general backlash description is leading to the mathematical model, which has all the model parameters separated. The identification based on this model is solved as a quasi-linear problem using an iterative algorithm with internal variables estimation.

1. INTRODUCTION

One of the most important nonlinearities that limit control systems performance in many applications is the so-called backlash (Kalas et al, 1985). Unfortunately, there are only few contributions in the literature on the identification of systems with backlash (Bai, 2002), (Cerone and Regruto, 2007), (Dong et al, 2009), (Dong et al, 2010), (Giri et al, 2008), (Hägglund, 2007), (Vörös, 2010a), (Vörös, 2010b).

In control systems it is assumed that the backlash is “linear”, i.e., straight lines approximate the upward and downward curves of the characteristic (Tao and Kokotovic, 1993), (Tao and Canudas de Wit, 1997), (Nordin and Gutman, 2002). This simplifies the system description, however, in some cases it may lead to inaccuracies. The components of control systems may be free from backlash when new, but after some time in use the wear results in an introduction of backlash in the systems. In general the form of backlash changes with time and wear, regardless of what form of backlash was present when the component was new. Therefore it may be appropriate to generalize the backlash and consider general upward and downward curves. The only works dealing with the identification of systems with general switch and backlash nonlinearities were published in (Giri et al, 2010), (Rochdi et al, 2010a), (Rochdi et al, 2010b). The proposed approach is based on two independent, but structurally symmetric identification schemes. The first one determines the points located on the descendent border of general nonlinearity as well as the parameters of the linear subsystem. The second identification scheme determines the points located on the ascendent border of general nonlinearity and the parameters of the linear subsystem. The key idea is to use pulse-type periodic input signals so that only the points of interest are excited on each border.

In this paper an identification method for cascade systems with output backlash based on a new mathematical model for general backlash is presented. First, an analytic description of this hard dynamic nonlinearity is described, which uses appropriate switching functions and their complements (Vörös, 2009). Then the identification method for cascade systems consisting of a linear dynamic system followed by a general output backlash is proposed. This is based on a mathematical model, where the parameters of linear dynamic system and the parameters characterizing the general backlash are separated, hence their estimation can be solved as a quasi-linear problem using an iterative method with internal variable estimation (Vörös, 2001, 2003, 2007).

2. GENERAL BACKLASH MODEL

In the case of “linear” backlash the left and right branches of the characteristic are considered to be straight lines. However, in some applications the straight lines are only advantageous approximations of general curves constituting the left and right branches of backlash as shown in Fig. 1.

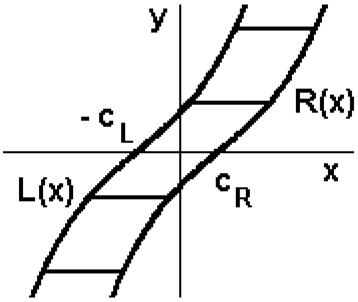


Fig. 1 General backlash characteristic

The general backlash characteristic can be described by the equation (Vörös, 2009)

$$y(t) = \begin{cases} L[x(t)] & x(t) \le z_L \\ y(t-1) & z_L \le x(t) \le z_R \\ R[x(t)] & x(t) \ge z_R \end{cases} \qquad (1)$$

where the mappings *L[x(t)]* and *R[x(t)]* describe the left and right branches of the characteristic, respectively, the x-axis values z_L and z_R are given as follows:

$$y(t-1) = L(z_L) \tag{2}$$

$$y(t-1) = R(z_R) \tag{3}$$

Assume the left and right curves can be approximated by the polynomials

$$L[x(t)] = \sum_{i=1}^{n} m_{Li}[x(t)+c_L]^i \tag{4}$$

$$R[x(t)] = \sum_{i=1}^{n} m_{Ri}[x(t)-c_R]^i \tag{5}$$

respectively, where $c_L > 0$, $c_R > 0$ are the intersections of *L[x(t)]* and *R[x(t)]* with the x-axis. Then the general backlash characteristic can be written as

$$y(t) = \begin{cases} \sum_{i=1}^{n} m_{Li}[x(t)+c_L]^i & x(t) \le z_L \\ y(t-1) & z_L \le x(t) \le z_R \\ \sum_{i=1}^{n} m_{Ri}[x(t)-c_R]^i & x(t) \ge z_R \end{cases} \tag{6}$$

where

$$y(t-1) = \sum_{i=1}^{n} m_{Li}[z_L + c_L]^i \tag{7}$$

$$y(t-1) = \sum_{i=1}^{n} m_{Ri}[z_R - c_R]^i \tag{8}$$

After introducing the internal variables

$$\xi_1(t) = x(t) + c_L \tag{9}$$

$$\xi_2(t) = x(t) - c_R \tag{10}$$

the following variables based on (7) and (8) can be defined:

$$f_1(t) = h[\xi_1(t)] = h[\sum_{i=1}^{n} m_{Li}\xi_1^i(t) - y(t-1)] \tag{11}$$

$$f_2(t) = h[\xi_2(t)] = h[y(t-1) - \sum_{i=1}^{n} m_{Ri}\xi_2^i(t)] \tag{12}$$

where the switching function

$$h(s) = \begin{cases} 0, & if\ s > 0 \\ 1, & if\ s \le 0 \end{cases} \tag{13}$$

is switching between two sets of values, i.e., (– ∞, s) and (s, ∞). Then the general backlash can be modeled by one difference equation as follows:

$$\begin{aligned} y(t) = \sum_{i=1}^{n} m_{Li}\xi_1^i(t) f_1(t) + \sum_{i=1}^{n} m_{Ri}\xi_2^i(t) f_2(t) + \\ y(t-1)[1-f_1(t)][1-f_2(t)] \end{aligned} \tag{14}$$

To include the deadzone parameters c_L and c_R into the backlash model, we can separate the first terms of the sums in (14) and half-substitute from (9) and (10) as follows:

$$\begin{aligned} & y(t) = m_{L1}x(t) f_1(t) + m_{L1}c_L f_1(t) + \\ & \sum_{i=2}^{n} m_{Li}\xi_1^i(t) f_1(t) + m_{R1}x(t) f_2(t) + m_{R1}c_R f_2(t) + \\ & \sum_{i=2}^{n} m_{Ri}\xi_2^i(t) f_2(t) + y(t-1)[1-f_1(t)][1-f_2(t)]. \end{aligned} \tag{15}$$

Now the input/output relation for the generalized backlash (15) is identical with that of (1). All the model parameters are separated and the model is linear in the input, output and internal variables. This model allows the upward and downward curves to be different provided that the intersection of the two curves is not in the region of practical interest.

3. SYSTEMS WITH GENERAL OUTPUT BACKLASH

In many real control systems the backlash appears in a cascade connection with linear dynamic systems. One of the simplest cases is the cascade system where a linear dynamic system is followed by a backlash as shown in Fig. 2. The linear dynamic system can be described by the difference equation as

$$x(t) = \sum_{i=1}^{na} a_i u(t-i) - \sum_{j=1}^{nb} b_j x(t-j) \tag{16}$$

where *u(t)* and *x(t)* are the inputs and outputs, respectively.

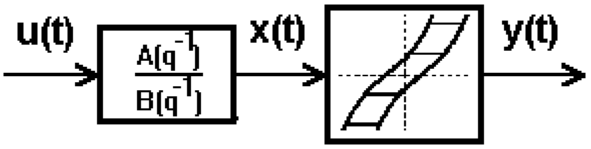


Fig. 2 Cascade system with general output backlash

Let the general backlash be described by (15). The output equation of this cascade system can be constructed by connecting (15) and (16). However, a direct substitution of (16) into (15) would lead to a quite complex expression, therefore the so-called key term separation principle can be applied (Vörös, 2010c). It means that (16) will be substituted only for *x(t)* in the first term of (15). Moreover, in this connection of two systems we can assume that $m_{L1} = 1$, hence the model equation for the cascade system with general output backlash can be written as

$$\begin{aligned} y(t) &= \sum_{i=1}^{na} a_i u(t-i)\, f_1(t) - \sum_{j=1}^{nb} b_j x(t-j)\, f_1(t) + \\ & c_L\, f_1(t) + \sum_{i=2}^{n} m_{Li} \xi_1^i(t)\, f_1(t) + m_{R1} x(t)\, f_2(t) + \\ & m_{R1} c_R\, f_2(t) + \sum_{i=2}^{n} m_{Ri} \xi_2^i(t)\, f_2(t) + \\ & y(t-1)[1-f_1(t)][1-f_2(t)] \end{aligned} \qquad (17)$$

where the parameters of both the linear system and the general backlash are separated and the equation is quasi-linear as the variables *ξ₁(t), ξ₂(t), f₁(t)* and *f₂(t)* depend on the backlash parameters and the internal variable *x(t)* depends on the linear system parameters. Defining the vector of data

$$\begin{aligned} \varphi(t) &= [u(t-1)\, f_1(t), \cdots, u(t-na)\, f_1(t), \\ & -x(t-1)\, f_1(t), \cdots, -x(t-nb)\, f_1(t), \\ & f_1(t), \xi_1(t)^2\, f_1(t), \cdots, \xi_1(t)^n\, f_1(t),\, x(t)\, f_2(t), \\ & -f_2(t), \xi_2(t)^2\, f_2(t), \cdots, \xi_2(t)^n\, f_2(t)]^T \end{aligned} \qquad (18)$$

and the vector of parameters

$$\begin{aligned} \theta &= [a_1, \cdots, a_{na}, b_1, \cdots, b_{nb}\, c_1, m_{L2}, \cdots, m_{Ln}, m_{R1}, \\ & c_2, m_{R2}, \cdots, m_{Rn}]^T \end{aligned} \qquad (19)$$

where

$$m_{L1} = 1, \quad c_L = c_1, \quad c_R = c_2 / m_{R1}, \qquad (20)$$

the cascade system with general output backlash can be written in the vector form as follows:

$$y(t) - y(t-1)[1-f_1(t)][1-f_2(t)] = \varphi^T(t)\,\theta\,. \qquad (21)$$

As the variables *ξ₁(t), ξ₂(t), f₁(t), f₂(t)* and the internal variable *x(t)* in (18) are unmeasurable and must be estimated, an iterative parameter estimation process has to be considered similarly as in (Vörös, 2007). Assigning the internal variable *x(t)* in the s-th step as

$$^{s}x(t) = \sum_{i=1}^{na} {}^{s}a_i u(t-i) - \sum_{j=1}^{nb} {}^{s}b_j\, {}^{s}x(t-j) \qquad (22)$$

and the estimated variables *ξ₁(t), ξ₂(t), f₁(t)* and *f₂(t)* in the s-th step as

$$^{s}\xi_1(t) = {}^{s}x(t) + {}^{s}c_L \qquad (23)$$

$$^{s}\xi_2(t) = {}^{s}x(t) - {}^{s}c_R \qquad (24)$$

$$^{s}f_1(t) = h[\sum_{i=1}^{n} {}^{s}m_{Li}\, {}^{s}\xi_1^i(t) - y(t-1)] \qquad (25)$$

$$^{s}f_2(t) = h[y(t-1) - \sum_{i=1}^{n} {}^{s}m_{Ri}\, {}^{s}\xi_2^i(t)] \qquad (26)$$

the error to be minimized in the estimation procedure is

$$\begin{aligned} ^{s+1}e(t) &= y(t) - y(t-1)[1-{}^{s}f_1(t)][1-{}^{s}f_2(t)] - \\ & {}^{s}\varphi^T(t)\ {}^{s+1}\theta \end{aligned} \qquad (27)$$

where $^{s}\varphi(t)$ is the data vector with the corresponding estimates of variables *x(t), ξ₁(t), ξ₂(t), f₁(t)* and *f₂(t)* according to (22) – (26) and $^{s+1}\theta$ is the (s+1)-th estimate of the parameter vector.

The steps in the iterative procedure may be now stated as follows:

a) Minimizing the least squares criterion based on (27)

$$^{s+1}J = \frac{1}{N} \sum_{t=1}^{N} {}^{s+1}e^2(t) \qquad (28)$$

where N is the number of measured input and output samples, the estimates of parameters $^{s+1}\theta$ are computed using $^{s}\varphi(t)$ with the s-th estimates of variables $^{s}x(t)$, $^{s}\xi_1(t)$, $^{s}\xi_2(t)$, $^{s}f_1(t)$ and $^{s}f_2(t)$.

b) Using (22) – (26) the estimates of $^{s+1}\varphi(t)$ are evaluated by means of the recent estimates of corresponding parameters.

c) If the estimation criterion is met the procedure ends, else it continues by repeating steps a) and b).

In the first iteration only the parameters of linear dynamic system are estimated and the initial values can be chosen zero. However, nonzero initial values of the general backlash parameters m_{R1}, c_L and c_R have to be considered for evaluation of $^{1}\varphi(t)$ to start up the iterative algorithm.

4. CONCLUSIONS

In this paper a new analytic form of general backlash characteristic description was used in the mathematical model for cascade systems including this type of dynamic nonlinearity in the output. Iterative algorithm was proposed enabling simultaneous estimation of both the backlash parameters and the parameters of the cascaded linear dynamic system on the basis of input/output data.

ACKNOWLEDGMENTS

I would like to thank the reviewers for valuable comments. The work has been supported by the Slovak Scientific Grant Agency (VEGA). This support is very gratefully acknowledged.

REFERENCES

Bai, E.W. (2002). Identific Bai, E.W. (2002). Identification of linear systems with hard input nonlinearities of known structure. *Automatica* **38,** 853-860.

Cerone, V. and D. Regruto (2007). Bounding the parameters of linear systems with input backlash. *IEEE Trans. Automatic Control* **52,** 531-536.

Dong, R., Y. Tan and H. Chen (2010). Recursive identification for dynamic systems with backlash. *Asian Journal of Control* **12,** 26-38.

Dong, R., Q. Tan and Y. Tan (2009). Recursive identification algorithm for dynamic systems with output backlash and its convergence. *Int. J. Appl. Math. Comput. Sci.* **19,** 631-638.

Giri, F., Y. Rochdi, F.Z. Chaoui and A. Brouri (2008). Identification of Hammerstein systems in presence of hysteresis-backlash and hysteresis-relay nonlinearities. *Automatica* **44,** 767-775.

Giri, F., Y. Rochdi, J.B. Gning and F.Z. Chaoui (2010). Hammerstein systems identification in presence of nonparametric backlash nonlinearities. In: Proc. American Control Conference, Baltimore, MD, USA, 4516-4521.

Hägglund, T. (2007). Automatic on-line estimation of backlash in control loops. *Journal of Process Control* **17,** 489-499.

Kalaš, V., L. Jurišica, M. Žalman, S. Almássy, P. Siviček, A. Varga and D. Kalaš (1985). *Nonlinear and Numerical Servosystems*. Bratislava, Slovakia: Alfa/SNTL (in Slovak).

Nordin, M., & Gutman, P.O. (2002). Controlling mechanical systems with backlash - a survey. *Automatica* **38,** 1633-1649.

Rochdi, Y., F. Giri, J.B. Gning and F.Z. Chaoui (2010a). Identification of block-oriented systems in the presence of nonparametric input nonlinearities of switch and backlash types. *Automatica* **46,** 785-958.

Rochdi, Y., F. Giri, J.B. Gning and F.Z. Chaoui (2010b). Frequency Identification of Wiener Systems Containing Nonparametric Memory Switch Operator. In: Proc. American Control Conference, Baltimore, MD, USA, 3263-3268.

Tao, G., & Canudas de Wit, C. Eds. (1997). Special issue on adaptive systems with non-smooth nonlinearities. *Int. J. Adapt. Control Signal Process.* **11**.

Tao, G. and P.V. Kokotovic (1993). Adaptive control of systems with backlash. *Automatica* **29,** 323-335.

Vörös, J. (2001). Parameter identification of Wiener systems with discontinuous nonlinearities. *Systems and Control Letters* **44,** 363-372.

Vörös, J. (2003). Modeling and identification of Wiener systems with two-segment nonlinearities. *IEEE Trans. Control Systems Technology* **11,** 253-257.

Vörös, J. (2007). Parameter identification of Wiener systems with multisegment piecewise-linear nonlinearities. *Systems and Control Letters* **56,** 99-105.

Vörös, J. (2009). On modeling and identification of systems with general backlash. In: Proc. 17th Int. Conf. Process Control, Štrbské Pleso, Slovakia, 234–237.

Vörös, J. (2010a). Modeling and identification of systems with backlash. *Automatica* **46,** 369-374.

Vörös, J. (2010b). Identification of cascade systems with backlash. *International Journal of Control* **83,** 1117-1124.

Vörös, J. (2010c). Compound Operator Decomposition and Its Application to Hammerstein and Wiener Systems. In: *Block-oriented Nonlinear System Identification* (F. Giri & E.-W. Bai Eds.). Lecture Notes in Control and Information Sciences, **Vol. 404** pp. 35-51 Springer-Verlag Berlin Heidelberg.

On-line identification simulation of forgetting methods to track time varying parameters using alternative covariance matrix

J. Vachálek*

* Institute of Automation, Measurement and applied Informatic, Faculty of Mechanical Engineering, Slovak University of Technology, Nám. Slobody 17, 812 31 Bratislava, SK
(Tel.: +421 7 5249 7193, e-mail: jan.vachalek@ stuba.sk)

Abstract: The paper compares abilities of forgetting methods to track time varying parameters of two different simulated models with different types of excitation. The observed parameters in simulations are the integral sum of the Euclidean norm of a deviation of the parameter estimates from their true values and a selected band prediction error count. As supplementary information we observe the eigenvalues of the covariance matrix. In the paper we used modified method of Regularized Exponential Forgetting with Alternative Covariance Matrix (REFACM or REZAKM) along with Directional Forgetting (DF or SZ) and three standard regularized methods.

Keywords: online identification, time varying parameters, covariance matrix, forgetting

1. INTRODUCTION

This paper is devoted to online identification methods and their practical application possibilities along with adaptive control; while monitoring long-run operation of time variant dynamic systems. Emphasis is set on long-run operation and therefore the working mechanism with non-informative data. The process of algorithm realization is elaborated as well. Online identification methods are explored, where non-informative data which could possibly destabilize numerical computation of the identified system parameters, is weighted by the chosen method to ensure „forgetting". The contribution of this paper lies in two newly created algorithms and their modifications for online identification; based on the technique utilizing an alternative covariance matrix. All algorithms are validated by simulations in Matlab Simulink software environment. Finally, the results obtained through the simulation algorithms mentioned in the article are compared to other commonly used algorithms.

2. PROBLEM STATEMENT

The Let as consider a stochastic system on which observations are made at discrete time instants k = 1, 2... . A directly manipulated input u_k and an indirectly affected output y_k (booth possibly multivariate) can be distinguished in the data pair $d_k = (u_k, y_k)$. The collection of all data observed on the system up to time t is denoted by $D_t = (d_1, d_2, ... d_t)$. The dependence of a new pair of data (u_k, y_k) on previous observations D_{k-1} can be described by a conditional probability density function (p.d.f) with the following structure

$$p(y_k, u_k | D_{k-1}, \theta_k) = p(y_k | u_k, D_{k-1}, \theta_k) p(u_k | D_{k-1}) \qquad (1)$$

Incomplete knowledge of the system behavior is expressed through a vector of unknown, time varying parameters $\theta_k \in \theta$. Note that the input generator described by the second term does not depend on these parameters directly, it is expected to utilize only prior information and information contained in observed data. The first term actually characterizes the system.

3. REF AND SLZ TECHNIQUE

Suppose that no explicit model of parameter changes is known. Yet, we can quantify our prior information (and possibly information taken from data already available) by introducing an alternative probability density function (p.d.f.) $p^*(\theta_{k+1}/D_k)$. The problem is then to construct (p.d.f.) $p(\theta_{k+1}/D_k)$ based on two hypotheses described by the p.d.f. $p(\theta_k/D_k)$ (the case of no parameter changes) and the alternative p.d.f. $p^*(\theta_{k+1}/D_k)$ (the case of worst expected changes). For simplicity in this section we use the notation $p_0(\theta)$, $p_1(\theta)$ a $p^*(\theta)$ for the posterior, alternative and resulting p.d.f.'s, respectively. In Kulhavý and Kraus (1996), formulated the task of choosing p^* given p_0 and p_1 as a Bayesian decision making problem. In the next we will make a short review of their solutions, let

$$p_{\hat{\theta},P} = \frac{1}{\sqrt{2\Pi}} |P|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\theta-\hat{\theta}\right)' P^{-1}\left(\theta-\hat{\theta}\right)\right) \qquad (2)$$

where $\hat{\theta}$ and P denote the mean and covariance of a particular p.d.f. then the following solutions were shown:

EF:

$$\hat{\theta}^* = \hat{\theta}_0, P^{*-1} = \lambda P_0^{-1} + (1-\lambda) P_1^{-1} \qquad (3)$$

LF:

$$\hat{\theta}^* = \hat{\theta}_0, P^* = \lambda P_0 + (1-\lambda)P_1 \tag{4}$$

Let's consider the model of system with time varying parameters θ_k, see Kulhavý and Kraus (1996). In order to be able to track parameter variations we complement the standard recursive last square (RLS) algorithm by exponential or linear forgetting according to (3) or (4) respectively. In addition the alternative mean is set equal to posterior mean $\hat{\theta}^{\mathrm{alt}}_{k+1|k} = \hat{\theta}_{k|k}$ and for simplicity the alternative covariance is set equal to the prior covariance $P^{\mathrm{alt}}_{k+1|k} = P_{1,0} = Q$. With this choice we can use general forgetting algorithm with the following choice of forgetting operator

$$F\{P_{k|k}, Q\} = \left[\lambda \quad P_{k|k}^{-1} + (1-\lambda)Q^{-1}\right]^{-1} \tag{5}$$

which construct harmonic mean for REF (or REZ in tabs) and

$$F\{P_{k|k}, Q\} = \lambda \quad P_{k|k} + (1-\lambda)Q \tag{6}$$

which construct arithmetic mean for SLF (or SLZ in tabs). In both cases the prior covariance matrix Q isn't forgotten and is repetitively taking into account in every step k see in Schmitz et al. (2003).

4. AUGMENTING REF AND SLF WITH ACM

The involved SLF and REF augmentation considers addition and keeping the initial information in the Alternative Covariance Matrix (ACM or AKM) form. The augmentation is based on the modified Dyadic reduction algorithm, where instead of adding a-priori covariance matrix Q, ACM is computed at each step. ACM is stabilizing the evolution of matrix P(0) after the recursive update. This operation is necessary for the SLF and REF algorithms to be augmented by the stabilization component in the ACM form. The aforementioned stabilization component prevents the destabilization of the original algorithms at long running employments; when slow time changes are to be expected in the observed parameters in relation to the sampling period. The modified algorithms have been named as follows: the modified and ACM augmented REF algorithm is to be called REFACM (or REZAKM in tabs), the modified SLF algorithm augmented by ACM will be named SLFACM (or SLZAKM in tabs).

5. SIMULATIONAL ALGORITHM VERIFICATION METHODOLOGY

Two different models were created for the verification of the properties of the introduced algorithms in the observation of time variant parameters of dynamic systems.. These two models (model no. 1. and no. 2.) have a different approach to input excitation (input signal generator A and B). Algorithm quality has been compared through the use of DF, which is considered to be standard in the field. The Praly Forgetting (PF or PZ in tabs) algorithm featured in the work Praly (1993) has been also used, using the weighted covariance matrix P. All algorithms were subject to the same test with the identical length using the two featured models.

All results were graphically evaluated, and analyzed in a table where algorithm quality has been shown numerically through parameters IS and PE.

5.1 *Description of model no. 1 and no. 2*

In the case of model 1, a second order model is considered with external disturbance $v_{(t)}$ according to:

$$y_k = \sum_{i=1}^{2} a_i y_{k-i} + \sum_{i=0}^{2} b_i u_{k-i} + \sum_{i=0}^{2} b_i u_{k-i} + e_t, e_k \approx N(0, \sigma^2) \tag{7}$$

The values of constant parameters are given by: a_2 = -0.9, b_0 = 0.5, b_1 = -0.25, b_2 = 0.1, d_1 = 0.8, d_2 = 0.2 and σ = 0.1.

The time variant parameter has been chosen as $a_{(1)}$ = 0.98, which has been kept constant half of the n simulation steps, then at time t = n / 2 changed its value to $a_{(1)}$ = − 0.98. The outside disturbance has been simulated as a square signal periodically changing its value from +1 to −1 each hundred simulation steps. The identification has been made difficult mainly by the rarely occurring disturbances, which contained minimal information about the parameter $d_{(i)}$.

For the needs of the simulation, two input signal generators have been assumed:

- Input signal generator A: discrete white noise generator
- Input signal generator B: the input signal has been generated using the following equation: $u_k^* = 0.8u_{k-1}^* + 0.2u_k$, where u^*_k is normally distributed white noise and u^*_{k-1} is the previous input value.

For model no. 2. only one change has been realized in comparison to model no. 1. This has been carried out by altering the time variant parameter $a_{(1,k)}$ = 0.98 cos(2π k/250). In this case, two different input generators were considered as well:

- Input signal generator A: discrete white noise generator
- Input signal generator B: the input signal has been generated similarly to model no 1., where $u_{(k)}$ has been only chosen from the interval $u_{(k)}$ ~ (0.5, 1).

6. VERIFICATION – MATLAB SIMULINK

From the previously mentioned algorithms DF has been chosen along with the three regularized methods: REF, SLF and PF. For the simulation verification a set of S-Function libraries has been created along with a common universal user interface. This interface (Figure 1. and Figure 2.) allows the user to select input data, simulated model and the observed algorithm. Output of the discussed simulations is a graphical representation of the observed parameters along with a data file containing the results for the following analysis. Integral sum (IS) of the Euclidian norm of parameter error and prediction error PE has been shown, which is the amount exceeded by the interval ±3σ2. The simulation experiments will be marked by the character pair XY, where X is the number of the utilized model (no. 1 or no. 2) and Y represents the generator utilized (A, respectively B).

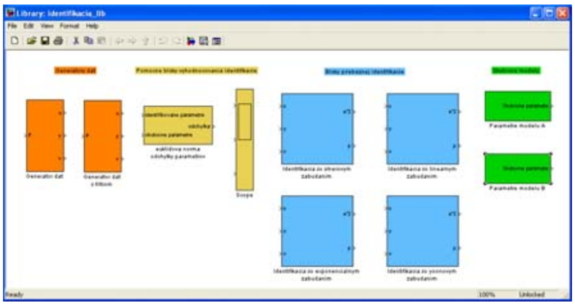


Fig. 1. User interface with selection of blocks

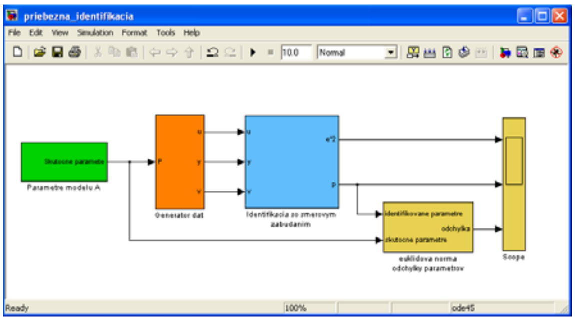


Fig. 2. Simulation block scheme in Simulink

7. EVALUATION OF SIMULATION RESULTS

This section introduces all results in a table form. For the detailed description of algorithm behavior during the simulations with different lengths, simulations lasting n = 1200, 6000, 12 000 and 120 000 have been evaluated as featured in Tables (1) to (4). It is clear from Table (3); that using simulation length n = 12 000 steps the artefacts of long lasting runs are already appearing. The result is the confirmation of REFACM algorithm quality in comparison to REF, which in the case 1A achieved better results then REF. Excellent results are achieved by the algorithm PF also. The least satisfactory performance is provided by the algorithm SLFACM. The data featured in Table (4) fully confirm the previous considerations of the REFACM algorithm quality. It is clear that using ACM as if a constraint has been enforced on parameter trending, which also implies the improvement of IS parameters in comparison of the results achieved by REF. The convergence of the REF covariance matrix is faster and finite in contrast to REFACM, where the convergence is slower and also the addition of excited ACM cannot be finite. The achieved simulation results and REFACM algorithm behaviour at 1200 and 120 000 simulation steps, show that as the running length increases the quality improves in contrast to REF. In Table (5) we have shown influence of weighting factor λ to quality of REFACM algorithms. Best results in simulations we observers in settings λ=0.8.

1 200 krokov	1A	**1B**	2A	**2B**

SZ	IS=409,7
	PE=151
		IS=390,7
		PE=70
			IS=689,6
			PE=326
				IS=805,7
				PE=143

SLZ	IS=213,7
	PE=24
		IS=129,0
		PE=13
			IS=208,2
			PE=50
				IS=440,1
				PE=24

REZ	IS=68,5
	PE=16
		IS=72,1
		PE=11
			IS=168,7
			PE=38
				IS=258,2
				PE=26

PZ	IS=89,3
	PE=23
		IS=112,2
		PE=14
			IS=242,0
			PE=38
				IS=478,4
				PE=37

SLZAKM	IS=780,2
	PE=200
		IS=779,6
		PE=141
			IS=1022,7
			PE=110
				IS=1012,2
				PE=251

REZAKM	IS=103,9
	PE=29
		IS=101,1
		PE=36
			IS=339,6
			PE=110
				IS=474,1
				PE=65

Tab. 1. Simulation length 1200 steps

6 000 krokov	1A	**1B**	2A	**2B**

SZ	IS=356,6
	PE=130
		IS=957,3
		PE=131
			IS=3499,4
			PE=1723
				IS=4016,0
				PE=717

SLZ	IS=153,4
	PE=18
		IS=208,6
		PE=12
			IS=928,5
			PE=210
				IS=2683,3
				PE=131

REZ	IS=118,3
	PE=18
		IS=120,3
		PE=12
			IS=740,9
			PE=174
				IS=1336,7
				PE=103

PZ	IS=125,9
	PE=23
		IS=114,4
		PE=13
			IS=921,9
			PE=197
				IS=1807,2
				PE=173

SLZAKM	IS=595,9
	PE=247
		IS=1062,3
		PE=98
			IS=4817,1
			PE=2282
				IS=7111,0
				PE=1405

REZAKM	IS=127,4
	PE=27
		IS=177,8
		PE=34
			IS=963,8
			PE=239
				IS=1985,1
				PE=166

Tab. 2. Simulation length 6000 steps

12 000 krokov	1A	**1B**	2A	**2B**

SZ	IS=645,7
	PE=253
		IS=1604,6
		PE=214
			IS=6701,0
			PE=3892
				IS=7474,5
				PE=1488

SLZ	IS=219,2
	PE=18
		IS=257,5
		PE=13
			IS=1533,4
			PE=363
				IS=3589,9
				PE=179

REZ	IS=168,9
	PE=13
		IS=253,9
		PE=14
			IS=1457,2
			PE=351
				IS=2525,3
				PE=188

PZ	IS=103,5
	PE=14
		IS=165,4
		PE=15
			IS=1760,8
			PE=379
				IS=3210,8
				PE=355

SLZAKM	IS=721,4
	PE=174
		IS=1365,4
		PE=82
			IS=9640,0
			PE=5105
				IS=12951
				PE=2793

REZAKM	IS=162,9
	PE=21
		IS=296,3
		PE=31
			IS=1844,9
			PE=654
				IS=4507,3
				PE=439

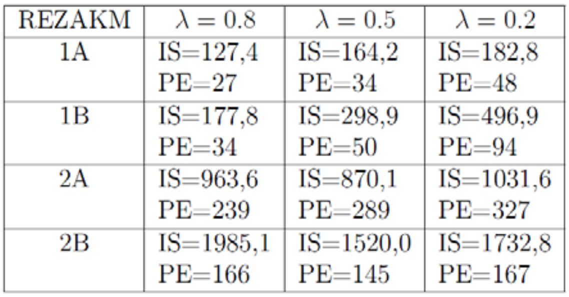
Tab. 3. Simulation length 12 000 steps

120 000 krokov	REZ	**REZAKM**

1A	IS=1035,4
	PE=17
		IS=880,8
		PE=51

1B	IS=3419,5
	PE=12
		IS=2963,7
		PE=29

Tab. 4. Simulation length 120 000 steps



REZAKM	$\lambda = 0.8$	$\lambda = 0.5$	$\lambda = 0.2$

1A	IS=127,4
	PE=27
		IS=164,2
		PE=34
			IS=182,8
			PE=48

1B	IS=177,8
	PE=34
		IS=298,9
		PE=50
			IS=496,9
			PE=94

2A	IS=963,6
	PE=239
		IS=870,1
		PE=289
			IS=1031,6
			PE=327

2B	IS=1985,1
	PE=166
		IS=1520,0
		PE=145
			IS=1732,8
			PE=167

Tab. 5. Influence of weighing factor λ

8. CONCLUSTION

The simulation verification tests featured in the previous section, evaluated in the Matlab Simulink environment confirm that the quality of the tested algorithms is diverse. From the viewpoint of our interest, that is the long running simulations, the best results are achieved by the REFACM algorithm – the contribution of this paper. The quality of REFACM in comparison with the other algorithms confirms the advantages of using ACM given the specific conditions featured in this work.

ACKNOWLEDGMENTS

The investigation reported in the paper was supported by Slovak Grant Agency APVV, project ID: APVV-0090-10 and APVV-0160-07. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

REFERENCES

Schmitz, U., Haber R., Bars R. (2003). *A predictive On-Off controller for nonlinear processes.* 14th. Int. Conference Process Control 2003, June 8-11, 2003, Štrbské Pleso, Slovakia

Vachálek, J. (2004). *Priebežná identifikácia laboratórneho modelu s využitím dátového úložiska pre množinu linearizovaných modelov.* Proceedings the 6th international scientific-technical conference Process control 2004, ŘÍP 2004, 8-11 June 2004, Kouty nad Desnou, Czech republic

Kulhavý, R., Kraus, F.J. (1996). *On Duality of regularized Exponencial and Linear Forgetting.* Automatica, 1996, vol. 32, No 10, pp. 1403-1415.

Praly, L. (1993). *Robustness of model reference adaptive control.* In 3th Yale WorkShop on application of adaptive system theory, 1993, Yale University, USA

Simulation of 2D physics of objects captured by web camera using OpenCV and Box2D

Michal Sedlák *

* *Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia (e-mail: michal.sedlak@stuba.sk)*

Abstract: The paper presents one approach to simulation of physics applied on objects captured by web camera. Introduced approach utilise OpenCV library for image capturing and contour detection. Objects detected by OpenCV are reconstructed from its outlines in Box2D environment so the physics can be applied to it. Because of restrictions of Box2D we needed to do approximation and scaling of outlines and tessellation of objects with Delaunay triangulation algorithm.

Keywords: OpenCV, Python, Box2D, physics, tessellation

1. INTRODUCTION

This paper describes applying of Newtonian physics to objects recognized in image captured from camera. Simulation of physics is used in many modern applications. You can find it in implementations used by game engines, there are as well more complex implementation used in 3D drawing and animation programs or exact and precise simulation in CAE and CAD programs. Paper describes process of animation of objects, from a capturing phase, over detection of the object outlines and interpretation of objects in physical engine, to animation of such objects. This approach can be applied in education of physics at elementary schools, with interactive blackboards, or in computer games.

2. OBJECT DETECTION AND OPEN COMPUTER VISION LIBRARY

To apply a physics to hand drawn objects we need to identify and isolate objects from image. We have used a web camera as a source and Open Computer Vision (OpenCV) library as processing tool of the images.

2.1 OpenCV

In regards the book of Bradski and Kaehler (2008) OpenCV is a library for open source programming functions for real time computer vision, with more than five hundred optimized algorithms. It can be used with C++, C and Python. We chose Python version, which is optimized Python wrapper to C++ functions.

In the beginning we have to capture image to work with. OpenCV library has implemented methods for image capture from camera. Simple image capture is shown in Listing 1.

```
1 self.camera = cv.CaptureFromCAM(−1)
2 self.image = cv.QueryFrame(self.camera)
3 self.DetectOutline(self.image)
```

Listing 1: Query image frame from web camera

In line 1 of Listing 1 we initialize our web camera. In variable camera is allocated and initialized object that can query web camera for new image. Then as we see in Listing 1 line 2 we can get the image from camera and store it in the variable named image. Captured image is shown in Fig. 1.

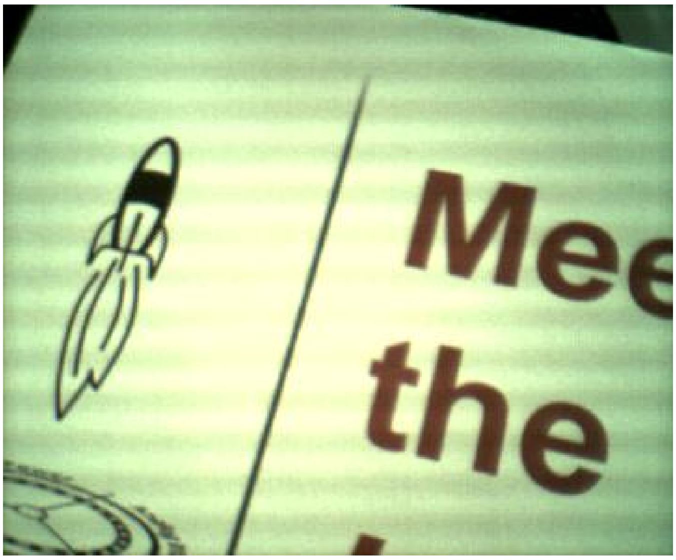


Fig. 1. Image captured form camera

Now when we have image data stored in the variable, we can process data to find outlines.

```
def DetectOutline(self, image):
  image_size = cv.GetSize(image)
  grayscale = cv.CreateImage(image_size, 8, 1)
  cv.CvtColor(image, grayscale, cv.CV_BGR2GRAY)
  cv.EqualizeHist(grayscale, grayscale)
```

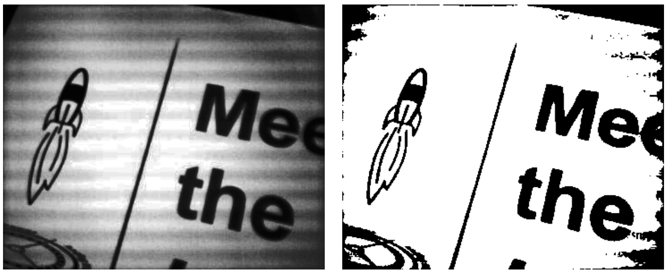
```
6   storage = cv.CreateMemStorage(0)
7   cv.Threshold(grayscale, grayscale, 50, 255,
        cv.CV_THRESH_BINARY)
8   self.contours = cv.FindContours(grayscale,
9     cv.CreateMemStorage(),
10    cv.CV_RETR_TREE,
11    cv.CV_CHAIN_APPROX_SIMPLE)
12  if len(self.contours) > 0:
13    self.contours = cv.ApproxPoly
          (self.contours,
14      storage,
15      cv.CV_POLY_APPROX_DP,
16      1.5,
17      1)
18  return self.contours
```

Listing 2: Outline detection

In function DetectOutline() in Listing 2 is shown how to find outlines of objects in image. First we convert image to grey scale as seen on Listing 2, line 3.

Then we run histogram equalization (Listing 2, line 5). Equalization makes objects better visible and gives better output for thresholding (Listing 2, line: 7) which makes black and white as you can see in Fig. 2a.

Outline detection is done with function cv.FindContours() (Listing 2, line: 8). Output of thresholding is shown in Fig. 2b.



(a) Equalize histogram (b) Thresholding

Fig. 2. Effects applied on images

After outline detection we have a tree of contours stored in the variable self.contours. These trees are iterable objects sorted from outer to inner outline connected by property h_next and v_next that we will describe in paragraph about creation of objects from outlines.

Contour can be very complicated and consist of thousands of points, which would cause objets with thousands of vertices. It is time demanding to simulate complicated objects, that is why we use polynomial approximation of the contour points. (line: 13). Visualisation of outlines is shown in Fig. 3.

Now we have all outlines stored in the outline tree structure, so we can create objects and apply a physics.

3. PHYSICS SIMULATION IN BOX2D

There is lot of physics engines that can be used for simulation of physics. Because we wanted to simulate physics only in 2D we could code our own implementation of physics, or use one of commercial or open source engines.

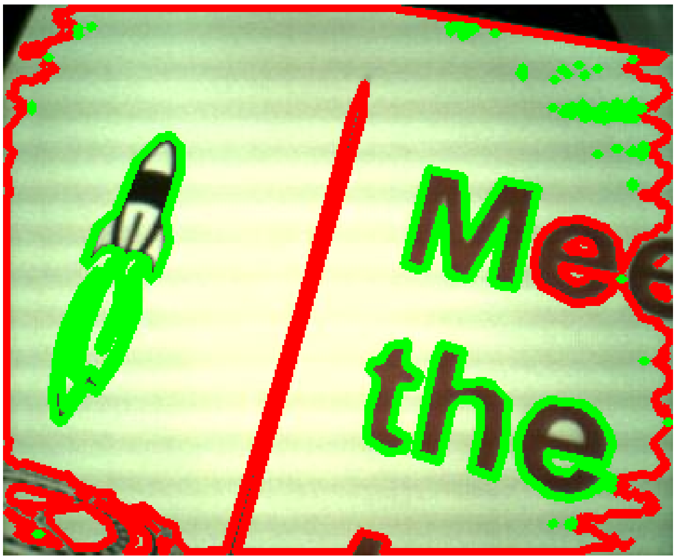


Fig. 3. Visualisation of contours

We chose Box2D [Thorn (2010)], which is open source 2D physics engine with implementation of rigid body objects and their collisions.

3.1 World

To create physics simulation we need to create world. World is object that manages memory, objects and simulation. Creation of world is shown in Listing 3:

```
1 self.worldAABB=box2d.b2AABB()
2 self.worldAABB.lowerBound = (−100.0, −100.0)
3 self.worldAABB.upperBound = ( 600.0, 600.0)
4 gravity = (0.0, −10.0)
5
6 doSleep = True
7 self.world = box2d.b2World(self.worldAABB,
      gravity, doSleep)
```

Listing 3: Creation of Box2D world

First we have to create boundaries of the world. We define them as vectors from bottom left (Listing 3, line: 2) to top right (Listing 3, line: 3). Objects have to be inside the boundaries, when an object touch the boundary it gets stuck. Then we define gravity vector (Listing 3, line: 4). The last thing before creation of the world we allow objects to sleep (Listing 3, line: 6). Object that are not moving fall asleep, then they are ignored by the engine. Last line of Listing 3 creates the world.

World is created and we are ready to create objects from outlines.

3.2 Objects

Every object that is simulated in Box2D consists of body and shapes. Our objects are described by the contour tree. To create objects we need to iterate through contour tree to find contours that belongs together and create objects for these contours.

3.3 Contour tree

Contour tree is object in which are stored points of each contour. Contours are connected by functions returning reference to other contours with h_next() and v_next(), where h_next() is reference to deeper contour, and v_next() is reference to another object contour. To iterate over all contours we have created recursive function shown in Listing: 4.

```
1  def CreateObjectsFromCountours(self, cont,
       h=0, v=0):
2    if v>0:
3      density = 10.0
4    else:
5      density = 0
6    if len(cont)>8:
7      self.CreateObject(cont,h,v)
8
9    if cont.v_next():
10     v += 1
11     self.CreateObjectsFromCountours(
           cont.v_next(), h, v)
12     v -= 1
13
14   if cont.h_next():
15     h += 1
16     self.CreateObjectsFromCountours(
           cont.h_next(), h, v)
```

Listing 4: Function to iterate through contour tree

We iterate through contour tree. First level is outer contour of the image (Listing 4 line: 2), because we do not want the outer contour to move, we set it as static by setting density to 0 (Listing 4 line: 5). Every other contour is dynamic body with density set to 10 (Listing 4 line: 3).

When we know what type of object we will create, we can create bodies and shapes for our contours.

3.4 Bodies, shapes and collisions

Bodies are backbone used by shapes. One body can contain more shapes, but one shape could be attached to only one body. Box2d is rigid body physics engine, that mean that shapes attached to body can not move against other, or body. Body have position and velocity. Forces, torques and impulses can be applied to body [Catto (2010)]. Bodies just hold the shapes and shapes are elements that collide together.

Listings 5,6,7 shows process of object creation:

```
1  def CreateObject(self, cont, h,v):
2    contM = []
3    for point in cont:
4      x = point[0]/30.0
5      y = point[1]/30.0
6      contM.append((x,y))
7
8    bd=box2d.b2BodyDef()
9    bd.position = ( 0.0, 0.0 )
10
11   edgeDef=box2d.b2EdgeChainDef()
12   edgeDef.setVertices(contM)
13
14   if v==0:
15     body = self.world.CreateBody(bd)
16     try:
17       self.contourBodies.append(body)
18     except:
19       self.contourBodies = [body]
20     body.CreateShape(edgeDef)
```

Listing 5: Creation of object from outer contour

Image size is measured in pixels and Box2D units are kilograms, meters, and seconds (KMS) we should scale images coordinates to fit in 0.1m to 10m. In that scale is performance of Box2D the best. We are doing it by dividing of value of pixel coordinates by 30.0 (Listing 5 line: 4)

Then we create a body definition that will represent our contour(Listing 5 line: 8) and set up it initial position in next line.

After that we create shape of body as chain of edges (Listing 5 line: 11) and assign the array of vertices to it (Listing 5 line: 12). Edges are special type of shapes that have no mass. Edges are represented as lines between vertices that collide with other non-edge objects. Edges are easy to create because they do not have to be concave unlike polygons.

At last we attach this shape to created body Listing 5 20. Because Box2D does not keep track about body definitions, we have to store bodies in to array for later use (Listing 5 line: 19).

Listing of the function CreateObject() continues in Listing 6. This part of function creates dynamic objects inside the outer contour. In this part we prepare list for bodies of objects, so we can modify objects that are already created or objects that we want append new shapes.

```
21 if v == 1:
22     try:
23       body = self.objectBodies[h]
24     except:
25       body = self.world.CreateBody(bd)
26       self.objectBodies[h] = body
```

Listing 6: Creation of objects

3.5 tessellation

Box2D supports only collisions between convex objects and contours of objects captured by camera are mostly not convex. So we have to break outlines to convex polygons. There is more ways how to break concave objects. We chose the 2D constrained Delaunay triangulation algorithm implemented by poly2tri Python library[Rognant et al. (1999)]. Function CreateObject() continuous in Listing 7

```
27     polyline = []
28     for (x,y) in cont:
29       polyline.append(p2t.Point(x,y))
30     cdt = p2t.CDT(polyline)
31     triangles = cdt.triangulate()
32     for t in triangles:
33       x1 = t.a.x/30.0
34       y1 = t.a.y/30.0
35       x2 = t.b.x/30.0
```

```
36        y2 = t.b.y/30.0
37        x3 = t.c.x/30.0
38        y3 = t.c.y/30.0
39        if math.hypot(x2−x1,y2−y1)<0.1:
40          x2 = x2 + math.copysign(0.1, x2−x1)
41          y2 = y2 + math.copysign(0.1, y2−y1)
42        if math.hypot(x3−x2,y3−y2)<0.1:
43          x3 = x3 + math.copysign(0.1, x3−x2)
44          y3 = y3 + math.copysign(0.1, y3−y2)
45        if math.hypot(x1−x3,y1−y3)<0.1:
46          x1 = x1 + math.copysign(0.1, x1−x3)
47          y1 = y1 + math.copysign(0.1, y1−y3)
48        poly=box2d.b2PolygonDef()
49        poly.setVertices(((x1, y1), (x2, y2),
              (x3, y3)))
50        poly.density = 1.0
51        poly.restitution = 0.0
52        poly.friction = 0.0
53        body.CreateShape(poly)
54        body.SetMassFromShapes()
```

Listing 7: Creation of objects

The creation of objects continues with tessellation. We need to assign vertices to structure that could be understood by poly2tri library (Listing 7 line: 29) and we initialize the CDT object (Listing 7 line: 30). In next line we call function that will create triangles from the vertices assigned before. These triangles are in image pixel coordinates, so we need to scale them at first (Listing 7 lines: 32-38). Now when we have triangles scaled we need to scale the triangles that are too small to triangles with size at least 0.1m because of speed optimalization, this is done in Listing 7 lines: 39-47). Now we have set of triangular shapes that could be attached to body (Listing 7 line: 53). Because these objects are compound objects, we need to set the center and amount of mass to this body. We can let Box2D set this properties based on shape information with function SetMassFromShapes() (Listing 7 line: 54). Visualisation of objects is in Fig. 4.

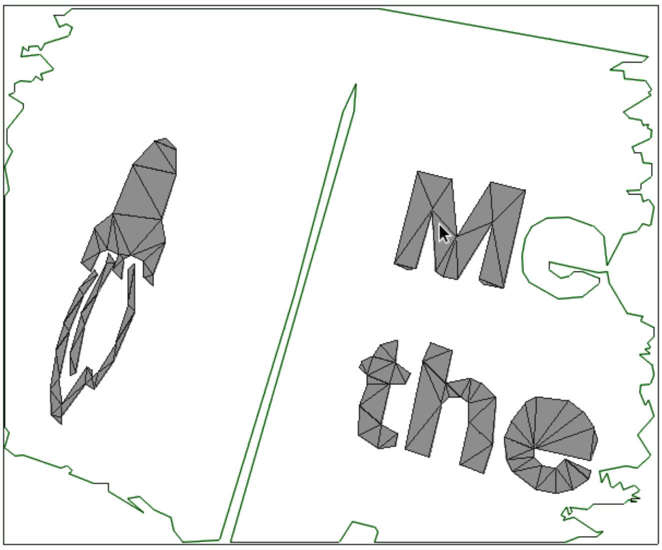


Fig. 4. Visualisation of objects after triangulation

After this we can add other objects and start simulation by function Step(). After few second of simulation are all objects on the bottom of the screen like is shown in Fig 5.

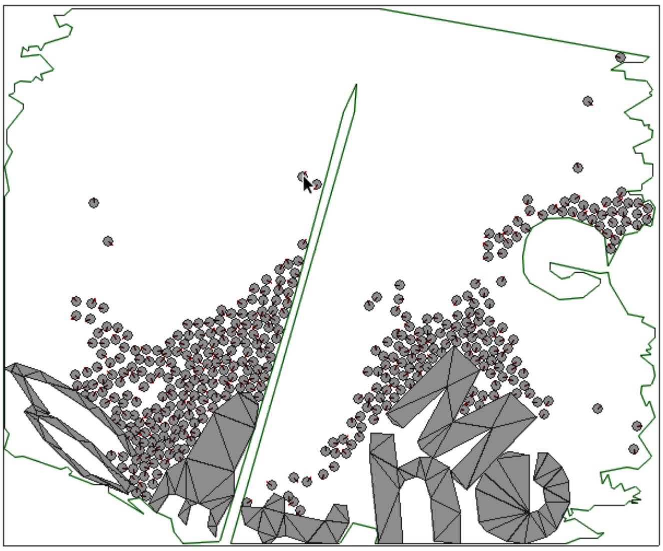


Fig. 5. Visualisation of simulation

4. FUTURE WORK

This approach can be used in interactive blackboards used for education of physics on elementary schools, in future work we plan to implement identification of some special objects like springs or joints.

We are planning to implement object tracking, and dynamic object morphing so we could interact with simulated objects. Because of Box2D is rigid body engine, is complicated to simulate physics of objects that change their shape in time. We plan to implement some soft body elements to make this possible.

The last stage will be usage of captured images as source of textures of simulated objects. Because this is only decorative element, we are planning to implement this task as a last one.

ACKNOWLEDGMENTS

This work was partially supported by the Project VEGA 1/0656/09: Integration and development of nonlinear and robust control methods and their application in controlling flying vehicles, by the project KEGA 3/7245/09 Building virtual and remote experiments for network of online laboratories. It was also supported by the grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly, Cambridge, MA, 2008.

Erin Catto. Box2D v2.0.1 User Manual. `http://code.google.com/p/pybox2d/downloads/detail?name=2.0.2%20documentation%20from%20wiki%20archive.zip`, July 2010. [Online; accessed 20-January-2011].

L. Rognant, J.M. Chassery, S. Goze, and J.G. Planes. The delaunay constrained triangulation: the delaunay stable algorithms. In *Information Visualization, 1999. Proceedings. 1999 IEEE International Conference on*, pages 147 –152, 1999. doi: 10.1109/IV.1999.781551.

Alan Thorn. *Game Engine Design and Implementation*. Jones & Bartlett Publishers, Cambridge, MA, 2010.

Reactor Furnace Control - PID and Predictive Methods Comparison

J. Mareš[*], P. Doležel[**], F.Dušek[**], A. Procházka[*]

[*]*Institute of chemical technology, Department of computing and control engineering*
Technická 5, 166 28 Prague 6, Czech Republic
e-mail: jan.mares@vscht.cz,uprt.vscht.cz
[**]*University of Pardubice, Faculty of electrical engineering and informatics, Department of process control,*
Nám. Čs. Legií 565, 532 10 Pardubice, Czech Republic
e-mail: petr.dolezel@upce.cz,www.upce.cz

Abstract: Paper deals with different techniques of nonlinear reactor furnace control. The first part briefly describes the real system (reactor furnace), which is a nonlinear system because of different heat transport mechanisms. Then different approaches to the system control are described. Firstly standard technique using PID controller, and secondly two predictive control strategies (Generalized Predictive Controller and Neural Network Predictive Controller).

1. INTRODUCTION

Different techniques of the reactor furnace control are described and compared in the paper.

Furnace is made for chemical reactor heating. The reactor provides measurements of oxidation and reduction qualities of catalyzers in the different temperatures. It is necessary to consider a nonlinear furnace behavior, because of huge range of reactor temperature (Dušek, et all.,1997).

2. REACTOR FURNACE DESCRIPTION

The furnace base is a cored cylinder made of insulative material. On the inner surface there are two heating spirals. Spirals are powered by the voltage 230 V. In the middle of the cylinder there is a reactor. The reactor temperature is measured by one platinum thermometer (see Figure 1).

The system is a thermal process with two inputs (spiral power and ambient temperature) and one output (reactor temperature). Thus, controlled variable is the reactor temperature and manipulated variable is spiral power, ambient temperature is measured error.

Nonlinearity of the system is caused by heat transfer mechanism.

When the temperature is low, heat transfer is provided only by conduction. However, when the temperature is high, radiation presents an important transfer principle.

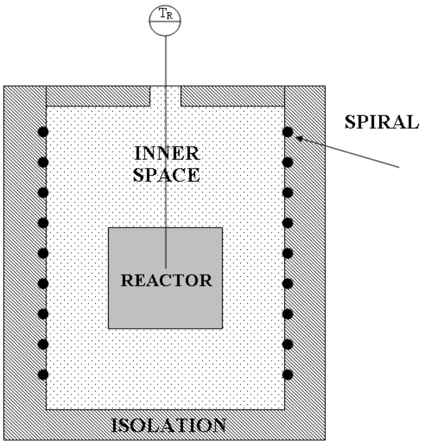


Fig. 1. Reactor furnace chart

Nonlinear mathematical model (set of four differential equations) and its linearization is described in (Mareš et all., 2009) and (Mareš et all., 2010a).

3. PID CONTROL

The first approach how to control the reactor furnace is the simplest way – PID control, where gain and time constants were set according to $T_{\Sigma.}$ method, more in (Kuhn, 1995). The method gives the PID control response slow but very robustness. Even nonlinear systems are possible to control quite satisfactorily.

The only necessity for the controller parameters estimation is to measure the step response of the system. Then we can calculate gain and parameter $T_{\Sigma.}$, see figure 2 and equation (1). Constants of the controller are calculated from these parameters, according to table 1.

$$\int_0^{T_\Sigma} y(t)dt = \int_{T_\Sigma}^{\infty} [y(\infty) - y(t)]dt \qquad (1)$$

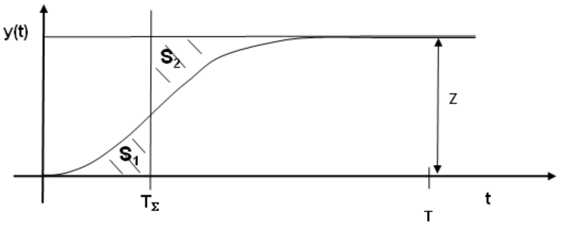


Fig. 2. T_Σ method principle

Table 1 – PID parameters calculation

	r_0	T_I	T_D
PI	0,5/Z	$0{,}5.T_\Sigma$	0
PID	1/Z	$0{,}66.T_\Sigma$	$0{,}167.T_\Sigma$

The step response was measured (step of the spiral power 0 – 100 W) and the PI controller parameters were estimated, table 2.

Table 2 – PI controller setting

	r_0	T_I	T_D
PI	3,38	223	0

The control experiment was realized at the system. Results are shown in figure 3, where the first chart shows the manipulated variable, the second chart shows the set point and controlled variable and the third chart shows the error between set point and controlled variable.

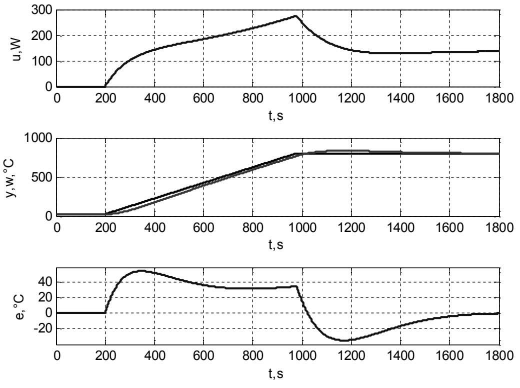


Fig. 3.PID control

4. GENERALIZED PREDICTIVE CONTROL

Generalized Predictive Control (GPC) belongs to the group of complex predictive controllers where model is needed. We assume the model in the form of equation (2).

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) + C(z^{-1})\frac{e(k)}{\Delta} \qquad (2)$$

where *A, B, C* are polynomials, *y (k)* is model output, *u (k)* is model input *e (k)* is output error and Δ is described by Δ= 1-z $^{-1}$. It is possible to convert (2) to the form of equation (3)

$$\overline{A}(z^{-1})y(t) = B(z^{-1})\Delta u(t-1) + C(z^{-1})e(t) \qquad (3)$$

where $\overline{A} = \Delta.A$.

The model is used for the calculation of the future output prediction. There are several different methods how to calculate it. One of the simplest ways (using the inverse matrix) is described in this chapter.

The prediction of N steps is possible to rewrite by the set of equations (4).

$$\begin{aligned}
&y(k+1) = b_1\Delta u(k) + b_2\Delta u(k-1) + \ldots \\
&\ldots + b_{n+1}\Delta u(k-n) - a_1 y(k) - a_2 y(k-1) - \ldots - a_{m+1}y(k-m) \\
\\
&y(k+2) = b_1\Delta u(k+1) + b_2\Delta u(k) + \ldots \\
&\ldots + b_{n+1}\Delta u(k-n+1) - a_1 y(k+1) - a_2 y(k) - \ldots - a_{m+1}y(k-m+1) \\
\\
&y(k+3) = b_1\Delta u(k+2) + b_2\Delta u(k+1) + \ldots \\
&\ldots + b_{n+1}\Delta u(k-n+2) - a_1 y(k+2) - a_2 y(k+1) - \cdots \\
&\qquad .. - a_{m+1}y(k-m+2) \\
&\vdots \\
&y(k+N) = b_1\Delta u(k+N-1) + b_2\Delta u(k+N) + \ldots \\
&\ldots + b_{n+1}\Delta u(k+N-n+1) - a_1 y(k+N-1) - \\
&\qquad - a_2 y(k+N) - \ldots - a_{m+1}y(k+N-m+1)
\end{aligned} \qquad (4)$$

In matrix form it is possible to write

$$\mathbf{A}.\begin{bmatrix} y(t+1) \\ y(t+2) \\ \vdots \\ y(t+N) \end{bmatrix} = \mathbf{B}.\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix} + \widetilde{\mathbf{B}}.\begin{bmatrix} \Delta u(t-1) \\ \Delta u(t-2) \\ \vdots \\ \Delta u(t-n) \end{bmatrix} + \widetilde{\mathbf{A}}\begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \qquad (5)$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -a_1 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -a_N & -a_{N-1} & \cdots & 1 \end{bmatrix} N\,\mathrm{x}\,N; \;\; \mathbf{B} = \begin{bmatrix} b_1 & 0 & \cdots & 0 \\ b_2 & b_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ b_N & b_{N-1} & \cdots & b_1 \end{bmatrix} N\,\mathrm{x}\,N$$

$$\widetilde{\mathbf{A}} = \begin{bmatrix} a_1 & a_2 & \cdots & a_{m+1} \\ a_2 & \cdots & a_{m+1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} N\,\mathrm{x}(m+1); \;\; \widetilde{\mathbf{B}} = \begin{bmatrix} b_2 & b_3 & \cdots & b_{n+1} \\ b_3 & \cdots & b_{n+1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} N\,\mathrm{x}\,n$$

Future output prediction of the system *y(t+i)* is possible to calculate by multiplying the equation (5) by the inverse matrix **A** $^{-1}$, equation (6).

$$\begin{bmatrix} y(t+1) \\ y(t+2) \\ \vdots \\ y(t+N) \end{bmatrix} = \mathbf{A}^{-1}.\mathbf{B}.\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix} + \mathbf{A}^{-1}.\widetilde{\mathbf{B}}.\begin{bmatrix} \Delta u(t-1) \\ \Delta u(t-2) \\ \vdots \\ \Delta u(t-n) \end{bmatrix} + \mathbf{A}^{-1}.\widetilde{\mathbf{A}}\begin{bmatrix} y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix} \qquad (6)$$

Last two terms describes only the system history, therefore it is possible to put them together to the matrix **F** and the vector of historical output and inputs $\mathbf{h} = [\mathbf{y} \quad \mathbf{u}]^T$. Thus, the equation of prediction is possible to write in the form of equation (7).

$$\mathbf{y} = \mathbf{G}.\mathbf{u} + \mathbf{F}.\mathbf{h} \qquad (7)$$

The aim of GPC is to calculate the vector of manipulated variable by minimizing of the cost function (8).

$$J = \mathbf{e}_N^T.\mathbf{e}_N + \lambda.\mathbf{u}^T.\mathbf{u} \tag{8}$$

where **e** is vector of control errors (length N), **u** is vector of manipulated variables (length N) and λ is weighting coefficient.

The cost function can be modified using output prediction (9) and set point vector **w.**

$$J = (\mathbf{w} - \mathbf{G.u} + \mathbf{F.h})^T(\mathbf{w} - \mathbf{G.u} + \mathbf{F.h}) + \lambda.\mathbf{u}^\mathbf{T}.\mathbf{u} \tag{9}$$

We can calculate the vector of manipulated variable **u** analytically using the square norm. Then we get equation (10).

$$\mathbf{u} = (\mathbf{G}^\mathbf{T}.\mathbf{G} + \lambda.\mathbf{I})^{-1}.\mathbf{G}^\mathbf{T}.(\mathbf{w} - \mathbf{F.h}) \tag{10}$$

We usually need only one actual value of the manipulated variable (the first element of the vector) therefore the final form of the control law is equation (11).

$$\Delta u = \mathbf{K}\left\{.\begin{bmatrix} w(t) \\ w(t+1) \\ \vdots \\ w(t+N) \end{bmatrix} - \mathbf{F}.\begin{bmatrix} \Delta u(t-1) \\ \Delta u(t-2) \\ \vdots \\ \Delta u(t-n) \\ y(t) \\ y(t-1) \\ \vdots \\ y(t-m) \end{bmatrix}\right\} \tag{11}$$

where **K** is the first row of matrix $(\mathbf{G}^\mathbf{T}.\mathbf{G}+\lambda.\mathbf{I})^{-1}.\mathbf{G}^\mathbf{T}$

GPC theory is formulated for the group of linear systems control but in the case of nonlinear systems it is not possible to use it because the linear model is not able to describe the nonlinear process well. Nonlinear system control needs nonlinear model or linearized model (this case).

In the case of piecewise linearized GPC we can do the linearization of the model and formulate it in the form of (3). Matrices **G** and **F** are possible to calculate from this form in defined number of linearization points, equation (7). Thus, the controller will switch between pre-calculated setting during control experiment (according to reactor temperature). Moreover, it is possible to interpolate between two adjoining settings. Nonlinear behavior of the system is substituted by piecewise linearized model. Complex description of this approach is in (Mareš et all.,2010b).

The control experiment was realized too. Results are shown in figure 4, where the description of charts is the same as in previous example.

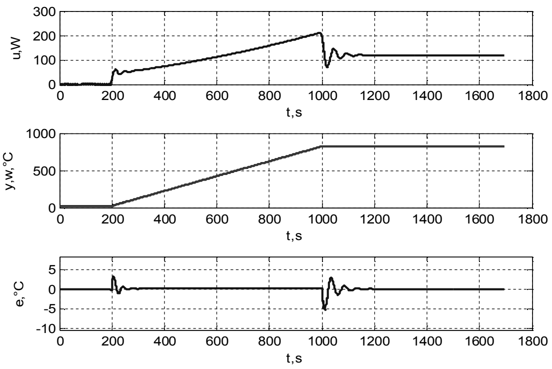


Fig. 4.Linearized GPC

5. NEURAL NETWORK PREDICTIVE CONTROL

Another approach to predictive control is described in this section. Predictive controller here uses a neural network (NN) model of nonlinear plant to predict future plant performance. The controller then calculates the control input that will optimize plant performance over a specified future time horizon.

The first stage of NN predictive control is to design a neural network which represents the dynamics of the plant. The prediction error between the plant output and NN output is used as the neural network training signal (see figure 5).

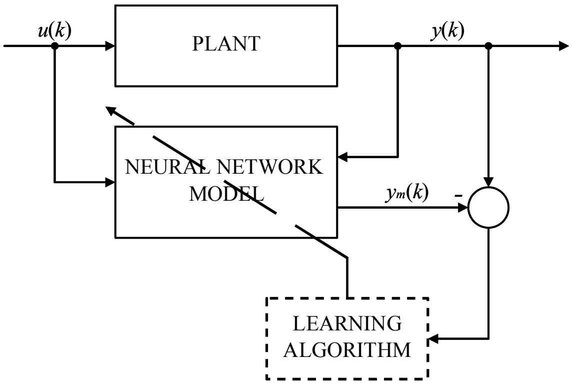


Fig. 5. – NN model identification

This neural network can be trained offline in batch mode, using data collected from some experiments with the plant. Any backpropagation algorithm can be used for network training. Process of neural network model design is discussed in detail in (Taufer et all., 2008).

In this control technique, neural network predicts the plant response over a specified time horizon. The predictions are used by some search technique to determine the control signal that minimizes the following performance criterion over the specified horizon N

$$J = \mathbf{e}_N^T.\mathbf{e}_N + \lambda.\mathbf{u}^T.\mathbf{u} \tag{12}$$

where $\mathbf{e}_N$ and **u** are the same vector as in (8).

The figure 6 illustrates the NN model predictive control process. The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of *u'*(*k*) that minimize the criterion *J*, and the optimal *u'*(*k*) is input to the plant.

It is obvious, that key part of block diagram below is optimization block or used search technique, more precisely. Mostly, optimal *u'*(*k*) is not found every sample time, because only fixed number of iterations is performed per one sample time.

Whole control technique is included in Neural Network Toolbox of Matlab.

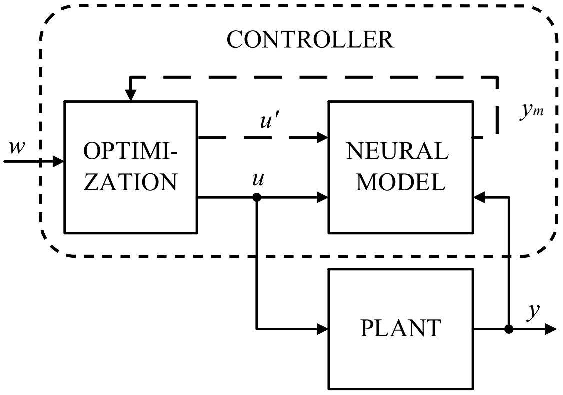


Fig. 6. NN predictive control

Control experiment with NN predictive controller was performed (Prediction horizon N = 20, λ = 0.1, golden section search routine). Neural network model was trained offline with Levenberg-Marquardt training algorithm and its topology is illustrated in figure 7. Control performance can be found in figure 8.

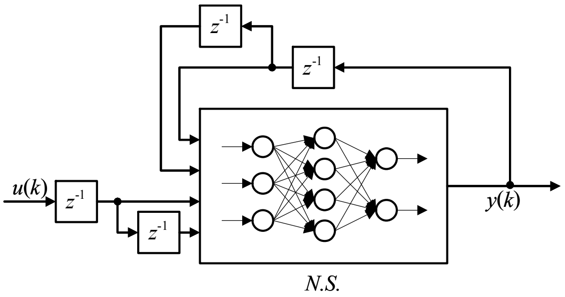


Fig. 7. NN model

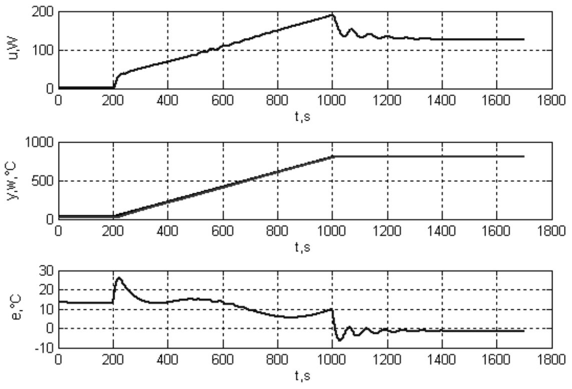


Fig. 8. NN predictive control performance

6. CONCLUSIONS

Paper deals with different techniques of real system (reactor furnace) control. As the introduction there is a brief description of the plant, which behavior is nonlinear because of the range of the reactor temperature.

The first part describes the simplest way – PID controller, where gain and time constants were set according to $T_{\Sigma.}$ method. The method gives the PID control response very slow but robustness. Therefore, nonlinear systems are possible to be controlled quite satisfactorily.

Then, the second part describes the predictive control design which uses firstly linearized mathematical model and secondly neural network model.

As conclusion, it is possible to say, that all three approaches gives satisfactory results and are able to control nonlinear system properly.

ACKNOWLEDGMENT

The work has been supported by the funds No. MSM 6046137306 and No. MSM 0021627505 of Ministry of Education of the Czech Republic. This support is very gratefully acknowledged.

7. REFERENCES

DUŠEK, F., ČIČMANEC, P., BULÁNEK, R.: *Řízení pece na zadaný průběh*In: 11th Conference PROCESS CONTROL '97, Tatranské Matliare 8.-11.6.1997, Volume 2,s.

KUHN, U. *A practical tuning rule for PID controllers: the t-sum design.* Automatisierungstechnische Praxis, 37(5), 10-16, 1995.

MAREŠ, J., DUŠEK, F., HONC, D. *Náhrada reálného zařízení modelem v Simulinku.* Sborník příspěvků z Technical Computing Prague 2009. Praha, KANAG – TISK, s.r.o. 69, 119 str. ISBN 978-80-7080-733-0.

MAREŠ, J., DUŠEK, F. *Nelineární a linearizovaný model reaktorové pece.* In *Proceedings of Conference ARTEP'10"*, 24.-26. 2 2010a.Technická univerzita Košice, 2010. Pp. 27-1 – 27-14. ISBN 978-80-553-0347-5.

MAREŠ, J., DUŠEK, F. *Reactor furnace predictive control.* In Proceedings of XXXVth Seminary ASR'10 „Instruments and Control", VSB- Technical University Ostrava, 2010b.

TAUFER, I.; DRÁBEK, O.; SEIDL, P. Umělé neuronové sítě – základy teorie a aplikace (10), *CHEMagazín*, 2008, vol. XVII, n. 1, p. 35-37. ISSN 1210-7409.

MPC-based approximation of dual control by information maximization

Jan Rathouský [*] **Vladimír Havlena** [**]

[*] *Department of Control Engineering, Czech Technical University in Prague, Czech Republic (e-mail: rathojan@fel.cvut.cz)*
[**] *Department of Control Engineering, Czech Technical University in Prague, Czech Republic (e-mail: havlena@control.felk.cvut.cz)*

Abstract: This paper proposes multiple-step active control algorithms based on MPC approach that approximate persistent system excitation in terms of the increase of the lowest eigenvalue of the parameter estimate information matrix. It is shown how the persistent excitation condition is connected with a proposed concept of stability of a system with uncertain parameters. Unlike similar methods, the proposed algorithms predict the information matrix for more than one step of control. The problem is formulated as an MPC problem with an additional constraint on the information matrix. This constraint makes the problem non-convex, thus only locally optimal solutions are guaranteed.

Keywords: Adaptive systems and control; Input and excitation design; Stochastic control; Dual control approximation; Model predictive control.

1. INTRODUCTION

Controller design is usually based on some performance specifications that should be satisfied for a system model. Thus the controller design is primarily based on some model that describes the system up to a certain precision. Various algorithms exist that take into account the model uncertainty, based on uncertainty model both in time and frequency domain. Usual algorithms, however, do not take into account the possibility that the control process itself may bring some information about the controlled system and thus improve the model.

The simultaneous optimal control and identification problem is referred to as a dual control problem Feldbaum (1960–61) which is known to be analytically solvable for only very special systems as in Sternby (1976); Åström and Helmersson (1986) as it requires solving the Bellman equation (Bertsekas (2005)). Numerical solution faces the curse of dimensionality problem. There exist approximations of the optimal solution based on suboptimal solution of the original problem,(Lee and Lee (2009); Lindoff et al. (1999, 1998); Wittenmark (1995, 2002); Chen and Loparo (1991)), or on problem reformulation (Filatov et al. (1996)). An overview of the state-of-the-art methods is given in Filatov and Unbehauen (2004), where an algorithm with dual properties is defined as one that actively gathers information during the control process while satisfying given control performance.

In this paper we propose three algorithms based on the idea of the persistent system excitation (Goodwin and Sin (1984)). The persistent excitation condition requires that the information about the system in the sense of its parameter information matrix is increased linearly, i.e.

$$P_{t+M}^{-1} - P_t^{-1} \geq \gamma I \tag{1}$$

for all t and some given M, where P_k^{-1} denotes the information matrix (i.e. the inverse of the variance matrix) after k steps of control, γ is a given real constant and I denotes the identity matrix of corresponding order. The inequality symbol $>$ is used in the positive definiteness meaning, i.e. for two matrices A and B, $A > B$ means that $A - B$ is a positive definite matrix.

The proposed algorithms are based on a constrained MPC control design that is adjusted such that the persistent excitation condition is satisfied. This control problem is formulated and analysed in section 3. In section 2 we show a motivation example for such design and propose a concept of stability of a system with uncertain parameters. This concept is based on a requirement that the mean value of a given quadratic criterion is finite over infinite control horizon. It is shown on the motivation example that the persistent excitation condition (1) is also sufficient for stability in this sense.

The proposed algorithms predict the information matrix over more than one step of control. This prediction is one of the two major problems of the methods, as the only practically computable prediction based on certainty equivalence assumption is used. The second major problem is the inherent non-convexity, the reason why only local solution is guaranteed to be found when using numerical methods for problem solution.

Section 4 contains derivations and descriptions of individual algorithms. All algorithms are designed for autoregressive systems with external input (ARX), although their formulation allows for future generalization for ARMAX systems with known MA part (Havlena (1993); Peterka

[⋆] This is a preliminary version of the paper Multiple-step active control with dual properties, which will be presented at the 18th IFAC World Congress, 2011, in Milan.

(1986)). Finally, section 5 shows simulations of the proposed methods and we conclude in section 6.

2. PERSISTENT EXCITATION AND STABILITY

In this section we will introduce a concept of stability of a closed loop system with fixed but unknown parameters and show how this concept is connected with the persistent excitation conditions.

2.1 The concept of stability

Let us consider a general ARX system. Such system has a form

$$y_k = \sum_{i=1}^{n} a_i y_{k-i} + \sum_{i=0}^{m} b_i u_{k-i} + e_k, \qquad (2)$$

where u_i and y_i are system inputs and outputs, respectively and e_i is a discrete-time white noise. Let us assume that the system parameters a_i and b_i are fixed but unknown constants. Also let us assume that at the time $k = 0$ we have some estimate of the parameter values, that we will denote $\widehat{a}_{i,0}$ and $\widehat{b}_{i,0}$. These parameter estimates can be used for controller design and they are expected to get more precise during the future control process. If the estimate is unbiased, the estimate error at time k, $\tilde{a}_{i,k} = a_i - \widehat{a}_{i,k}$ and $\tilde{b}_{i,k} = b_i - \widehat{b}_{i,k}$ is a random vector with zero mean and variance matrix P_k.

Because the future estimates of parameters are not available at the initial time $k = 0$, it is convenient to model such situation by a stochastic process, the parameters of which are random variables $a_{i,k}$ and $b_{i,k}$ with mean $\widehat{a}_i = \widehat{a}_{i,0}$ and $\widehat{b}_i = \widehat{b}_{i,0}$ and variance matrix P_k. The advantage is that properties of the estimate errors remain unchanged. It will also be supposed for simplicity of computations that $a_{i,k}$ and $b_{i,k}$ are independent with respect to time k.

Let us now consider a linear quadratic (LQ) controller for this system. The controller minimizes the following criterion

$$J_N = \sum_{k=1}^{N} \left\{ r u_k^2 + y_k^2 \right\}, \qquad (3)$$

minimization of which leads to a feedback control law. One way to cope with unknown system parameters in controller design is to use the certainty equivalence (CE) approach, i.e. substitute these parameters with their mean value. In the previous model, it means to use $\widehat{a}_i$ and $\widehat{b}_i$ instead of $a_{i,k}$ and $b_{i,k}$, respectively.

The question now is, whether such control will be stable. If the real parameters are far from their mean values, the LQ control based on CE becomes unstable. If the set of parameters for which the closed loop system becomes unstable has a constant nonzero probability, then the criterion mean $\mathsf{E}\, J_N$ will go to infinity as $N \to \infty$. This is the case, when the parameter estimate is not updated during the control process and its variance remains unchanged. The only way to make the criterion mean $\mathsf{E}\, J_N$ converge to a finite value is to make the probability of the unstable set of parameters decrease sufficiently fast to zero. Based on the previous analysis, we can define the stability of a closed loop system in the following way: A closed loop system is stable, if $\mathsf{E}\, J^{\infty} = \lim_{N\to\infty} \mathsf{E}\, J_N < \infty$.

We will now show on a simple example that if the variance of the parameters decreases as N^{-1}, or equivalently, if its inverse (or information) increases linearly, than the stability is guaranteed for a CE feedback LQ controller. The condition of a linear information growth is also called a condition of persistent excitation in identification theory and guarantees that the parameter estimates converge fast to the real values.

2.2 Derivation of stability condition

As stated before, we will now show that a linear growth of information is sufficient to guarantee stability in the previously defined sense. We will not show a formal proof but rather use a simple example to demonstrate the idea.

Let us consider the following simple discrete integrator system

$$y_k = y_{k-1} + b_k u_k + e_k, \qquad (4)$$

with only one unknown parameter b_k, that is modeled as a random variable in compliance with the previous subsection. For $r = 0$ in (3), the CE feedback control law is

$$u_k = -\frac{1}{\widehat{b}} y_{k-1}, \qquad (5)$$

where $\widehat{b}$ is the parameter mean value. The control law is defined for all nonzero $\widehat{b}$, which is exactly the condition for the system to be controllable. The system output is then

$$y_k = y_{k-1} - \frac{b_k}{\widehat{b}} y_{k-1} + e_k = \frac{\tilde{b}_k}{\widehat{b}} y_{k-1} + e_k. \qquad (6)$$

The noise e_k will be further omitted for simplicity, as it does not change the result. From (6) it follows that

$$y_k = \frac{\prod_{i=1}^{k} \tilde{b}_i}{\widehat{b}^k} y_0, \qquad (7)$$

and

$$J_N = \sum_{k=1}^{N} \left\{ y_k^2 \right\} = \sum_{k=1}^{N} \left\{ \frac{\prod_{i=1}^{k} \tilde{b}_i^2}{\widehat{b}^{2k}} y_0^2 \right\}. \qquad (8)$$

The mean of the criterion is then

$$\mathsf{E}\, J_N = \mathsf{E} \sum_{k=1}^{N} \left\{ y_k^2 \right\} = \sum_{k=1}^{N} \mathsf{E} \left\{ \frac{\prod_{i=1}^{k} \tilde{b}_i^2}{\widehat{b}^{2k}} y_0^2 \right\}. \qquad (9)$$

The criterion will only converge to a finite value, if the elements of the series converge to zero fast enough. But, using the independence assumption,

$$\mathsf{E}\, y_k^2 = \mathsf{E} \left\{ \frac{\prod_{i=1}^{k} \tilde{b}_i^2}{\widehat{b}^{2k}} y_0^2 \right\} = \frac{\prod_{i=1}^{k} \sigma_i^2}{\widehat{b}^{2k}} y_0^2, \qquad (10)$$

where σ_i^2 is the variance of b_i. Now, if the linear growth of information is guaranteed, i.e. $\sigma_i^2 = \frac{\sigma_1^2}{i}$, it holds

$$\mathsf{E}\, y_k^2 = \left(\frac{\sigma_i}{\widehat{b}} \right)^{2k} \frac{1}{k!} y_0^2, \qquad (11)$$

so the series is convergent for any σ_1 and any nonzero $\widehat{b}$. This idea can be even generalized for varying $\widehat{b} = \widehat{b}_k$, if it is guaranteed that $|\widehat{b}_k| > \epsilon$ for all k and some $\epsilon > 0$.

3. PROBLEM FORMULATION AND ANALYSIS

This paper deals primarily with ARX systems that are usually given in a form of the following equation

$$y_k = z_k^T \theta + e_k = x_k^T \theta_x + u_k b_0 + e_k, \qquad (12)$$

where $\theta = [b_0, a_1, b1, \ldots, a_n, b_n]^T = [b_0, \theta_x^T]^T$ is a vector of parameters and $z_k = [u_k, y_{k-1}, u_{k-1}, \ldots, y_{k-n}, u_{k-n}]^T = [u_k, x_k^T]^T$.

The presented algorithms, however, are derived using state-space descriptions of a linear stochastic discrete-time system (Aström (1970)), in a usual form

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ee_k \\ y_k &= Cx_k + Du_k + e_k, \end{aligned} \tag{13}$$

with the usual meaning of symbols, i.e. A, B, C, D and E are system matrices of proper dimensions, u_k, y_k and x_k are the system input, output and state, respectively and e_k is a gaussian white noise sequence with zero mean and constant finite variance.

Therefore the following nonminimal state-space representation of an ARX system (12) will be used

$$A = \begin{bmatrix} a_1 & b_1 & \ldots & b_{n-1} & a_n & b_n \\ 0 & 0 & \ldots & 0 & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 \\ . & . & . & . & . & . \\ 0 & 0 & \ldots & 1 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} b_0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} E = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$C = [\, a_1 \; b_1 \; \ldots \; b_{n-1} \; a_n \; b_n \,] \;\; D = [\, b_0 \,] \tag{14}$$

The state in this representation is x_k defined above and is directly measurable, as it is formed by previous inputs and outputs. The symbol θ has the meaning of the current estimate $\widehat{\theta}_0$ from section 2 and will be used for simplicity of notation.

3.1 Problem formulation

A standard MPC problem is formulated as a minimization problem

$$\begin{aligned} U^* = \arg\min_U J_N &= \arg\min_U \sum_{k=1}^{N} \left\{ru_k^2 + y_k^2\right\}, \\ \text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \\ & y_k = Cx_k + Du_k \\ & |u_k| \leq u_{max}, \quad |y_k| \leq y_{max} \end{aligned} \tag{15}$$

where N is the control horizon, r is a positive real tuning parameter and u_{max}, y_{max} are hard constraints on inputs and outputs, respectively. To ensure persistent system excitation, the criterion must also take into account the improvement of information gained after some amount of control inputs, i.e. the persistent excitation condition (1). In the case of an ARX system (12) it takes the form (Anderson and Moore (2005))

$$P_{t+M}^{-1} - P_t^{-1} = \sum_{k=t+1}^{t+M} \left\{z_k z_k^T\right\} \geq \gamma I, \tag{16}$$

where z_k is the system regressor at time k defined in (12). We will consider $t = 0$ for simplicity of notation, the case of general t is straightforward.

Let us now introduce some notation. It holds that $\sum_{k=1}^{M} z_k z_k^T = Z_M Z_M^T$, where $Z_M = [z_1, \ldots, z_M]$. The regressors are columns of the matrix Z_M and can be expressed as a linear function of the initial condition of the system $x_0 = [y_0, u_0, y_{-1}, u_{-1}, \ldots, y_{-n+1}, u_{-n+1}]^T$ and the input vector $U = [u_1, \ldots, u_N]^T$ as

$$z_k = F_k \begin{bmatrix} x_0 \\ U \end{bmatrix}, \quad k = 1, \ldots, M, \tag{17}$$

where F_k is a matrix of appropriate dimensions. Similarly, the rows of Z_M are formed by shifted inputs and outputs, particularly $[u_1, \ldots, u_M]$ to $[u_{-n+1}, \ldots, u_{M-n}]$ and $[y_0, \ldots, y_{M-1}]$ to $[y_{-n+1}, \ldots, y_{M-n}]$. Let us denote the k-th row of Z_M as $w_k, k = 1, \ldots, 2n+1$. Also w_k can be expressed by

$$w_k^T = G_k \begin{bmatrix} x_0 \\ U \end{bmatrix}, \quad k = 1, \ldots, 2n+1, \tag{18}$$

where G_k is a matrix of appropriate dimensions. The vector $Y = [y_1, \ldots, y_N]$ can be expressed as

$$Y = H \begin{bmatrix} x_0 \\ U \end{bmatrix}, \tag{19}$$

where H is a matrix of corresponding dimensions. Also let us call M the excitation horizon. Putting together (15) and (16) and using the introduced notation (17), (18) and (19) the problem has the form

$$\begin{aligned} U^* = \arg\min_U & \left\{rU^T U + \left[\, x_0^T \;\; U^T \,\right] H^T H \begin{bmatrix} x_0 \\ U \end{bmatrix}\right\} \\ \text{s.t.} \quad & |u_k| < u_{max}, \quad |y_k| < y_{max} \\ & \sum_{k=1}^{M} \left\{z_k z_k^T\right\} \geq \gamma I \end{aligned} \tag{20}$$

Because the suitable γ is hard to be stated apriori, it can be seen as a tuning parameter for the algorithm. We can see that there is a tradeoff between the criterion value J_N and the minimum eigenvalue γ. In some cases, it is more natural to reverse the problem – define the maximum criterion value and maximize γ within these constraints. Let us denote the optimal criterion value of the MPC problem (15) as J_N^*. Then the alternative formulation of the problem is

$$\begin{aligned} \gamma^* = & \arg\max_U \gamma, \\ \text{s.t.} \quad & \left\{rU^T U + \left[\, x_0^T \;\; U^T \,\right] H^T H \begin{bmatrix} x_0 \\ U \end{bmatrix}\right\} \leq J_N^* + \Delta J \\ & |u_k| \leq u_{max}, \quad |y_k| \leq y_{max} \\ & \sum_{k=1}^{M} \left\{z_k z_k^T\right\} \geq \gamma I \end{aligned} \tag{21}$$

for a given maximum criterion change ΔJ.

3.2 Problem analysis

The problem (20) or its alternative (21) differ from the original MPC (15) only in the last condition. As the MPC problem is convex and standard algorithms exist for its solution, the presented algorithms in fact differ only in how they cope with the last condition (16).

Because the information matrix (16) consists of quadratic and bilinear terms, both problems are non-convex in control inputs, as demonstrated in Figure 1, which shows the lowest eigenvalue of the information matrix of a second order ARX system after two steps of control as a function of the two inputs u_1 and u_2. This is a difference from simple one-step approaches where the solution always lies on the constraints (Filatov and Unbehauen (2004)) and is a reason for using numerical methods.

The second problem caused by the extra condition (16) is that the sum cannot be actually computed precisely, because it contains future outputs that do not depend only on future inputs, but also on the parameter values and input noise. A lower bound of the mean of the variance matrix could be achieved by computing $\mathsf{E}\, z_k \,\mathsf{E}\, z_k^T \le \mathsf{E}(z_k z_k^T)$ from Jensen's inequality. However, even computing $\mathsf{E}\, z_k$ is complicated, as the computation needs higher moments of the parameter joint distribution. Therefore, the conditional mean $\mathsf{E}_{\theta_0}\, z_k = \mathsf{E}(z_k|\theta_k = \theta_0, \forall k = 1, \dots, M)$ is used instead of the mean $\mathsf{E}\, z_k$.

Also note that because the information matrix increment in (16) is a sum of M dyads, its rank is less or equal to M. Therefore it is necessary that $M \ge 2n+1$ (i.e. the length of the regressor) to be able to achieve that all its eigenvalues are positive. On the other hand, N should be significantly greater than M so that the control criterion can take into account the future impact of identification procedure on the control quality. The last observation is that as the criterion minimization and information maximization are in contradiction, the persistent excitation condition may not be possible to satisfy, i.e. the problem may easily be infeasible for some choice of ΔJ and γ.

4. PROBLEM SOLUTION

In the previous section, the problem was formulated as a non-convex problem. The non-convexity introduced by (16) can be handled in several ways. This section presents three different methods to solve the problem (20) and (21).

4.1 Rank 1 algorithm

The rank 1 algorithm is based on a convex relaxation of the problem and concentrating all non-convexity into a rank constraint. Using the notation (17), (16) is rewritten as

$$\sum_{k=1}^{M} F_k \begin{bmatrix} x_0 \\ U \end{bmatrix} \begin{bmatrix} x_0^T U^T \end{bmatrix} F_k^T > \gamma I, \tag{22}$$

or in a simplified form

$$\sum_{k=1}^{M} F_k U_X F_k^T > \gamma I, \tag{23}$$

using the notation

$$\begin{bmatrix} x_0 \\ U \end{bmatrix} \begin{bmatrix} x_0^T\ U^T \end{bmatrix} = \begin{bmatrix} x_0 x_0^T & x_0 U^T \\ U x_0^T & U U^T \end{bmatrix} = U_X. \tag{24}$$

The matrix U_X consists of constant terms $x_0 x_0^T$, terms $x_0 U^T$ and $U x_0^T$ linear in U, and the term UU^T quadratic in U. The quadratic term makes the problem (23) unsolvable as an LMI directly, and therefore the following reformulation is used

$$U_{X2} = \begin{bmatrix} x_0 x_0^T & x_0 U^T \\ U x_0^T & U_q \end{bmatrix} \tag{25}$$

$$\text{s.t.} \quad \operatorname{rank}(U_{X2}) = 1, \tag{26}$$

where U_q is now a general symmetric, positive definite matrix, replacing the quadratic term UU^T. All non-convexity is now concentrated in the rank constraint (26) and dropping this constraint the task can be solved as a normal LMI problem (Boyd et al. (1994)) in more variables, known also as Schor's relaxation (Vandenberghe and Boyd (1996); Lasserre (2000)).

Expressing the criterion as a Schur complement (Bernstein (2005)) this relaxation makes it possible to solve the original problem as a rank constrained LMI

$$\begin{aligned} U^* = &\arg\min_{U} \lambda \\ \text{s.t.} \quad &\begin{bmatrix} \lambda & \begin{bmatrix} x_0^T\ U^T \end{bmatrix} H^T & U^T \\ H \begin{bmatrix} x_0 \\ U \end{bmatrix} & I & 0 \\ U & 0 & \frac{1}{r} I \end{bmatrix} \ge 0 \\ &|u_k| < u_{max}, \quad |y_k| < y_{max} \\ &\sum_{k=1}^{M} F_k U_{X2} F_k^T > \gamma I \\ &\operatorname{rank} U_{X2} = 1 \end{aligned} \tag{27}$$

Again, two versions corresponding to formulation (20) or (21) are possible.

4.2 Gershgorin circle algorithm

This algorithm is based on eigenvalue approximation in terms of Gershgorin circles (Bernstein (2005)). For a real matrix A with entries a_{ij} define $R_i = \sum_{j \ne i} |a_{ij}|$, i.e. the sum of absolute values of elements of the i-th row without the diagonal element. Then each eigenvalue lies in at least one of the Gershgorin circles defined as intervals $[a_{ii} - R_i; a_{ii} + R_i]$ for every i. This idea can be used to create constraints on the elements of the information matrix P_M^{-1}. If the diagonal elements a_{ii} are greater than some γ_1 and the nondiagonal sum less than γ_2, then the lowest eigenvalue must be greater than $\gamma_1 - \gamma_2$.

Let us now formulate the above idea as an optimization problem. The standard MPC part of the algorithm is formed by the first two lines of (20) and the additional constraints are imposed on the elements a_{ij} of the information matrix $P_M^{-1} = Z_M Z_M^T$. Using the fact that $a_{ij} = w_i w_j^T$ and notation (18), it is necessary to ensure that

$$\begin{aligned} b_{ij} &> \begin{bmatrix} x_0^T\ U^T \end{bmatrix} G_i^T G_j \begin{bmatrix} x_0 \\ U \end{bmatrix}, \forall i, j = 1, \dots, 2n+1, i < j \\ b_{ij} &> -\begin{bmatrix} x_0^T\ U^T \end{bmatrix} G_i^T G_j \begin{bmatrix} x_0 \\ U \end{bmatrix}, \forall i, j = 1, \dots, 2n+1, i < j \\ b_{ij} &= b_{ji} \\ \gamma_2 &> \sum_{j \ne i} b_{ij}, \forall i = 1, \dots, 2n+1 \\ \gamma_1 &< \begin{bmatrix} x_0^T\ U^T \end{bmatrix} G_i^T G_i \begin{bmatrix} x_0 \\ U \end{bmatrix}, \forall i = 1, \dots, 2n+1 \end{aligned} \tag{28}$$

where b_{ij} are artificial variables that have the meaning of absolute values of a_{ij}. Because the matrix P_M^{-T} is symmetrical, the first two constraints in are only required for $i < j$.

4.3 Orthogonal regressors algorithm

This algorithm is based on the idea, that the regressors shape the information ellipsoid, that is the ellipsoid $x^T(P_M^{-1})^{-1}x = x^T P_M x = 1$. The eigenvalues of P_M^{-1} correspond to the ellipsoid radii. Therefore similarly to the previous algorithm, it is necessary to ensure that the regressors' norms $|z_i| > \gamma_1$ and that the regressors are 'as much orthogonal as possible', meaning that for all $i \neq j$, $z_i^T z_j < \gamma_2$. The problem again consists of the first two lines of (20) and the following constraints

$$\begin{aligned} b_{ij} &> \begin{bmatrix} x_0^T & U^T \end{bmatrix} F_i^T F_j \begin{bmatrix} x_0 \\ U \end{bmatrix}, \forall i,j = 1,\ldots,M, i<j \\ b_{ij} &> -\begin{bmatrix} x_0^T & U^T \end{bmatrix} F_i^T F_j \begin{bmatrix} x_0 \\ U \end{bmatrix}, \forall i,j = 1,\ldots,M, i<j \\ b_{ij} &< \gamma_2, \forall i,j = 1,\ldots,M, i<j \\ \gamma_1 &< \begin{bmatrix} x_0^T & U^T \end{bmatrix} F_i^T F_i \begin{bmatrix} x_0 \\ U \end{bmatrix}, \forall i = 1,\ldots,M \end{aligned} \tag{29}$$

The structure of the problem is similar to the previous one, the difference is in the problem dimension. While the number of constraints is $\frac{(2n+1)(2n)}{2}$ and the dimension of the vectors is M in the Gershgorin algorithm, in this case it is the reverse, i.e. the dimension of regressors is $2n+1$ and the number of constraints is $\frac{(M)(M-1)}{2}$. From this follows that in this case, M should be equal to $2n+1$, as the number of regressors should not be higher than their dimension.

4.4 Stability

The stability of the proposed algorithms in the usual (Lyapunov) sense can be guaranteed for the nominal system, i.e. the system for which $\tilde{a_{i,k}} = 0$ and $\tilde{b_{j,k}} = 0$ for all $i = 1\ldots n$, $j = 1\ldots m$ and all $k = 1\ldots N$. This follows from the stability of the MPC controller (Rawlings and Muske (1993)) and from the fact that the criterion is bounded, therefore the difference $\delta u_i = u_i - u_i^*$ is square summable, u_i^* denoting the MPC optimal solution.

5. SIMULATIONS

Simulations of the previously proposed algorithms are shown in this section. The following ARX system was used

$$y_k = 1.64y_{k-1} - 0.67y_{k-2} + 0.2u_k + 0.22u_{k-1} - 0.12u_{k-2} + e_k, \tag{30}$$

which is obtained by discretization of a system $1/(s+1)^2$ with a sampling period $T_s = 0.2s$ and modified in order to have $b_0 \neq 0$. The system is controlled to zero from the initial state $x_0 = [10,0,0,0]^T$, i.e. the initial output $y_0 = 10$. Note that the nonminimal representation (14) is used, so the system order is 4. The control was designed for $N = 30$, $M = 5$, $r = 1$ and $\Delta J = 0.1J_N^*$. Figures 2 and 3 show the inputs and outputs of a control process for optimal MPC controller and all three designed controllers, respectively. Figure 4 shows the development of the variance matrix in the sense of its greatest eigenvalue.

The Rank 1 algorithm was used in the form of (21) and was solved by YALMIP (Löfberg (2004)) in MATLAB, with help of the LMIRANK solver (Orsi et al. (2006)). As the solver only searches for feasible points, the algorithm was run sequentially with γ varying according to the interval bisection method to find the maximum information. Both the Gershgorin and the regressor algorithm were solved by the MATLAB standard function FMINCON.

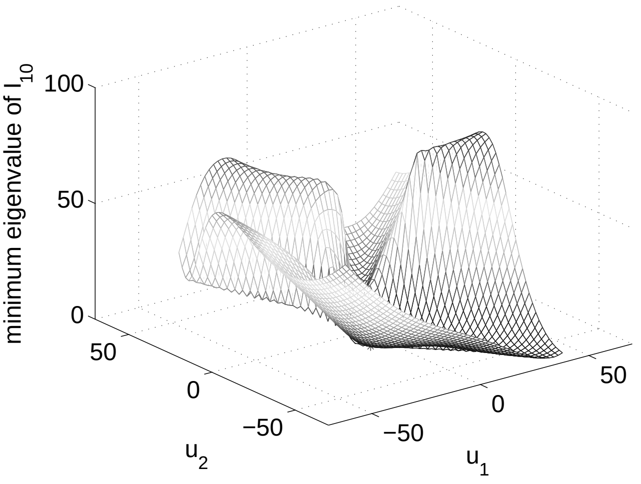


Fig. 1. The lowest eigenvalue of the predicted information matrix after $M = 5$ steps of control as a function of the two first inputs u_1 and u_2 around the optimal MPC solution for the ARX system (30)

6. CONCLUSIONS

This paper proposes three different algorithms for simultaneous identification and control, based on a standard MPC approach with a demand on the parameter information matrix in a form of the persistent excitation condition. The paper also shows a motivational example which explains the connection of the persistent excitation with a presented concept of stability.

The proposed algorithms are derived from a general formulation and in some cases it is shown that the persistent excitation may not be satisfied precisely and only approximations are found. However, simulations show that the the use of the proposed methods lead to better identification.

The drawback of all three methods is the inherent nonconvexity of the problem that causes convergence to local optima only. Therefore the performance is not guaranteed and may vary depending on the algorithms settings such as the starting point or control and excitation horizon.

ACKNOWLEDGEMENTS

This work was supported by the the grant 102/08/0442 of the Czech Science Foundation and by the internal CTU grant SGS SGS10/077/OHK3/1T/13.

REFERENCES

Anderson, B.D.O. and Moore, J.B. (2005). *Optimal Filtering.* Dover Publications.

Aström, K.J. (1970). *Introduction to Stochastic Control Theory.* Academic Press, New York.

Åström, K.J. and Helmersson, A. (1986). Dual control of an integrator with unknown gain. *Computers & Mathematics with Applications – part A*, 12(6), 653–662.

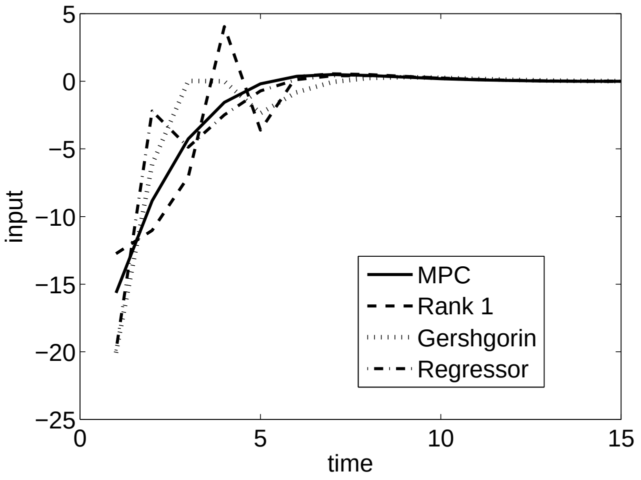


Fig. 2. The control input designed by classical MPC and modifications by all three proposed algorithms for excitation horizon $M = 5$.

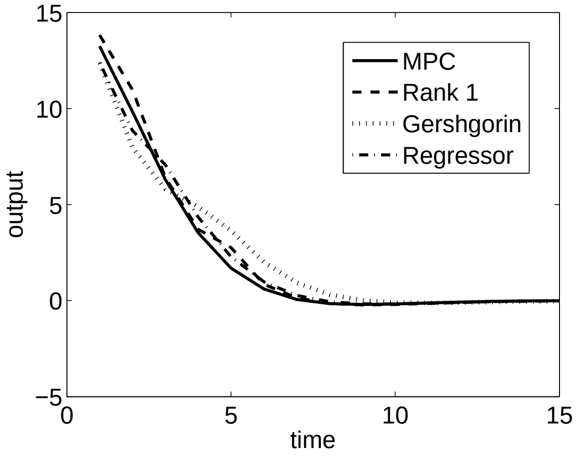


Fig. 3. The output of a system controlled by classical MPC and modifications by all three proposed algorithms for excitation horizon $M = 5$.

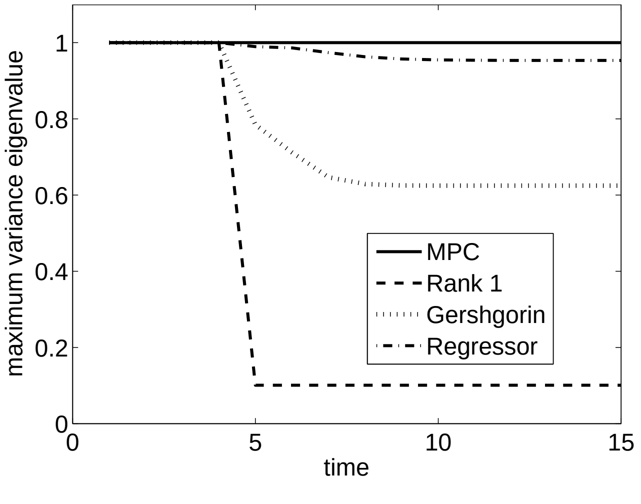


Fig. 4. The maximum eigenvalue of the estimate variance matrix for control designed by classical MPC and modifications by all three proposed algorithms for excitation horizon $M = 5$.

Bernstein, D.S. (2005). *Matrix Mathematics.* Princeton University Press.

Bertsekas, D.P. (2005). *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 3rd edition.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory.* Society for Industrial and Applied Mathematics.

Chen, R. and Loparo, K.A. (1991). Dual control of linear stochastic systems with unknown parameters. In *IEEE International Conference on Systems Engineering*, 65–68. IEEE.

Feldbaum, A.A. (1960–61). Dual control theory I-IV. *Automation and Remote Control*, 21, 22.

Filatov, N.M. and Unbehauen, H. (2004). *Adaptive Dual Control: Theory and Applications.* Springer Verlag.

Filatov, N., Keuchel, U., and Unbehauen, H. (1996). Dual control for an unstable mechanical plant. *IEEE Control Systems Magazine*, 16(4), 31 – 37.

Goodwin, G.C. and Sin, K.S. (1984). *Adaptive Filtering Prediction and Control.* Prentice-Hall, INC.

Havlena, V. (1993). Simultaneous parameter tracking and state estimation in a linear-system. *Automatica*, 29(4), 1041–1052.

Lasserre, J.B. (2000). Convergent lmi relaxations for nonconvex quadratic programs. In *Proceedings of the 39th IEEE Conference on Decision and Control*, 5041 – 5046. IEEE.

Lee, J.M. and Lee, J.H. (2009). An approximate dynamic programming based approach to dual adaptive control. *Journal of Process Control*, 19(5), 859 – 864. doi:DOI: 10.1016/j.jprocont.2008.11.009.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference.* Taipei, Taiwan. URL `http://users.isy.liu.se/johanl/yalmip`.

Lindoff, B., Holst, J., and Wittenmark, B. (1998). Analysis of suboptimal dual control. In *IFAC Workshop on Adaptive Control and Signal Processing*, 20–27. Glasgow, UK.

Lindoff, B., Holst, J., and Wittenmark, B. (1999). Analysis of approximations to dual control. *Int. J. Adaptive Control and Signal Processing*, 13, 593–620.

Orsi, R., Helmke, U., and Moore, J.B. (2006). A newton-like method for solving rank constrained linear matrix inequalities. *Automatica*, 42(11), 1875 – 1882.

Peterka, V. (1986). Control of uncertain processes: Applied theory and algorithms. *Kybernetika*, 22(3–6).

Rawlings, J.B. and Muske, K.R. (1993). The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10), 1512 – 1516.

Sternby, J. (1976). A simple dual control problem with an analytical solution. *IEEE Transactions on Automatic Control*, AC–21, 840–844.

Vandenberghe, L. and Boyd, S. (1996). Semidefinite programming. *SIAM Review*, 38, 49–95.

Wittenmark, B. (2002). Adaptive dual control. In H. Unbehauen (ed.), *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO.* Eolss Publishers, Oxford, UK. Paper 6.43.15.6.

Wittenmark, B. (1995). Adaptive dual control methods: An overview. In *In 5th IFAC symposium on Adaptive Systems in Control and Signal Processing*, 67–72.

On-line Recognition of Autonomous Robot Position via Camera DragonFly

J. Mareš[*], A. Procházka[*], P. Doležel[**],

[*]*Institute of chemical technology, Department of computing and control engineering*
Technická 5, 166 28 Prague 6, Czech Republic
e-mail: jan.mares@vscht.cz,uprt.vscht.cz
[**]*University of Pardubice, Faculty of electrical engineering and informatics, Department of process control,*
Nám. Čs. Legií 565, 532 10 Pardubice, Czech Republic
e-mail: petr.dolezel@upce.cz,www.upce.cz

Abstract: Contribution deals with one possible applications of image processing - recognition of moving object position. Traced object is robot MINDSTORM NXT, which is described in the first part of the paper. The methodology of image processing and object recognition is discussed in the second part of the paper. The third part describes the application example.

1 INTRODUCTION

Signals (in the form of pictures or videos) processing is popular, interesting and widely discussed branch of research which has many applications in praxis:

- transport engineering
- biomedicine data processing
- chemical industry

Video processing is possible to divide into two groups:

- offline - the whole set of images is taken and then processed
- online - each taken image is immediately processed.

Both methods have their advantages and disadvantages. Offline images processing is suitable for astronomical or medical images processing, pollution image processing ect., [Pavelka et all, 2005]. Online image processing is t is rather good for tracking, observing and control.

The contribution describes one of the applications of the second (online) way - online tracking of moving robot.

2 DEVICE DESCRIPTION

2.1 Robots Mindstorm NXT

Robots MINDSTORM NXT are products of LEGO group, more in [www.eduxe.cz] and [Hlinovský et all, 2010]. It is typical LEGO brick box which is equipped by 32-bit microcontroller with three output connectors and four input connectors see Figure 1. Three servomotors are the outputs of the system and inputs are presented by

- ultrasonic sensor,
- sound sensor,
- light sensor
- touch sensor.

Optionally, it is possible to add other sensors as gyroscope, compass, or camera.

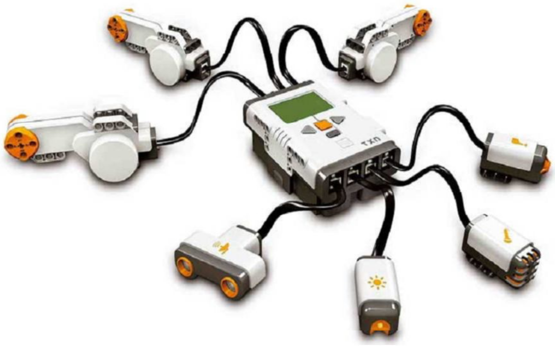


Fig. 1. MINDSTORM microcontroller

Lego MINDSOTRM brick box is possible to set as any variant of mobile robot (see Figure 2). Robots can be absolutely autonomous (PC is used only for their programming) or can be depending on PC (robots provides only measuring, control is provided by the computer). Communication to PC is realized via USB 2.0 or Bluetooth.

All programmes were built in MATLAB environment and MINDSTORM NXT Toolbox was used for communication to robot, [www.mindstorm.rwth-aachen.de]. Toolbox includes basic communication functions:

- initialize robot
- get data from defined sensor

- rotate defined servomotor

Fig. 2. Different variants of robot

2.2 Camera DRAGONFLY

Camera DRAGONFLY is standard digital camera (see Figure 3). It features on-camera color processing and auto white balance, see [Kubíček, 2004].

Specification details are:

- CS-mount lens with variable focus and auto iris
- resolution max 1296x964
- Max 30 frames per second
- IEEE-1394 (Fire Ware) connector.

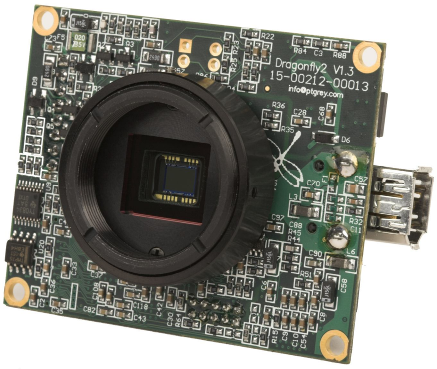


Fig. 3. Camera DRAGONFLY

3 METHODOLOGY OF TRACKING OBJECTS

Generally, object recognition at any background is nontrivial problem. It is necessary to choose the object, find its centre and calculate the position.

If the background has constant colour which is different from the colour of the object, only one object is recognized at the image and one centre position is found.

Otherwise, if the background has a difficult structure and the background colour is similar to object colour, many object are recognized and it is not easy to find out which one is the right one.

One of possible approaches is the frames difference method. Firstly, the image of background (without moving object) is taken. Then (at every sample time) this image is subtracted from the image of moving object. Constant parts of images are black (subtracted each other) and only tracked object remains. Images with and without subtracted background are shown in Figures 4 and 5.

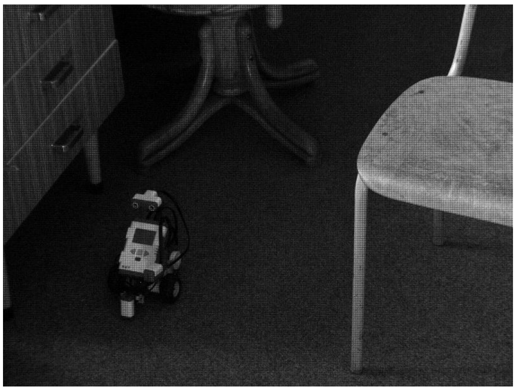


Fig. 4.Object and background with difficult structure

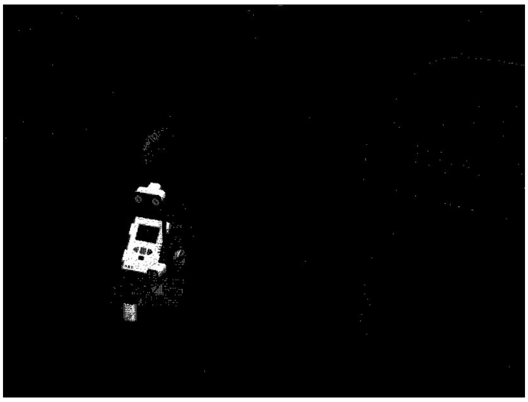


Fig. 5.Image after subtraction of background

After subtraction and the object recognition, it is necessary to find the center and calculate the position (in praxis tresholding and filtration is used too). The object center is possible to find by cutting out the subset bounded by upper, lower, left and right points of tracked object from the image. Object center is given by the center of this new matrix.

For calculation of real object position in 2-D coordinate system, camera calibration is needed. It is necessary to evaluate horizontal and vertical angles of the camera. Using the calibration grid placed in the distance *d* from camera it is possible to find both horizontal *shorizontal* and vertical *svertical* sizes of the figure. These parameters can then be used for evaluation of the limits of angles (using rectangular dark and light triangles).

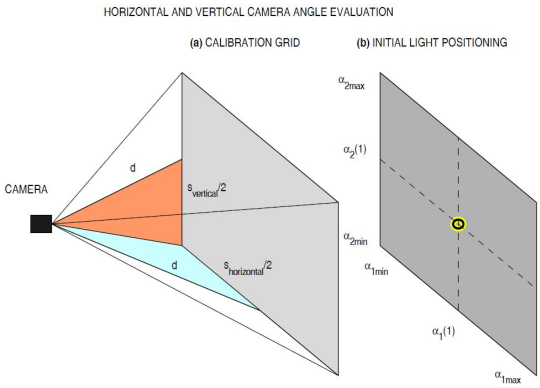


Figure 6 – Camera calibration

$$\alpha_{horizontal} = 2.arctg\left(\frac{s_{horizontal}/2}{d}\right) \tag{1}$$

$$\alpha_{vertical} = 2.arctg\left(\frac{s_{vertical}/2}{d}\right) \tag{2}$$

Finally, it is necessary to calculate calibration straight-lines for horizontal and vertical directions.

Moving object tracking algorithm:

1. Initialize hardware
2. Calibrate the camera
3. Take the first frame of background (without moving object)
4. Take frames of moving object in the cycle
   i. Take the frame

   ii. Calculate the difference

   iii. Find the center

   iv. Calculate the position

4 APPLICATION EXAMPLE

Methodology of moving object tracking was applied to recognition MINDSTORM robot position.

Robot moved randomly in defined space. Its movement was tracked by the camera which was placed above it. Camera setting was one frame per second. Each frame was immediately processed and actual robot position was calculated.

Important parts of MATLAB code are shown in chapter 4.1 and the result- moving object trajectory is plotted in Figure 7.

4.1 *MATLAB code*

Images acquiring and processing

```
obj=videoinput('dcam',1,'Y8_1024x768');
set(obj,'FramesPerTrigger',1);
preview(obj)
start(obj);
[f_pozadi,t]=getdata(obj);

for i=1:1:5
start(obj);
[f_pozice,t]=getdata(obj);
diff=imabsdiff(f_pozadi,f_pozice);
tresh=graythresh(diff);
bw=(diff>=tresh*255);
L=bwlabel(bw);
s=regionprops(L,'area','centroid');
area_vector=[s.Area];
[tmp,idx]=max(area_vector);
Centroid =s(idx(1)).Centroid;
X(i)=centroid(1);
Y(i)=centroid(2);
alfaXh(i)=ah*X(i)+bh;
alfaYv(i)=av*Y(i)+bv;
xposition(i)=alfaXh(i)/alfah*sh*100;
yposition(i)=alfaYv(i)/alfav*sv*100;
end
figure
plot(xposition,yposition,'o-r',)
```

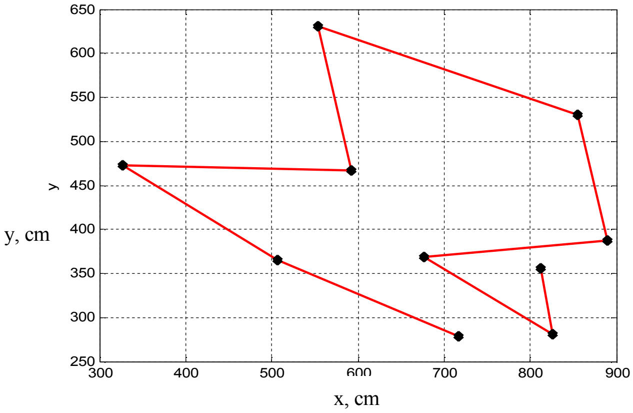


Fig. 7.Moving robot trajectory

5 CONCLUSIONS

Image processing is popular and widely discussed branch of research nowadays.

Contribution deals with one of possible approaches to image processing - moving object tracking (i.e. for control purposes).

One digital camera was used as a sensor of localization of moving robot, thus the trajectory in two-dimensional space was calculated. If two or more cameras were used, 3-D reconstruction could be evaluated.

This methodology of object tracking is usable for one moving object tracking at anystatic background.

ACKNOWLEDGMENTS

The work has been supported by the funds No. MSM 6046137306 and No. MSM 0021627505 of Ministry of Education of the Czech Republic. This support is very gratefully acknowledged.

6 REFERENCES

HLINOVSKÝ, M., KRUCZEK, A. Předmět ROBOTI na ČVUT FEL v Praze. *In Proceedings of Conference ARTEP'10",* 24.-26. 2 2010.Technická univerzita Košice, 2010. ISBN 978-80-553-0347-5.

KUBÍČEK. M. Using Dragonfly IEEE-1394 Digital Camera and Image Acquisition Toolbox. In *Sborník MATLAB 2004*, pages 280–282. VŠCHT Praha, 2004.

LEGO Educational Division v ČR. [cit. 7. 1. 2011]. Available at:

http://www.eduxe.cz/

PAVELKA, A., KUBÍČEK, M., PROCHÁZKA, A. Motion Observation and Modelling in the Virtual Reality Environment. In *Sborník MATLAB 2005*. VŠCHT Praha, 2005.

RWTH Aachen University. [cit. 7. 1. 2011].   Available at:

http://www.mindstorms.rwth-aachen.de

Hough Transform Use in Image Processing of Microscopic Alloy Images

P. Slavíková[*], M. Mudrová[*]A. Michalcová[**]and A. Procházka[*]

[*]*Institute of Chemical Technology Prague, Faculty of Chemical Engineering, Department of Computing and Control Engineering, Technická 6, 166 28 Prague 6*
[**]*Institute of Chemical Technology Prague, Faculty of Chemical Technology, Department of Metal Materials and Corrosion Engineering, Technická 6, 166 28 Prague 6*

Abstract: Presented paper presents principle of Hough transform on fundamental geometric forms and algorithm of its application. Fundamental geometric forms are used to explain core of this method. These statements are expanded on more complex simulated image and critical aspects of using Hough transform are discussed. Tested methods were applied on microscopic images of Al Alloy which were treated by conditions of Vickers indentation test.

1. INTRODUCTION

Field of Hough transform use interferes in many branches of practical applications. Line and shape detection is the very essence of detection of particles in cryo-electron microscopy images (Zhu et al. 2001, Zhu et al. 2003) for example. Aerial images are utilized for semi-automatic and automatic detection of rectangular structures, such as vehicles (Zhao et al. 2001) and buildings (Jaynes et al. 2003).

Most reported shape detection techniques are based on edge and line primitives. Some of these techniques are briefly described in next paragraphs.

Lagunovsky and Ablameyko (1999) based detection algorithm on lines. Lines are extracted first and these segments are grouped in straight lines in the next step. The length and orientation of these straight lines are compared and used to detect quadratangles that are futher approximated by rectangles.

Zhu et al. (2002) used the rectangular Hough transform (RHT) approach to detect rectangular particles in cryo-electron microscopy images. If the sides of rectangle are known, the RHT uses a 2D accumulator array to detect the center and orientation of rectangle. This method produced good results but known rectagle dimensions are required.

In this paper, we explore behaviour of Hough Transform (Duda et al. 1972) and its application to microscopic images of alloy images treated by Vickers indentation test (ČSN ISO 6507-1 1999).

The remainder of this paper is structured as follows. In the next Section, the Hough transform is briefly explaned. In Section 3, an influence of an edge detection on the Hough Transform is described. Section 4 analyse smoothing effects on the Hough Transform and its application to microscopic images is presented in Section 5. Results are concluded in Section 6.

2. HOUGH TRANSFORM

Hough Transform (HT) is processing method designed for line detection in image. Basic principle of HT is finding points which are common to one bisector. Polar coordinates are used for expressing this line. Usage of this method can be expanded to detection of fundamental geometric shapes, like square, rectangle or circle.

Very important is observing a good quality binary image which only can be treated by HT to gain equations of lines in image. This goal requires an application of edge detection methods as necessary pre-processing methods. Image filtering should be considered, as well.

Lines in an image can be represented as a subset of points. By finding bisector parameters from every pair of points, resulting lines can be gained. Because of computational demands of this problem, Hough proposed an alternative method. The very essence of this approach is a conversion of parameters to polar coordinates.

Consider an image with point (x, y). Through this point can be constructed infinity amount of bisectors with general equation (1) where a is a slope and b is a bisector intersection with y-axis. Both of these parameters go toward infinity if the line becomes more vertical. Conversion of this system into polar coordinates solves this problem efficiently. A presented process converts a and b parameters to parameter ρ which represents a distance of bisector from origin and parameter θ which express an angle of bisector with x-axis.

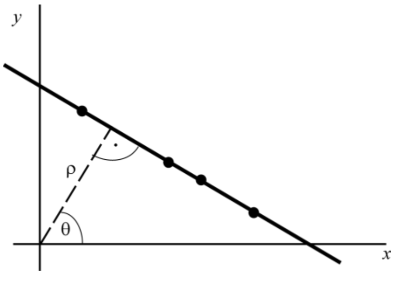


Fig. 1 Distance ρ and angle θ representation of a bisector

According to Fig.1, Eq. (1) can be expressed like Eq. (2).

$$y = ax + b \qquad (1)$$

$$y = -\frac{\cos\theta}{\sin\theta}x + \frac{\rho}{\sin\theta} \qquad (2)$$

$$\rho = x\cos\theta + y\sin\theta \qquad (3)$$

By rearranging of Eq. (2), final Eq. (3) is gained. In contrast of *a* and *b* parameters, the parameter θ has finite domain of definition, θ ϵ <0, π>. Because we handle with image, parameter ρ has also finite domain of definition, ρ ϵ <0, *D*> where *D* is diagonal of the image. The field of ρ and θ parameters is called Hough space.

2.1 Hough Transform of Point

Consider binary image with several white pixels (Fig. 2). This image was treated by Hough transform algorithm, so its Hough space was gained (Fig. 3).

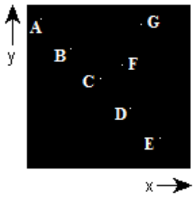


Fig. 2 Test points image

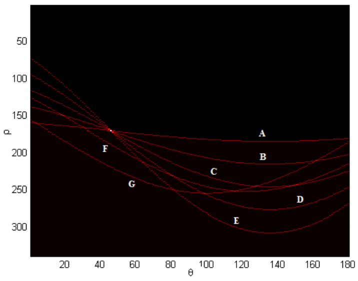


Fig. 3 Hough space of test points image (Fig. 2)

Set of testing point is presented in Fig. 2. It is obvious that points A, B, C, D, E make line and points F and G are out of this line. Hough space of this image (see Fig. 2) is shown in Fig. 3. Curves A – E intersects in one point which represents ρ and θ parameters of Eq. (3). Hough space gives information that points A – E lay on line with same equation, so they lay on same line.

Other intersections are present in this Hough space. It corresponds to other lines which are created by points [A, F], [B, F], [A, G], [B, G] and so on.

2.2 Hough Transform of Circle

Although HT detects just lines, circles in the image can be simply detected, as well. Consider binary image of circle. There are no lines to detect, so in Hough space no intersections are expected (Fig. 4). This property is typical for circle and can be used for its detection.

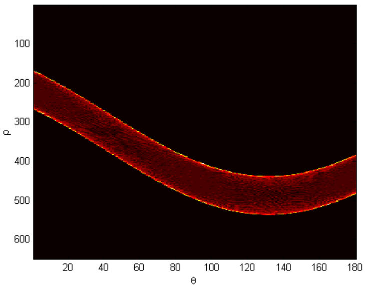


Fig. 4 Hough space of test circle image

2.3 Hough Transform of Square

HT of square also gives typical result. Square is composed from four line segments and four their intersections which lay on escribed circle. So Hough space (Fig. 5) looks like combination of point case and circle case.

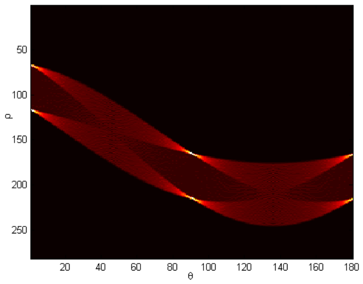


Fig. 5 Hough space of test square image

Curves intersections represent four cusps of square and curves are HT result of side points (Fig.5). Global shape of Hough space curves demonstrate mentioned scripted circle. Similar result can be gained for HT of various tetragons.

3. INFLUENCE OF EDGE DETECTION ON HOUGH TRANSFORM

For processing of real images a proper pre-processing algorithm should be applied. Input data must be converted into intensity image. In the next step edges are detected and at first resulting image will be treated by selected threshold method to gain binary image. Now, HT can be applied.

It is obvious, that edge detection is a critical parameter for successful line detection. Variuos edge detectors can be used - both gradient methods and detection algorithms. Prewitt’s and Laplasian of Gaussian (LoG) methods were selected for representation gradient method group and Canny’s detector was chosen from the second group.

These methods were applied on simulated testing image (Fig. 6) and a quality of edge and line detection was observed.

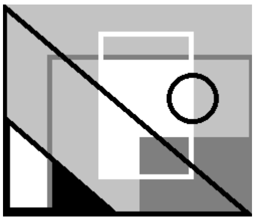


Fig. 6 Simulated image for observing of edge and line detection quality

Result of gradient method group edge detection is shown in Figs. 7, 8, 9 and line detection by HT is presented in Figs. 10, 11, 12.

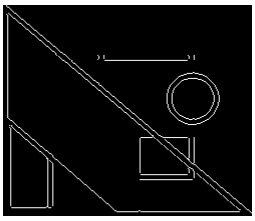


Fig. 7 Prewitt's edge detection of simulated image (Fig. 6)

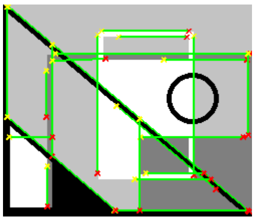


Fig. 10 Line detection of Prewitt's edge detection (Fig. 7)

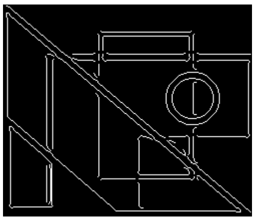


Fig. 8 LoG edge detection of simulated image (Fig. 6)

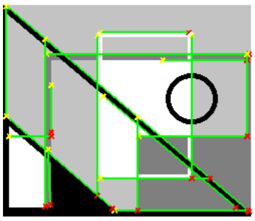


Fig. 11 Line detection of LoG edge detection (Fig. 8)

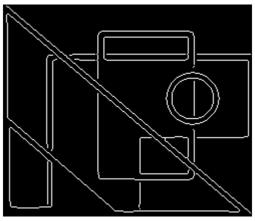


Fig. 9 LoG edge detection of simulated image (Fig. 6)

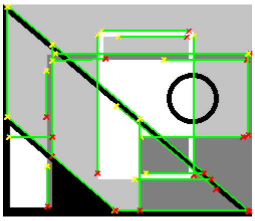


Fig. 12 Line detection of LoG edge detection (Fig. 9)

Presented binary images of edges in simulated image proved that Canny's edge detection algorithm is the best way how to process these data. LoG method was successful, as well, but there are some imperfections in detection of intersections. Prewitt's detector wasn't successful because it didn't recognized less intensive edges. Circle in simulated image wasn't pointed in any images in Figs. 10 - 12. This event has simply interpretation. In used HT algorithm, just lines were detected. For detection of circle, interpretation of Hough space data should be changed. Hough space of Canny edge detector image is shown in Fig. 13, for example.

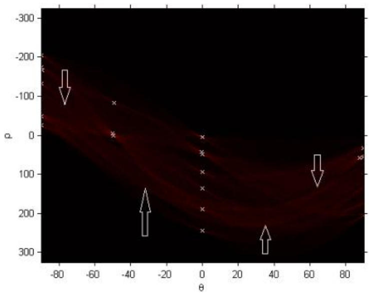


Fig. 13 Hough Space of Canny Edge Detector Image with Marked Feature Typical for Circle

4. SMOOTHING EFFECTS INFLUENCE ON LINE DETECTION QUALITY

In this part, influence of smooth image will be presented. Simulated image (Fig. 6) was smoothed by averaging filter of various sizes and effects on quality of line detection were observed. For edge detection, the most successful Canny's algorithm was used. Results of edge detection in smoothed images are presented in Figs. 14 - 16.

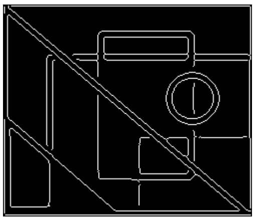


Fig. 14 Edge detection of 5x5 smoothed image

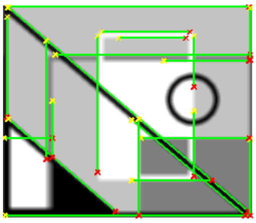


Fig. 17 Line detection of 5x5 smoothed image

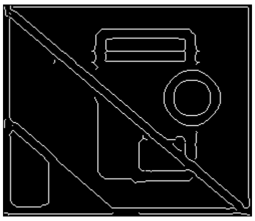


Fig. 15 Edge dection of 10x10 smoothed image

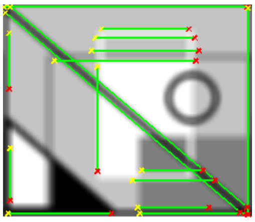


Fig. 18 Line detection of 10x10 smoothed image

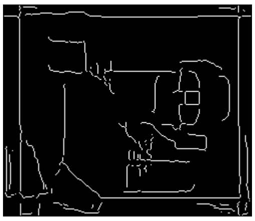


Fig. 16 Edge detection of 30x30 smoothed image

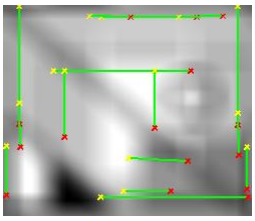


Fig. 19 Line detection of 30x30 smoothed image

Hough Transform was applied to pre-processed images and detected lines are shown in Figs. 17 - 19. These results show that smoothing hasn't direct influence on quality of line detection. Image smoothed by 5x5 averaging filter gave almost the same result as unsmoothed image (Fig. 17). Line detection of 10x10 smoothed image was less successful but almost all edges in Fig. 15 were detected. Smoothing by 30x30 averaging filter failed at line detection but the most of detected edges was detected as in former case. This result demonstrates that a smoothing has no direct effect on line detection but defects edge detection, only.

5. HOUGH TRANSFORM APPLICATION ON MICROSCOPIC ALLOY IMAGES

The final part of presented paper is focused on the application of HT on microscopic images of Al alloy. These images were taken by a scanning electron microscope and display new Al alloy treated by conditions of Vickers indentation test (ČSN ISO 6507-1).

This standardized method is based on an alloy sample treatment by diamond tetragonal pyramid (top angle $\alpha = 136°$) with a known load F to gain standardized puncture. The area of incurred puncture is inverse proportional to an induration of examined alloy. The puncture area is gained by measuring diagonals d_1 and d_2. Vickers induration HV is defined by Eq. (4)

$$HV = \frac{2F}{g}\sin\left(\frac{\alpha}{2}\right)\frac{1}{d^2} = 0.189F\,\frac{1}{d_1 d_2} \qquad (4)$$

where g is a gravity acceleration.

The goal of presented project is automatic evaluation of this puncture and measurement its diagonals by no human affection with help of HT.

Al alloy image was selected for development of processing algorithm. Proper pre-processing was necessary first step. A histogram of input image was evaluated then it was equalized and an image intensity was adjusted. Mathematic morphology methods were applied later. Modified image was dilated by a rectangle structural element, first, and then eroded by a similar element. Edge detection was applied on the resulting image. The last step was line detection by HT (see Figs. 20 – 23).

Troublesome effects of alloy sample preparation (long scratches in Fig. 20) were taken out by morphological methods, so only edges belonging to puncture were detected. Resulting image showed that lines in puncture image were detected successfully.

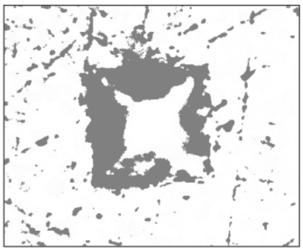


Fig. 20 Al alloy image after histogram equalization and intensity adjustment

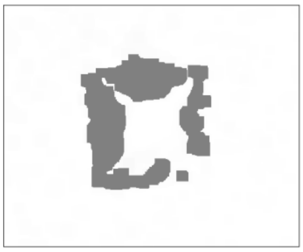


Fig. 21 Al alloy image after mathematical morphology

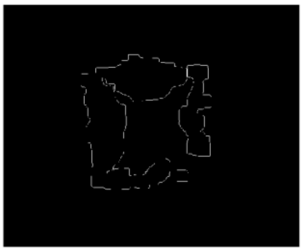


Fig. 22 Edge detection of Al alloy image

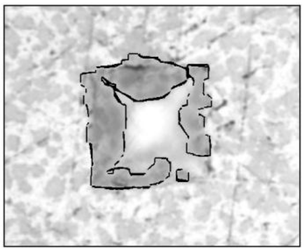


Fig. 23 Line detection of Al alloy image

Presented algorithm was applied on other images of Vickers induration test. A selected result of the algorithm examination is displayed at Fig. 24.

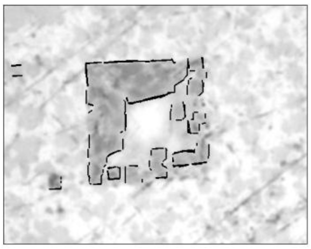


Fig. 24 Line detection of sample image No. 23

6. RESULTS AND CONCLUSION

Presented results showed that Hough transform is powerful tool for detection of shapes in the image. Testing image application proposed robustness from smoothing images, so it was proved that weak step of HT algorithm is efficiency of edge detection.

Algorithm of HT use on Al alloy images was developed and verified on selected set of alloy images. It is necessary to pre-processed input images to avoid influence of troublesome effects. After this, edge and line detection can be implemented successfully.

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Education of the Czech Republic (program No. MSM 6046137306) and Internal Grant Agency ICT (grant No. 445 88 1102). This support is very gratefully acknowledged.

REFERENCES

Ballard, D. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* **13**, 111 – 122.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **8**, 679 – 698.

ČSN ISO 6507-1. (1999). Kovové materály – Zkouška tvrdosti podle Vickerse. *Český normalizační institut.*5 s.

Davidescu, A. (2005). Analysis od HVOF-sprayed MCrAlY Coatings using SEM Image Processing. *Journal of Optoelectronics and Advanced Materials***6**, 3107 - 3110.

Duda, P. and P. Hart (1972). Use of Hough transform to detect lines and curves in pictures. *Comunications of the ACM* **15**, 11 – 15.

Furukawa, Y. and Y. Shinagawa (2003). Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space. *Computer Vision and Image Understanding* **92,** 1 – 25.

Gonzales, R. C and R. E. Woods (2002). *Digital Image Processing*. Second Edition, Prentise-Hall. New Jersey.

Illingworth, J. and J. Kittler (1988). A survey of the Hough transform. *Computer Vision Graphics and Image Processing* **44**, 87 – 116.

Jaynes, C., A. Hanson and R. Riseman (2003). Recognition and reconstruction of buildings from multiple aerial images. *Computer Vision and Image Understanding.* **90**, 68 – 98.

Kiryati, N., Y. Eldar and A. Bruckstein (1991). A propabilistic Hough transform. *Pattern Recognition* **24**, 303-316.

Moon, H., R. Chellapa and A. Rosenfeld(2002). Performance analysis of a simple vehicle detection algorithm. *Image and Vision Computing* **20,** 1 – 13.

Lagunovsky, D. and S. Ablameyko (1999). Straight-line-based primitive extraction in gray-scale object recognition. *Pattern Recognition Letters* **20**, 1055 – 1014.

Leavers, V. (1993). Survey: Which Hough transform. *Computer Vision Graphics and Image Processing* **58**, 250 – 264.

Lu, J. andQ. Ruan (2006). Aluminium Alloy X-ray Classification Using Texture Analysis. 8th International Conference on Signal Processing, 16th – 20th November 2006, Beijing (2006).

Wang, H. and F. Shen (2002). Corner detection based on modified Hough transform. *Pattern Recognition Letters***23**, 1039 – 1049.

Weiss, L. and A. Rosenfeld (1995). A convex polygon is determinate by its Hough transform. *Pattern Recognition Letters* **16**, 305 – 306.

Zhao, T. and R.Nevatia (2001). Car detection in low resolution aerial images. In: *International Conference on Image Processing*, 710 – 717.

Zhu, Y., F. Carragher, F. Mouche and C. Potter (2003). Automatic particle detectionthrough efficient Hough transforms. *IEEE Transaction on Medical Imaging* **22**, 1053 – 1062.

Zhu, Y., B. Carragher, D.Kriegman, Milligan, R.A. and C. Potter (2001). Automated identification of filaments in cryo-elecron microscopy images. *Journal of Structural Biology***135**, 302 - 312.

Zhu, Y., B. Carragher andC. Potter (2002). Fast detžection of generic biological particles in cryo-en images through efficient Hough transforms. In:*International Symposium for Biomedical Imaging*, 205 – 208.

Software background of the ’Mechatronics-based rehabilitation at home’ concept

J. Sobota, O. Severa, M. Kocánek, M. Čech, P. Balda

University of West Bohemia in Pilsen
Univerzitní 8, 306 14 Plzeň, Czech Republic
[jsobota,osevera,mkocanek,mcech,pbalda]@kky.zcu.cz

Abstract: In this paper the concept of ’Mechatronics-based Rehabilitation at Home’ is presented. The idea is to transfer as much rehabilitation care as possible from hospitals and rehabilitation centres to patients’ homes while keeping the effectiveness and quality of rehabilitation process as we know it today. Such concept naturally heavily relies on technological innovation, especially on control systems and communication technologies. Only then is it possible that the patients perform the rehabilitation exercises remotely and autonomously and the rehabilitation specialists and physiotherapists still have enough information about the progress in the patient to adjust or change the treatment as necessary. Software tools addressing the challenging requirements are presented.

Keywords: rehabilitation, remote visualization, 3D HMI, SFC editor, encryption

1. INTRODUCTION

The largest challenge in rehabilitation in the following years will be to increase (or at least keep) the quality, access and efficacy of care. With respect to the increasing numbers of elderly people and growing importance of chronic welfare diseases, it is obvious that the concept of rehabilitation as we know it today is unsustainable.

The rehabilitation therapy is nowadays offered by therapists in healthcare centres or at the patient’s home, usually on one-on-one basis. This approach has several disadvantages, especially when the therapy lasts over years:

- Traveling to/from a healthcare centre is time consuming and expensive.
- Patient’s daily program is affected (sometimes even controlled) by the schedule of rehabilitation sessions.
- Although a number of rehabilitation aids and equipment is available, which allows autonomous execution of specific exercises, the major part of the therapy is performed by the physiotherapist. This is physically demanding and expensive due to labor costs.

With the ageing population, all the mentioned problems will intensify, as the number of available therapists will relatively decrease.

There is a clear understanding that better healthcare and quality of rehabilitation can be achieved through technological innovation. Nowadays there are various motorized devices available, ranging from simple continuous passive motion devices to more complex systems aimed at neurological patients. But it is common to all the available devices that they are aimed at the use in rehabilitation centres, where a lot of patients use the same machine supervised by a physiotherapist.

According to the foreseen consequences of the ageing population, especially the growing importance of rehabilitation care and its availability, attempts are made to move a significant part of the rehabilitation treatment from hospitals and rehabilitation centres to the patient’s home and to minimize the need of physiotherapist’s presence during the treatment. In order to make this happen, the rehabilitation devices must be ready for home use.

2. GENERAL STRUCTURE OF A HOME REHABILITATION DEVICE

The keyword for this transition is mechatronics, the devices have to combine mechanical design with electronic components and IT technologies. A general stucture of the ’Mechatronics-based Rehabilitation at Home’ (MRH) concept is depicted in Fig 1. The concept is divided into three main parts, which can be further decomposed into several building blocks (or subsystems).

The essential part is the MRH device itself, which is located at the patient’s home and is accessible through the Internet. The patient performs a set of predefined exercises, which are controlled and monitored by a control system through appropriate actuators and sensors. The progress of the exercises is displayed by the Human-Machine-Interface (HMI) and the patient is thus informed about how well he/she is doing, which helps to keep the patient focused and motivated. The individual training sessions are recorded for evaluation of the rehabilitation treatment.

The counterpart to the MRH device is the supervision software used by the rehabilitation specialist or physiotherapist. The parameters of individual exercises can be adjusted, which includes the general type of therapy (con-

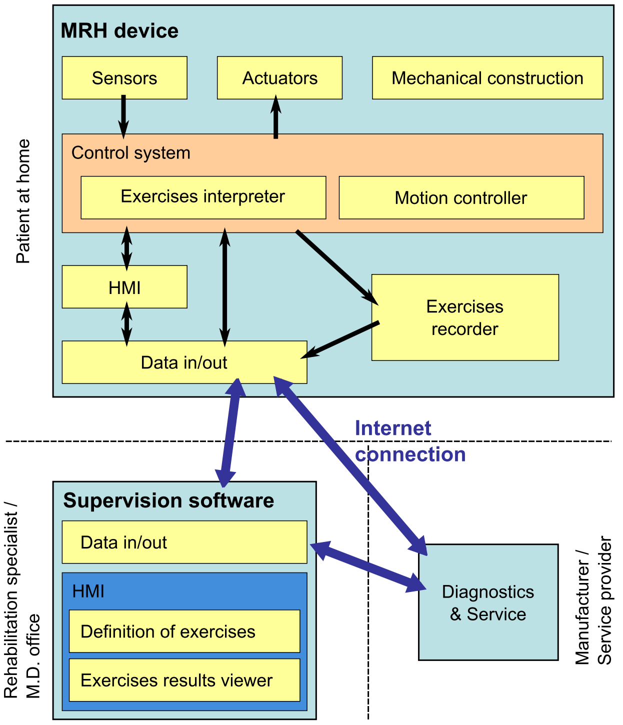


Fig. 1. Building blocks of the 'Mechatronics-based Rehabilitation at Home' concept

tinuous passive motion, active therapy against resistive forces, etc.) as well as more specific parameters like speed of movements, number of repetitions, ranges of motion in individual joints or the resistive forces.

Both these main parts can communicate with the manufacturer or service provider to allow remote diagnostics and service (wear detection, firmware upgrade, etc.)

A specific subsystem is the mechanical design of the MRH device. The construction and mechanical setup is naturally the crucial part of the whole system, it must be easy-to-use for the patient. Even patients with lower motor abilities must be able to position and attach their extremities to the device so that the rehabilitation exercise can be performed safely and efficiently, the risk of performing a non-physiologic movements or even injury due to improper fixation to the device must be minimized. Although this part of MRH device design and development is a demanding and challenging task, it is not addressed in this paper as it is focused on the software part of the MRH concept.

3. SOFTWARE ELEMENTS

As was mentioned above, the software incorporated into the MRH concept plays a significant role. The control system included in the MRH device must not only gather data from sensors and control the actuators, but it must also allow visualization of the data and support communication for remote access over the Internet.

3.1 REX control system

A suitable platform for the MRH device is the REX control system (Balda et al., 2005). The key features of this system include:

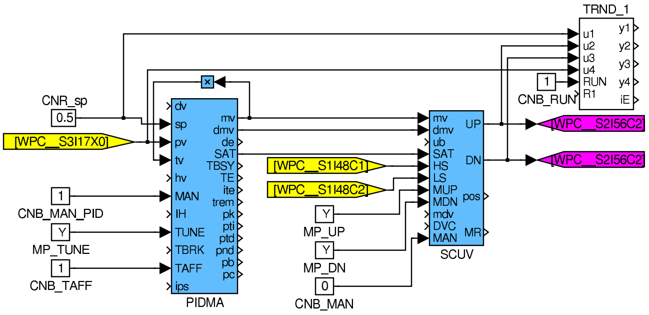


Fig. 2. Example of REX system programming - a PID position control loop for stepper motor

- Control algorithms consist of function blocks from the REX control system library - both standard and advanced control algorithms are included
- Drag and drop programming, simple modification of control algorithms
- Portability to various platforms - various target devices and operating systems ranging from Industrial PCs to microchips
- Compatibility with Matlab/Simulink - the same configuration can be used for simulation in Matlab/Simulink and for the final deployment of the real-time control algorithm (Fig. 2), but no xPC target nor Real-Time Workshop is required
- The control algorithm running on the target platform is accessible through Internet
- Wide range of communication protocols - Modbus, OPC, Ethernet PowerLink, EtherCAT, CAN
- Sophisticated archiving subsystem for storing and retrieving data

All the mentioned properties form a universal platform fulfilling the requirements on software design and development for mechatronics-based rehabilitation devices for home use. The REX control system is well equipped for data acquisition from sensors, processing and storing the data and generating physical motion through the actuators. Majority of the features was described earlier in detail and thus the emphasis is put on new features which were introduced in version 2.0 of the REX control system and have a significant importance for the MRH field.

3.2 HMI development in Java

Human-Machine-Interface is an integral part of any control application. Traditionally, it is composed of several screens displaying current values of important variables from a machine or process. Initially, the screens were quite simple and based on 2-dimensional (2D) graphics. Fifteen years ago, a boom of 3-dimensional (3D) scenes started thanks to the growing power and capabilities of computers (in particular graphics cards and accelerators). Such visualization provides more realistic view of the object existing in real world. The number of elements in the scene is basically not limited, thus one can create a full 3D model of even very complex scene. If the visualization platform together with the data acquisition subsystems satisfy real-time requirements then one gets at each moment an up to date live view of the reality. Let us call it further RT-3D-HMI (real-time). This live view is valuable for remote supervision

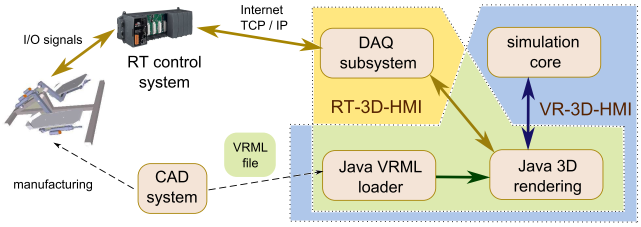


Fig. 3. Scheme of the 3D HMI

of the rehabilitation exercises, the physiotherapist has all information about the exercise in the most natural form as if the patient was performing the exercises in front of him/her.

On the other hand, the 3D-HMI may be supplied by simulation data from a model containing a model of both the MRH device and its control system. Then one gets a fully functioning virtual reality model of the device. Such approach has been proven useful for distance learning (Le Blanc et al., 2005) or training new human operators in industrial practice (Pei et al., 2009). Let us denote such tools as VR-3D-HMI (virtual-reality). The terminology is clearly explained in Fig. 3.

Nowadays, the 3D-modules are available in the majority of leading visualization software (e.g. Indusoft, LabView). Unfortunately, the utilization of mentioned large software packages has several limits. They usually do not allow exporting 3D screens into self-contained Java applets which can be further embedded into a platform-independent web page. Consequently, such HMIs are often executable only on classical desktop computer equipped with Microsoft Windows. Further, the HMI integration with the machine and control system model is quite complicated. Finally, such software is too expensive for simple or embedded applications.

Java platform was already proved suitable for developing interactive web tools [9]. It completely obviates the above mentioned problems. The 3D HMI is based on Java3D package (Oracle, 2011) (rendering package) which is often used to display 3D scenes in Java (Wu and Miao, 2008).

One has always to take two basic parts into account when creating the 3D model. The first part is the 3D rendering engine which takes care of drawing model on the screen. The second part is some SW tool for 3D model creation - usually CAD system (SolidWorks, Autodesk Inventor, CATIA, etc). Note that the CAD system is always used when designing and manufacturing new devices. Hence, using it as a source of model for 3D visualization is quite natural. However, there must exist a 'bridge' between these two parts. In other words, the CAD system must provide export of the 3D model into a format acceptable by the rendering engine.

3D graphics formats Generally, both RT-3D HMI and VR-3D HMI use the same 3D model created by CAD software. For Java platform, it was decided that the VRML (Virtual Reality Modeling Language) will be the final input format for Java 3D renderer. It is well supported and very widely used 3D format. Nowadays it has a successor called X3D which is backward compatible. Although VRML or X3D formats are broadly used on the World Wide Web, only a minority of the 3D CAD software support these formats in default (e.g. SolidWorks). The standard data exchange format supported by almost all CAD systems is STEP. Basically, all of the mentioned formats have a capability to preserve the objects in a natural tree structure which is essential for animation of the individual parts. Let us describe the key features of the formats in more detail.

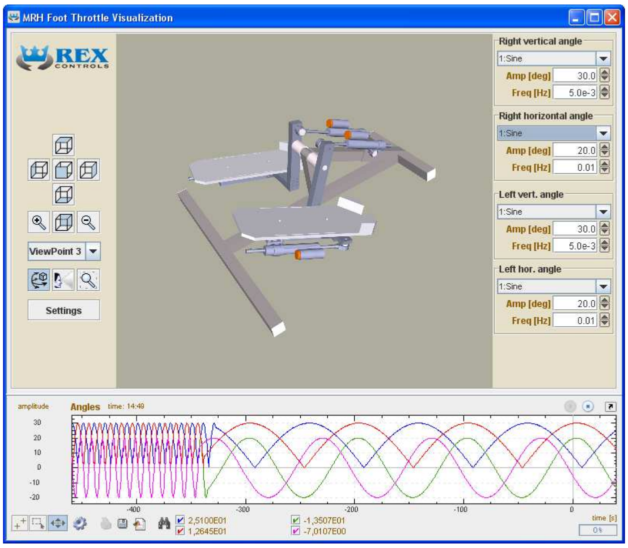
VRML Virtual Reality Modeling (Markup) Language is a standard text file format for representing 3-dimensional (3D) interactive vector graphics, designed particularly with the World Wide Web in mind. It is standardized by ISO (ISO/IEC 14772-1:1997) and maintained by Web3D Consortium. Common extensions are *.wrl or *.wrz for gzip-compressed VRML files.

X3D is a successor of VRML. It is standardized by ISO (ISO/IEC 19775/19776/19777) and also maintained by Web3D Consortium. X3D scene can be stored in a text XML file (*.x3d), Open Inventor-like syntax of VRML97 (*.x3dv) or in binary file (*.x3db). X3D specification support several profiles. These profiles define various levels of capabilities. The big advantage of XML structure is that there exist data parsers (SAX, DOM) in almost all leading programming languages.

Step-file standardized under ISO 10303-21 is the most widely used data exchange form of STEP (Standard for the Exchange of Product model data). This ASCII text file represents data according to a STEP Application Protocols. Step-files are well supported by various 3D CAD software packages. If there is no direct export into VRML available, such format may serve as a bridge between some 3D CAD and VRML. However, one must find a way to convert STEP files into VRML outside the CAD system. One possibility is to use CADExchanger (Lygin, 2010) as a convertor from STEP-File into X3D and then use Flux Studio (Mediamachines, 2007) for further modifications and final export to VRML.

Java 3D package and Java VRML loader Java3D is a set of APIs which is the Java expansion in the three-dimensional field based on OpenGL and D3D. As was mentioned earlier the CAD software provides the 3D model and Java3D can render any 3D scene. Anyways, there is need for a bridge between these two technologies. Java3D VRML Loader is a part of Java3D project. By a few lines of Java code one can parse VRML files and create a tree which represents the whole scene. This tree structure is passed to the Java3D renderer which takes care of the periodical repainting of the scene.

Foot throttle example Fig. 4 shows an example of the resulting RT-3D-HMI for a foot throttle, a rehabilitation machine developed at Fontys University of Applied Sciences for exercises focused on lower extremities, specifically the ankles and knees. The application is divided to four major parts. There is a 3D control panel on the left side followed by the main 3D canvas and the user control panel on the right side. The physiotherapist can control orientation, zoom, view and setting of the 3D canvas through 3D control panel to get a clear overview of the device and position of the individual parts. The user control panel suits remote supervision needs. It is



MRH Foot Throttle Visualization

Right vertical angle: 1:Sine
Amp [deg] 30.0
Freq [Hz] 5.0e-3

Right horizontal angle: 1:Sine
Amp [deg] 20.0
Freq [Hz] 0.01

Left vert. angle: 1:Sine
Amp [deg] 30.0
Freq [Hz] 5.0e-3

Left hor. angle: 1:Sine
Amp [deg] 20.0
Freq [Hz] 0.01

☑ 2,5100E01
☑ 1,2645E01
☑ -1,3507E01
☑ -7,0107E00

Fig. 4. HMI for foot throttle rehabilitation device

possible to send commands to the control algorithm, read values and potential error messages. More specifically, the user can control the speed of individual rotations as well as the limit angles based on the patient's abilities. Last noticeable component is a trend panel at the bottom of the application. This trend shows the positions of individual joints on a timescale. A variation of the HMI is prepared for the patient, illustrating the exercise being executed as well as the number of remaining repetitions, estimate of the remaining time, etc. Thus at each moment the patient exactly knows what is happening and what will follow, which greatly improves his/her focus and dedication to the training.

The 3D-HMI entirely depends on data acquired from RT control system, i.e. the visualization exactly reflects the state of reality. Hence, the live connection to the process is necessary (DAQ subsystem in Fig. 3). The connection to the real time device controlling the foot throttle device is ensured by the JavaREX communication protocol (Balda and Čech, 2006). The developed HMI is able to run as an application or applet, so it is possible to embed it into a webserver within the MRH device, which allows remote connection to the device through any web browser with basic Java Runtime Environment.

3.3 Sequential logic programming

Sequential logic is usually a significant part of control algorithms. The MRH applications are no exception. The rehabilitation exercises consist of many individual motion sequences which needs to be switched based on numerous conditions. Plus there is additional logic handling possible error events and states. Even control algorithms of the regulatory type are very often equipped with a sequential logic e.g. for switching of control modes and strategies.

Sequential logic in general can be inserted into control algorithms by several ways. Very simple logic can be programmed in some textual programming language (C, C++, structured text, etc.) directly. As the size of the logic increases, which is the case of MRH applications, the code maintenance becomes more and more complicated and this approach is very error-prone. Hence, several graphical techniques for sequential logic design have been developed. The sequential function chart (SFC) technique (formerly also called Grafcet) is one of them. This formalism is widely accepted by the automation community (vendors, users). Moreover, it is standardized by the IEC 61163-3 standard (IEC, 2003), which describes five languages for programmable logic controllers (PLC) algorithm development.

For MRH applications the need for a graphical sequential function chart editor, which could be integrated into the REX control system development environment RexDraw (Balda et al., 2005) is obvious. The effort to find some open source standalone and smart SFC editor failed. Of course, there are many implementations of SFC editor integrated in commercial PLC development tools but they could not be used for this purpose. This fact initiated the development of an editor which allows the user not only to create various SFC schemes and store them in XML files but also to check their syntax (validate them), compile them for the selected target and to visualize (monitor) the state development of the particular scheme in real-time. The editor is written in C# using Microsoft .NET framework.

Editor architecture Almost overall logic of the SFC Editor can be divided to the three basic categories. The first category contains classes related to the objects which are visible in the graphical user interface (GUI) and should be painted to the canvas. The second category contains classes responsible for handling a chart validation and compilation processes. The last category contains classes implementing the monitoring functionality for algorithm state evolution of the particular chart.

Graphical user interface As mentioned above, the editor is based on Microsoft .NET technology and it is coded in C# language. That is why the user interface consists of familiar windows controls.

Graphical user interface (GUI) of the SFC editor depends on command line arguments passed to the SFC editor. If the editor is called with a parameter determining one of the target devices then it acts as a single document interface (SDI) application shown in Fig. 5. If it is launched as a standalone application (without command line parameters) it acts like multiple document interface (MDI) application.

The standard menus and toolbars provide options for creating new SFC charts, closing, opening, exporting, and printing existing SFC charts. Zooming command (zoom in, zoom out) of the chart view are located in the toolbar at the bottom of the application window.

SFC Editor employs a drag and drop method for creating and editing SFC charts in graphical user interface (inspired by Microsoft Visio or Matlab Simulink editors). Thus, a user can place and order the blocks in the chart arbitrarily.

All the SFC graphical formalism blocks (and some special blocks) are located in the dock panel named ToolBox on the right edge of the editor. A user can easily insert SFC blocks to the chart by simple dragging of the selected block to the designated place in the chart.

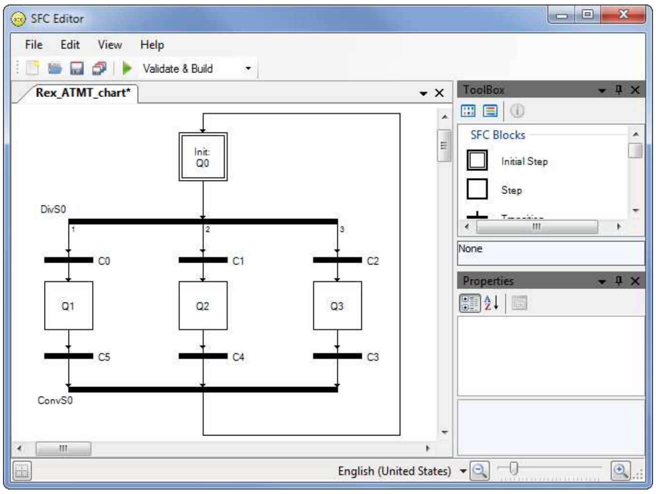


Fig. 5. SFC editor in single document interface mode

SFC block properties can be edited in the Property panel, located by default in the right bottom side of the editor window. The property panel uses smart PropertyGrid component for viewing and editing all public block's properties.

All detected mistakes in the SFC chart and their descriptions are shown in the error list. Similarly, results of the successfully built chart are displayed in an output panel located by default in the left side of the SFC editor.

In addition to such a standard functionality as the printing command of the current chart, which uses a special printing dialog, the SFC editor offers exporting of the chart to the scalable vector graphics (SVG) format.

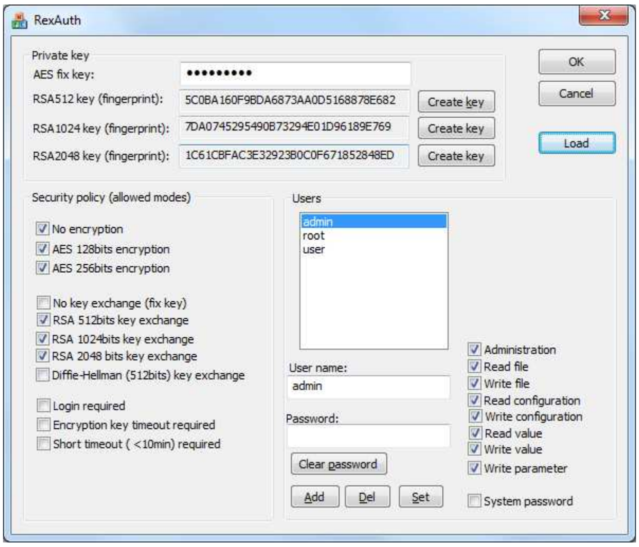
GUI of the SFC editor is mostly inspired by the environment of the Microsoft Visual Studio. The last running mode of the SFC editor, in which the GUI is modified, is the monitoring mode (SFC editor runs as a diagnostic tool). In this case, the additional toolbar with special monitoring controls (Start, Stop, etc.) appears.

3.4 User authentication and encryption

Another important aspect of the MRH device is data security. Because the data stored in the device and transmitted over the Internet are of medical character, it must be considered confidential. Therefore the MRH device must be inaccessible by unauthorized users (hackers) and the transmitted data must be encrypted.

If no security was used, the device would be accessible practically for everybody. The diagnostic program RexView is available for download on the Internet and thus the only necessary information to gain access to the device would be its IP address. The new version of REX control system supports username- and password-based access restriction. Moreover, it is possible to specify rights and privileges for each user.

Further, the communication must be encrypted, otherwise it would be possible to read the data by unauthorized users. To avoid TCP/IP interception, the communication is encrypted using the AES algorithm.


RexAuth

Private key
AES fix key: ••••••••••
RSA512 key (fingerprint): 5C0BA160F9BDA6873AA0D5168878E682
RSA1024 key (fingerprint): 7DA0745295490B73294E01D96189E769
RSA2048 key (fingerprint): 1C61CBFAC3E32923B0C0F671852848ED

Security policy (allowed modes)
☑ No encryption
☑ AES 128bits encryption
☑ AES 256bits encryption

☐ No key exchange (fix key)
☑ RSA 512bits key exchange
☑ RSA 1024bits key exchange
☑ RSA 2048 bits key exchange
☐ Diffie-Hellman (512bits) key exchange

☐ Login required
☐ Encryption key timeout required
☐ Short timeout (<10min) required

Users
admin
root
user

User name: admin
Password:

☑ Administration
☑ Read file
☑ Write file
☑ Read configuration
☑ Write configuration
☑ Read value
☑ Write value
☑ Write parameter

☐ System password

Fig. 6. RexAuth - configuration of security subsystem in the REX control system

Security subsystem in the REX control system The security measures of the REX control system are configured in the RexAuth program as illustrated in Fig. 6. It is possible to allow more encryption levels, the available options are:

- `No encryption` - Data communication is not encrypted.
- `AES 128-bit encryption` - Data is encrypted by AES (Rijndael) algorithm with 128-bit key
- `AES 256-bit encryption` - Data is encrypted by AES (Rijndael) algorithm with 256-bit key

The type of encryption is proposed by the client (supervision software) at the physiotherapist's computer and the server (MRH device) accepts or declines it according to security settings. If encryption is enabled, it is necessary to define how the decryption key will be transferred:

- `No key exchange` - A fixed AES key is used; must be specified in the Private key field.
- `RSA 512-bit key exchange` - The client generates the key for the AES encryption and sends it to the server encrypted by the RSA algorithm (the key for RSA algorithm is 512 bits long). The public key of the server is used for RSA encryption.
- `RSA 1024-bit key exchange` The same as above, only the RSA key is 1024 bits long).
- `RSA 2048-bit key exchange` The same as above, only the RSA key is 2048 bits long).
- `Diffie-Hellman key exchange` A widespread approach where the random key generation is distributed among the client and the server. There is no need to generate the private key in advance.

Configuration of users and their rights is quite straightforward, each user has its own privileges:

- `Administration` - Defines the right to perform special operations like pausing and resuming the control algorithm, performing diagnostic tasks, etc.
- `Read file` - Allows reading of files from the target device.
- `Write file` - Allows writing files to the target device.

- `Read configuration` - Allows reading the whole configuration file of the control system.
- `Write configuration` - Allows writing of the whole configuration file of the control system, i.e. completely change the control algorithm.
- `Read value` - Allows reading of individual signals and parameters in the control algorithm.
- `Write value` - Allows writing of individual signals of the control algorithm (setting them to constant values).
- `Write parameter` - Allows writing of individual parameters of the control algorithm. This right includes the `Write value` right automatically.

4. CONCLUSION

In this paper the concept of 'Mechatronics-based Rehabilitation at Home' was presented. The idea to transfer as much rehabilitation care as possible from hospitals and rehabilitation centres to patients' homes can be put into reality if modern technology is used. A key role in this concept play the communication and control systems. A control system suitable for home-rehabilitation under remote supervision was presented, especially the 3D visualization, graphical programming and communication security aspects were highlighted.

5. ABOUT THE MRH PROJECT

The 'Mechatronics-based Rehabilitation at Home' is a joint initiative of 5 regions of the European Union under the 'Innovation for Welfare' framework (I4W, 2011). A collaborative platform has been created to exchange experience and expertise. The project members are:

- Fontys University of Applied Sciences (Eindhoven, the Netherlands)
- Technical University of Catalonia (Barcelona, Spain)
- University of Brescia (Brescia, Italy)
- Upper Austria University of Applied Sciences (Linz, Austria)
- University of West Bohemia in Pilsen (Pilsen, Czech Republic)

More information can be found at the project website (MRH, 2011).

6. ACKNOWLEDGEMENTS

This paper was supported from the funds of the European Union (ERDF/INTERREG IVC) through the 'Innovation for Welfare' project (subproject I4W-22-MRH). This support is gratefully acknowledged as well as the reviewers' comments.

REFERENCES

Balda, P., Schlegel, M., and Štětina, M. (2005). Advanced control algorithms + Simulink compatibility + Real-time OS = REX. In *Preprints of the 16th IFAC World Congress*. Prague, Czech Republic.

Balda, P. and Čech, M. (2006). Java interface to REX control system. In *Proceedings of Process control 2006*, 1–8. Pardubice.

I4W (2011). Innovation for Welfare, project webpage. www.innovation4welfare.eu.

IEC (2003). IEC 61131-3, Edition 2.0, Programmable Controllers - Part 3: Programming Languages.

Le Blanc, A., Bunt, J., Petch, J., and Kwok, Y. (2005). The virtual learning space - an interactive 3D environment. In *Proceedings of 10th International Conference on 3D Web Technology*, 93–102. Bangor, Wales.

Lygin, R. (2010). CAD Exchanger. http://www.cadexchanger.com/.

Mediamachines (2007). Mediamachines web blog. http://mediamachines.wordpress.com/flux-player-and-flux-studio/.

MRH (2011). Mechatronics-based Rehabilitation at Home, project webpage. www.mrh-project.eu.

Oracle (2011). Project Java3D. https://java3d.dev.java.net/.

Pei, Y., Wu, H., and Zong, R. (2009). Simulated joint robot-hand system based on Java3D. In *Proceedings of the 2009 2nd international congress on image and signal processing*, volume 1-9, 2805–2808. IEEE.

Wu, Y. and Miao, Z. (2008). Construction of the virtual environment based on Java3D. In *Proceedings of IET 2nd International Conference on Wireless, Mobile and Multimedia Networks, ICWMMN 2008*, 512–515. Beijing.

State Feedback Control Design Using Eigenstructure Decoupling

R. Fónod and P. Kocsis

Department of Cybernetics and Artificial Intelligence
Faculty of Electrical Engineering and Informatics
Technical University of Košice, Košice, Slovakia
fax: ++ 421 55 625 3574
e-mail: robert.fonod@student.tuke.sk; pavol.kocsis@tuke.sk

Abstract: In this paper the design of controlling a class of linear systems via state feedback eigenstructure assignment is investigated. The design aim is to synthesize a state feedback control law such that for prescribed eigenvalues of the closed-loop control system corresponding eigenvectors are as close to decoupled ones as possible. The set of parametric vectors and the set of closed-loop eigenvalues represent the degrees of freedom existing in the control design, and can be further properly chosen to meet some desired specification requirement, such as mode decoupling and robustness. An illustrative example and the simulation results show that the proposed parametric method is effective and simple.

Keywords: Mode decoupling, singular value decomposition, state feedback, linear control systems, eigenstructure assignment.

1. INTRODUCTION

The static and the dynamic pole placement belongs to the prominent design problems of modern control theory, and, although its practical usefulness has been continuously in dispute, it is one of the most intensively investigated in control system design. It seems that the state-feedback pole assignment in control system design is one from the preferred techniques. In the single-input case the solution to this problem, when it exists, is unique. In the multi-input multi output (MIMO) case various solutions may exist (Filasová (1999), Ipsen (2009)), and to determine a specific solution additional conditions have to be supplied in order to eliminate the extra degrees of freedom in design strategy.

In last significant progress has been achieved in this field, coming in its formulation closest to the algebraic geometric nature of the pole placement problem (Kautsky et al. (1985), Wonham (1985)). The reason for the discrepancy in opinions about the conditioning of the pole assignment problem is that one has to distinguish among three aspects of the pole placement problem, the computation of the memoryless feedback control law matrix gain, the computation of the closed loop system matrix eigenvalues spectrum and the suppressing of the cross-coupling effect (Wang (2003)), where one manipulated input variable cause change in more outputs variables .

Thus, eigenstructure assignment seems to be a powerful technique concerned with the placing of eigenvalues and their associated eigenvectors via feedback control laws, to meet closed-loop design specifications. The eigenvalues are the principal factors that govern the stability and the rates of decay or rise of the system dynamic response. The right and left eigenvectors, on the other hand, are dual factors that together determine the relative shape of the system dynamic response (Kocsis and Krokavec (2008), Sobel and Lallman (1989)).

The general problem of assigning the system matrix eigenstructure using the state feedback control is considered in this paper. Based on the classic algebraic methods (Golub and Van Loan (1989), Datta (2004), Poznyak (2008)), as well as on the algorithms for pole assignment using Singular Value Decomposition (SVD) (Filasová (1997), Krokavec and A. Filasová (2006)) the exposition of the pole eigenstructure assignment problem is generalized here to handle the specified structure of the left eigenvector set in state feedback control design for MIMO linear systems. Extra freedom, which makes dependent the closed-loop eigenvalues spectrum, is used for closed-loop state variables mode decoupling.

The integrated procedure provides a straightforward methodology usable in linear control system design techniques when the memory-free controller in the state-space control structures takes the standard form. Presented application for closed-loop state variables mode decoupling is relative simple and its worth can help to disclose the continuity between eigenstructure assignment and system variable dominant dynamic specification.

2. PROBLEM STATEMENT

Linear dynamic systems with n degree of freedom can be modelled by the state-space equations

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) \qquad (1)$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{q}(t) \qquad (2)$$

with constant matrices $\boldsymbol{A} \in I\!R^{n\times n}$, $\boldsymbol{B} \in I\!R^{n\times r}$, and $\boldsymbol{C} \in rR^{m\times n}$. Generally, to the controllable time-invariant linear MIMO system (1) a linear state feedback regulator control law, defined generally as

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{q}(t) + \boldsymbol{L}\boldsymbol{w}(t) \tag{3}$$

with $\boldsymbol{K} \in I\!R^{r\times n}$, $\boldsymbol{L} \in I\!R^{r\times m}$ gives rise to the closed-loop system

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}_c\boldsymbol{q}(t)+\boldsymbol{B}\boldsymbol{L}\boldsymbol{w}(t) \tag{4}$$

which closed loop poles are eigenvalues of matrix $\boldsymbol{A}_c = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})$ and $\boldsymbol{A}_c \in I\!R^{n\times n}$.

Throughout the paper it is assumed the pair $(\boldsymbol{A}, \boldsymbol{B})$ is controllable.

3. BASIS PRELIMINARIES

3.1 Orthogonal Complement

Definition 1. (Null space) Let $\boldsymbol{E}$, $\boldsymbol{E} \in I\!R^{h\times h}$, rank$(\boldsymbol{E}) = k < h$ be a rank deficient matrix. Then the null space $\mathcal{N}_{\boldsymbol{E}}$ of $\boldsymbol{E}$ is the orthogonal complement of the row space of $\boldsymbol{E}$.

Proposition 1. Let $\boldsymbol{E}$, $\boldsymbol{E} \in I\!R^{h\times h}$, rank$(\boldsymbol{E}) = k < h$ be a rank deficient matrix. Then an orthogonal complement $\boldsymbol{E}^{\perp}$ of $\boldsymbol{E}$ is

$$\boldsymbol{E}^{\perp} = \boldsymbol{D}\boldsymbol{U}_2^T \tag{5}$$

where $\boldsymbol{U}_2^T$ is the null space of $\boldsymbol{E}$ and $\boldsymbol{D}$ is an arbitrary matrix of appropriate dimension.

Proof. (Filasová and Krokavec (2010b)) The SVD of $\boldsymbol{E}$, $\boldsymbol{E} \in I\!R^{h\times h}$, rank$(\boldsymbol{E}) = k < h$ gives

$$\boldsymbol{U}^T\boldsymbol{E}\boldsymbol{V} = \begin{bmatrix}\boldsymbol{U}_1^T\\ \boldsymbol{U}_2^T\end{bmatrix}\boldsymbol{E}[\boldsymbol{V}_1\ \boldsymbol{V}_2] = \begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{0}_{12}\\ \boldsymbol{0}_{21} & \boldsymbol{0}_{22}\end{bmatrix} \tag{6}$$

where $\boldsymbol{U}^T \in I\!R^{h\times h}$ is the orthogonal matrix of the left singular vectors, $\boldsymbol{V} \in I\!R^{h\times h}$ is the orthogonal matrix of the right singular vectors of $\boldsymbol{E}$ and $\boldsymbol{\Sigma}_1 \in I\!R^{k\times k}$ is the diagonal positive definite matrix

$$\boldsymbol{\Sigma}_1 = \mathrm{diag}\left[\,\sigma_1 \ \cdots \ \sigma_k\,\right], \ \sigma_1 \geq \cdots \geq \sigma_k > 0 \tag{7}$$

which diagonal elements are the singular values of $\boldsymbol{E}$. Using orthogonal properties of $\boldsymbol{U}$ and $\boldsymbol{V}$, i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_h$, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_h$, $\boldsymbol{U}_2^T\boldsymbol{U}_1 = \boldsymbol{0}$, then

$$\begin{aligned}\boldsymbol{E} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = [\boldsymbol{U}_1\ \boldsymbol{U}_2]\begin{bmatrix}\boldsymbol{\Sigma}_1 & \boldsymbol{0}_{12}\\ \boldsymbol{0}_{21} & \boldsymbol{0}_{22}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}_1^T\\ \boldsymbol{V}_2^T\end{bmatrix} = \\ = [\boldsymbol{U}_1\ \boldsymbol{U}_2]\begin{bmatrix}\boldsymbol{S}_1\\ \boldsymbol{0}_2\end{bmatrix} = \boldsymbol{U}_1\boldsymbol{S}_1\end{aligned} \tag{8}$$

where $\boldsymbol{S}_1 = \boldsymbol{\Sigma}_1\boldsymbol{V}_1^T$. Thus, (8) implies

$$\boldsymbol{U}_2^T\boldsymbol{E} = \boldsymbol{U}_2^T\,[\boldsymbol{U}_1\ \boldsymbol{U}_2]\begin{bmatrix}\boldsymbol{S}_1\\ \boldsymbol{0}_2\end{bmatrix} = \boldsymbol{0} \tag{9}$$

It is evident that for an arbitrary matrix $\boldsymbol{D}$ is

$$\boldsymbol{D}\boldsymbol{U}_2^T\boldsymbol{E} = \boldsymbol{E}^{\perp}\boldsymbol{E} = \boldsymbol{0} \tag{10}$$

respectively, which implies (5). ■

3.2 System Model Canonical Form

Proposition 2. If rank$(\boldsymbol{C}\boldsymbol{B}) = m$ then there exists a coordinates change in which $(\boldsymbol{A}^{\circ}, \boldsymbol{B}^{\circ}, \boldsymbol{C}^{\circ})$ takes the structure

$$\boldsymbol{A}^{\circ} = \begin{bmatrix}\boldsymbol{A}_{11}^{\circ} & \boldsymbol{A}_{12}^{\circ}\\ \boldsymbol{A}_{21}^{\circ} & \boldsymbol{A}_{22}^{\circ}\end{bmatrix},\ \boldsymbol{B}^{\circ} = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{B}_2^{\circ}\end{bmatrix},\ \boldsymbol{C}^{\circ} = [\boldsymbol{0}\ \boldsymbol{I}_m] \tag{11}$$

where $\boldsymbol{A}_{11}^{\circ} \in I\!R^{(n-m)\times(n-m)}$, $\boldsymbol{B}_2^{\circ} \in I\!R^{m\times m}$ is a nonsingular matrix, and $\boldsymbol{I}_m \in I\!R^{m\times m}$ is identity matrix.

Proof. (Filasová and Krokavec (2010a)) Considering the state-space description of the system (1), (2) with $r = m$ and defining the transform matrix $\boldsymbol{T}_1^{-1}$ such that

$$\boldsymbol{C}_1 = \boldsymbol{C}\boldsymbol{T}_1 = [\,\boldsymbol{0}\ \boldsymbol{I}_m\,], \qquad \boldsymbol{T}_1^{-1} = \begin{bmatrix}\boldsymbol{I}_{n-m}\ \boldsymbol{0}\\ \boldsymbol{C}\end{bmatrix} \tag{12}$$

then

$$\boldsymbol{B}_1 = \boldsymbol{T}_1^{-1}\boldsymbol{B} = \boldsymbol{T}_1^{-1}\begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{B}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{B}_1\\ \boldsymbol{C}\boldsymbol{B}\end{bmatrix} = \begin{bmatrix}\boldsymbol{B}_{11}\\ \boldsymbol{B}_{12}\end{bmatrix} \tag{13}$$

If $\boldsymbol{C}\boldsymbol{B} = \boldsymbol{B}_{12}$ is a regular matrix (in opposite case the pseudoinverse of $\boldsymbol{B}_{12}$ is possible to use), then the second transform matrix $\boldsymbol{T}_2^{-1}$ can be defined as follows

$$\boldsymbol{T}_2^{-1} = \begin{bmatrix}\boldsymbol{I}_{n-m} & -\boldsymbol{B}_{11}\boldsymbol{B}_{12}^{-1}\\ \boldsymbol{0} & \boldsymbol{I}_m\end{bmatrix} \tag{14}$$

$$\boldsymbol{T}_2 = \begin{bmatrix}\boldsymbol{I}_{n-m} & \boldsymbol{B}_{11}\boldsymbol{B}_{12}^{-1}\\ \boldsymbol{0} & \boldsymbol{I}_m\end{bmatrix} \tag{15}$$

This results in

$$\boldsymbol{B}^{\circ} = \boldsymbol{T}_2^{-1}\boldsymbol{B}_1 = \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{B}_2^{\circ}\end{bmatrix} = \boldsymbol{T}_2^{-1}\boldsymbol{T}_1^{-1}\boldsymbol{B} = \boldsymbol{T}_c^{-1}\boldsymbol{B} \tag{16}$$

where

$$\boldsymbol{B}_{11} = \boldsymbol{B}_1, \quad \boldsymbol{B}_2^{\circ} = \boldsymbol{B}_{12} = \boldsymbol{C}\boldsymbol{B}, \quad \boldsymbol{T}_c^{-1} = \boldsymbol{T}_2^{-1}\boldsymbol{T}_1^{-1} \tag{17}$$

and analogously

$$\boldsymbol{C}^{\circ} = \boldsymbol{C}_1\boldsymbol{T}_2 = \boldsymbol{C}\boldsymbol{T}_c = [\,\boldsymbol{0}\ \boldsymbol{I}_m\,]\,\boldsymbol{T}_2 = [\,\boldsymbol{0}\ \boldsymbol{I}_m\,] \tag{18}$$

Finally, with it yields

$$\boldsymbol{A}^{\circ} = \boldsymbol{T}_c^{-1}\boldsymbol{A}\boldsymbol{T}_c = \boldsymbol{T}_2^{-1}\boldsymbol{T}_1^{-1}\boldsymbol{A}\boldsymbol{T}_1\boldsymbol{T}_2 \tag{19}$$

Thus, (16), (18), and (19) implies (11). This concludes the proof. ■

Note, the structure of $\boldsymbol{T}_1^{-1}$ is not unique and others can be obtained by permutations of the first $n-m$ rows in the structure defined in (12).

3.3 System Modes Properties

Proposition 3. Given system eigenstructure with distinct eigenvalues then for $j, k \in \{1, 2, \ldots n\}, l \in \{1, 2, \ldots m\}$, $m = r$

i. the k-th mode $(s - s_k)$ is unobservable from the l-th system output if the l-th row of matrix $\boldsymbol{C}$ is orthogonal to the k-th eigenvector of the closed-loop system matrix $\boldsymbol{A}_c$, i.e. with $j \neq k$

$$\boldsymbol{c}_l^T\boldsymbol{n}_k = \boldsymbol{n}_j^T\boldsymbol{n}_k = 0, \quad \boldsymbol{C}^T = [\boldsymbol{c}_1\ \cdots\ \boldsymbol{c}_m] \tag{20}$$

ii. the k-th mode $(s - s_k)$ is uncontrollable from the l-th system input if the l-th column of matrix $\boldsymbol{B}$ is orthogonal to the k-th eigenvector of the closed-loop system matrix $\boldsymbol{A}_c$, i.e. with $j \neq k$

$$\boldsymbol{n}_k^T\boldsymbol{b}_l = \boldsymbol{n}_k^T\boldsymbol{n}_j = 0, \quad \boldsymbol{B} = [\boldsymbol{b}_1 \,\cdots\, \boldsymbol{b}_r] \tag{21}$$

Proof. (Krokavec and A. Filasová (2006)) Let $\boldsymbol{n}_k$ is the k-th right eigenvector corresponding to the eigenvalue s_k, i.e.

$$\boldsymbol{A}_c\boldsymbol{n}_k = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{n}_k = s_k\boldsymbol{n}_k \tag{22}$$

By definition, the closed-loop system resolvent kernel is

$$\boldsymbol{\Upsilon} = (s\boldsymbol{I}_n - \boldsymbol{A}_c)^{-1} \tag{23}$$

If the closed-loop system matrix is with distinct eigenvalues, (22) can be written in the compact form

$$\boldsymbol{A}_c\,[\boldsymbol{n}_1 \,\cdots\, \boldsymbol{n}_n] = [\boldsymbol{n}_1 \,\cdots\, \boldsymbol{n}_n]\begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{bmatrix} \tag{24}$$

$$\boldsymbol{A}_c\boldsymbol{N} = \boldsymbol{N}\boldsymbol{S}, \quad \boldsymbol{N}^{-1} = \boldsymbol{N}^T \tag{25}$$

respectively, where

$$\boldsymbol{S} = \operatorname{diag}[s_1 \cdots s_n]\,, \quad \boldsymbol{N} = [\boldsymbol{n}_1 \cdots \boldsymbol{n}_n] \tag{26}$$

Using the property of orthogonality given in (25), the resolvent kernel of the system takes form

$$\boldsymbol{\Upsilon} = \left(s\boldsymbol{N}\boldsymbol{N}^{-1} - \boldsymbol{N}\boldsymbol{S}\boldsymbol{N}^{-1}\right)^{-1} = \boldsymbol{N}(s\boldsymbol{I} - \boldsymbol{S})^{-1}\boldsymbol{N}^T \tag{27}$$

$$\boldsymbol{\Upsilon} = [\,\boldsymbol{n}_1 \,\cdots\, \boldsymbol{n}_n\,]\begin{bmatrix} \dfrac{1}{s-s_1} & & \\ & \ddots & \\ & & \dfrac{1}{s-s_n} \end{bmatrix}\begin{bmatrix} \boldsymbol{n}_1^T \\ \vdots \\ \boldsymbol{n}_n^T \end{bmatrix} \tag{28}$$

$$\boldsymbol{\Upsilon} = \sum_{h=1}^{n}\frac{\boldsymbol{n}_h\boldsymbol{n}_h^T}{s-s_h} \tag{29}$$

respectively. Thus, the closed loop transfer functions matrix takes form

$$\boldsymbol{G}(s) = \boldsymbol{C}\,(s\boldsymbol{I} - \boldsymbol{A}_c)^{-1}\boldsymbol{B}\boldsymbol{L} = \sum_{h=1}^{n}\frac{\boldsymbol{C}\boldsymbol{n}_h\boldsymbol{n}_h^T\boldsymbol{B}}{s-s_h}\boldsymbol{L} \tag{30}$$

It is obvious that (30) implies (20), (21). This concludes the proof. ■

4. EIGENSTRUCTURE ASSIGNMENT

In the pole assignment problem, a feedback gain matrix $\boldsymbol{K}$ is sought so that the closed-loop system has a prescribed eigenvalues spectrum $\Omega(\boldsymbol{A}_c) = \{s_h : \Re(s_h) < 0,\ h = 1, 2, ..., n\}$. Note, the spectrum $\Omega(\boldsymbol{A}_c)$ is closed under complex conjugation, and the observability and controllability of modes is determined by the closed-loop eigenstructure.

Considering the same assumptions as above then (22) can be rewritten as

$$[s_h\boldsymbol{I} - \boldsymbol{A}\ \boldsymbol{B}]\begin{bmatrix} \boldsymbol{n}_h \\ \boldsymbol{K}\boldsymbol{n}_h \end{bmatrix} = \boldsymbol{L}_h\begin{bmatrix} \boldsymbol{n}_h \\ \boldsymbol{K}\boldsymbol{n}_h \end{bmatrix} = \boldsymbol{0} \tag{31}$$

where $\boldsymbol{L}_h \in \mathbb{R}^{n\times(n+r)}$,

$$\boldsymbol{L}_h = [\,s_h\boldsymbol{I} - \boldsymbol{A}\ \boldsymbol{B}\,] \tag{32}$$

Subsequently, the singular value decomposition (SVD) of $\boldsymbol{L}_h$ gives

$$\begin{aligned} \begin{bmatrix} \boldsymbol{u}_{h1}^T \\ \vdots \\ \boldsymbol{u}_{hn}^T \end{bmatrix} \boldsymbol{L}_h \left[\boldsymbol{v}_{h1} \cdot \boldsymbol{v}_{hn}\boldsymbol{v}_{h,(n+1)} \cdot \boldsymbol{v}_{h,(n+r)}\right] = \\ = \begin{bmatrix} \sigma_{h1} & & & \\ & \ddots & & \mathbf{0}_{n+1} \cdot \mathbf{0}_{n+r} \\ & & \sigma_{hn} & \end{bmatrix} \end{aligned} \tag{33}$$

where $\{\boldsymbol{u}_{hl}^T, l = 1, 2, \ldots, n\}$, $\{\boldsymbol{v}_{hk}, k = 1, 2, \ldots, n+r\}$ are sets of the left and right singular vectors of $\boldsymbol{L}_h$ associated with the set of singular values $\{\sigma_{hl}, l = 1, 2, \ldots, n\}$

It is evident that vectors $\{\boldsymbol{v}_{hj}, j = n+1, n+2, \ldots, n+r\}$ satisfy (31), i.e.

$$\boldsymbol{L}\boldsymbol{v}_{hj} = [s_h\boldsymbol{I} - \boldsymbol{A}\ \boldsymbol{B}]\,\boldsymbol{v}_{hj} = \boldsymbol{0} \tag{34}$$

The set of vectors $\{\boldsymbol{v}_{hj}, j = n+1, n+2, \ldots, n+r\}$ is a non-trivial solution of (32), and results the null space of $\boldsymbol{L}_h$, $h = 1, 2, \ldots, n$

$$\begin{bmatrix} \boldsymbol{n}_h \\ \boldsymbol{K}\boldsymbol{n}_h \end{bmatrix} \in \mathcal{N}\,[\,s_h\boldsymbol{I} - \boldsymbol{A}\ \boldsymbol{B}\,] \tag{35}$$

The null space (35) consists of the normalized orthogonal set of vectors. Any combination of these vectors (the span of null space) will provide a vector $\boldsymbol{n}_h$ which used as an eigenvector produces the desired eigenvalue s_h in the closed-loop system matrix.

Proposition 4. The canonical form eigenstructure optimization provides optimal eigenstructure also for that model from which the canonical form was derived.

Proof. Using (16), (18), (19) and (22) it can be written

$$\begin{aligned} (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{n}_h &= (\boldsymbol{T}_c\boldsymbol{A}^\circ\boldsymbol{T}_c^{-1} - \boldsymbol{T}_c\boldsymbol{B}^\circ\boldsymbol{K}\boldsymbol{T}_c\boldsymbol{T}_c^{-1})\boldsymbol{n}_h = \\ &= \boldsymbol{T}_c(\boldsymbol{A}^\circ - \boldsymbol{B}^\circ\boldsymbol{K}^\circ)\boldsymbol{T}_c^{-1}\boldsymbol{n}_h = s_h\boldsymbol{n}_h \end{aligned} \tag{36}$$

$$(\boldsymbol{A}^\circ - \boldsymbol{B}^\circ\boldsymbol{K}^\circ)\boldsymbol{T}_c^{-1}\boldsymbol{n}_h = s_h\boldsymbol{T}_c^{-1}\boldsymbol{n}_h \tag{37}$$

$$(\boldsymbol{A}^\circ - \boldsymbol{B}^\circ\boldsymbol{K}^\circ)\boldsymbol{n}_h^\circ = \boldsymbol{A}_c^\circ\boldsymbol{n}_h^\circ = s_h\boldsymbol{n}_h^\circ \tag{38}$$

respectively, where $\boldsymbol{K}^\circ = \boldsymbol{K}\boldsymbol{T}_c$, $\boldsymbol{n}_h = \boldsymbol{T}_c\boldsymbol{n}_h^\circ$. Writing compactly $\{\boldsymbol{n}_h = \boldsymbol{T}_c\boldsymbol{n}_h^\circ,\ h = 1, 2, \ldots n\}$ as follows

$$\boldsymbol{N} = \boldsymbol{T}_c\boldsymbol{N}^\circ, \quad \boldsymbol{N}^{-1} = \boldsymbol{N}^{\circ T}\boldsymbol{T}_c^{-1} \tag{39}$$

then using (27), (30), (39) it yields

$$\begin{aligned} \boldsymbol{G}(s) &= \boldsymbol{C}\boldsymbol{\Upsilon}\boldsymbol{B}\boldsymbol{L} = \boldsymbol{C}\boldsymbol{N}(s\boldsymbol{I} - \boldsymbol{S})^{-1}\boldsymbol{N}^{-1}\boldsymbol{B}\boldsymbol{L} = \\ &= \boldsymbol{C}\boldsymbol{T}_c\boldsymbol{N}^\circ(s\boldsymbol{I} - \boldsymbol{S})^{-1}\boldsymbol{N}^{\circ T}\boldsymbol{T}_c^{-1}\boldsymbol{B}\boldsymbol{L} = \\ &= \boldsymbol{C}^\circ\boldsymbol{N}^\circ(s\boldsymbol{I} - \boldsymbol{S})^{-1}\boldsymbol{N}^{\circ T}\boldsymbol{B}^\circ\boldsymbol{L} \end{aligned} \tag{40}$$

$$\boldsymbol{G}(s) = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A}_c)^{-1}\boldsymbol{B}\boldsymbol{L} = \boldsymbol{C}^\circ(s\boldsymbol{I} - \boldsymbol{A}_c^\circ)^{-1}\boldsymbol{B}^\circ\boldsymbol{L} \tag{41}$$

$$\boldsymbol{G}(s) = \sum_{h=1}^{n}\frac{\boldsymbol{C}\boldsymbol{n}_h\boldsymbol{n}_h^T\boldsymbol{B}}{s-s_h}\boldsymbol{L} = \sum_{h=1}^{n}\frac{\boldsymbol{C}^\circ\boldsymbol{n}_h^\circ\boldsymbol{n}_h^{\circ T}\boldsymbol{B}^\circ}{s-s_h}\boldsymbol{L} \tag{42}$$

respectively. It is obvious that optimizing $\boldsymbol{C}^\circ\boldsymbol{n}_h^\circ$ is optimized $\boldsymbol{C}\boldsymbol{n}_h$. This concludes the proof. ■

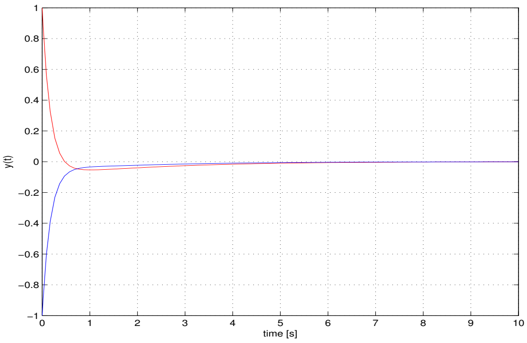


Fig. 1. System output response

5. PARAMETER DESIGN

Using eigenvector orthogonal properties, (22) can be rewritten for $h = 1, 2, \ldots, n$ as follows

$$(s_h\boldsymbol{I} - \boldsymbol{A}^\circ)\boldsymbol{n}_h^\circ = -\boldsymbol{B}^\circ\boldsymbol{K}^\circ\boldsymbol{n}_h^\circ = -\boldsymbol{B}^\circ\boldsymbol{r}_h^\circ \tag{43}$$

$$\boldsymbol{n}_h^\circ = -(s_h\boldsymbol{I} - \boldsymbol{A}^\circ)^{-1}\boldsymbol{B}^\circ\boldsymbol{r}_h^\circ = \boldsymbol{V}_h^\circ\boldsymbol{r}_h^\circ \tag{44}$$

respectively, where

$$\boldsymbol{r}_h^\circ = \boldsymbol{K}^\circ\boldsymbol{n}_h^\circ, \quad \boldsymbol{V}_h^\circ = -(s_h\boldsymbol{I} - \boldsymbol{A}^\circ)^{-1}\boldsymbol{B}^\circ \tag{45}$$

Subsequently, it can be obtained

$$\boldsymbol{r}_h^\circ = \boldsymbol{V}_h^{\circ\,\ominus 1}\boldsymbol{n}_h^\circ \tag{46}$$

where

$$\boldsymbol{V}_h^{\circ\,\ominus 1} = (\boldsymbol{V}_h^{\circ T}\boldsymbol{V}_h^\circ)^{-1}\boldsymbol{V}_h^{\circ T} \tag{47}$$

is Moore-Penrose pseudoinverse of $\boldsymbol{V}_h^\circ$.

Of interest are the eigenvectors of the closed-loop system having minimal orthogonal projection to rows of the orthogonal complement $\boldsymbol{C}^{\circ T\perp}$ of the output matrix $\boldsymbol{C}^{\circ T}$ and associated with m element eigenvalues subset $\rho(\boldsymbol{A}^\circ)$ ($m =$ rank$(\boldsymbol{C}^\circ)$, $\rho(\boldsymbol{A}^\circ) \subset \Omega(\boldsymbol{A}^\circ)$) of the desired closed-loop eigenvalues set $\Omega(\boldsymbol{A}^\circ) = \{s_h,\ \Re(s_h) < 0,\ h = 1, 2, \ldots, n\}$, $\Omega(\boldsymbol{A}^\circ) = \Omega(\boldsymbol{A})$. The rest $(n-m)$ eigenvalues can be associated with rows of the complement matrix $\boldsymbol{C}^\bullet$ obtained in such way that all zero elements in $\boldsymbol{C}^\circ$ be changed to ones, and all ones to zeros. Note, direct use of $\boldsymbol{C}^\circ$ maximize matrix weights of modes.

Let $\rho(\boldsymbol{A}^\circ) = \{s_h,\ \Re(s_h) < 0,\ h = 1, 2, \ldots, n\}$, then

$$\boldsymbol{r}_h^\diamond = \boldsymbol{V}_h^{\circ\,\ominus 1}\boldsymbol{c}_h^{\circ T\perp T},\ h = 1, 2, \ldots, m \tag{48}$$

$$\boldsymbol{r}_h^\bullet = \boldsymbol{V}_h^{\circ\,\ominus 1}\boldsymbol{c}_h^{\bullet T},\ h = m{+}1, \ldots n \tag{49}$$

Thus, computing

$$\boldsymbol{n}_h^\diamond = \boldsymbol{V}_h^\circ\boldsymbol{r}_h^\diamond, \quad \boldsymbol{n}_h^\bullet = \boldsymbol{V}_h^\circ\boldsymbol{r}_h^\bullet \tag{50}$$

it is possible to construct and to separate the matrix $\boldsymbol{Q}^\circ$ of the form

$$\boldsymbol{Q}^\circ = [\boldsymbol{v}_1^\diamond \cdots \boldsymbol{v}_m^\diamond \boldsymbol{v}_{m+1}^\bullet \cdots \boldsymbol{w}_m^\bullet] = \begin{bmatrix}\boldsymbol{P}^\circ \\ \boldsymbol{R}^\circ\end{bmatrix} \tag{51}$$

with $\boldsymbol{P}^\circ \in I\!R^{n\times n}$, $\boldsymbol{R}^\circ \in I\!R^{r\times n}$ such that

$$\boldsymbol{K}^\circ = \boldsymbol{R}^\circ\boldsymbol{P}^{\circ -1}, \quad \boldsymbol{K} = \boldsymbol{K}^\circ\boldsymbol{T}_c^{-1} \tag{52}$$

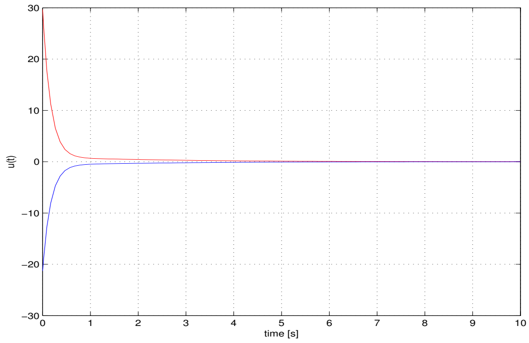


Fig. 2. Control actions

6. ILLUSTRATIVE EXAMPLE

The system under consideration was described by (1), (2), where

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -9 & -5 \end{bmatrix},\ \boldsymbol{B} = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 5 \end{bmatrix},\ \boldsymbol{C}^T = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 0 \end{bmatrix}$$

Constructing the transformation matrices

$$\boldsymbol{T}_c^{-1} = \begin{bmatrix} 4.0 & 0.5 & -2.5 \\ 1.0 & 2.0 & 1.0 \\ 1.0 & 1.0 & 0.0 \end{bmatrix},\ \boldsymbol{T}_c = \begin{bmatrix} 1.0 & 2.5 & -5.5 \\ -1.0 & -2.5 & 6.5 \\ 1.0 & 3.5 & -7.5 \end{bmatrix}$$

the system model canonical form parameters were computed as

$$\boldsymbol{A}^\circ = \begin{bmatrix} -1 & 10.5 & 6 \\ 0 & -3.0 & -2 \\ 0 & 1.0 & -1 \end{bmatrix},\ \boldsymbol{B}^\circ = \begin{bmatrix} 0 & 0 \\ 7 & 10 \\ 3 & 4 \end{bmatrix},\ \boldsymbol{C}^\circ = [\,\boldsymbol{0}\ \boldsymbol{I}_2\,]$$

Thus, considering $\Omega(\boldsymbol{A}^\circ) = \{-0.5,\ -1.2,\ -6\}$ it is

$$\boldsymbol{V}_1^\circ = \begin{bmatrix} -37.3846 & -54.4615 \\ 0.7692 & 0.9231 \\ -4.4615 & -6.1538 \end{bmatrix}$$

$$\boldsymbol{V}_2^\circ = \begin{bmatrix} -10.0610 & -5.4878 \\ 4.5122 & 6.0976 \\ -7.5610 & -10.4878 \end{bmatrix}$$

$$\boldsymbol{V}_3^\circ = \begin{bmatrix} -5.2059 & -7.3059 \\ 2.4118 & 3.4118 \\ 0.1176 & 0.0176 \end{bmatrix}$$

and with $\boldsymbol{c}^{\circ T\perp} = [1\ 0\ 0]$, $\boldsymbol{c}_1^{\bullet T} = [1\ 0\ 1]$ it yields

$$\boldsymbol{r}_1^\circ = \begin{bmatrix} 0.3891 \\ -.0.2854 \end{bmatrix},\ \boldsymbol{r}_2^\circ = \begin{bmatrix} -0.1645 \\ 0.1194 \end{bmatrix} \boldsymbol{r}_3^\bullet = \begin{bmatrix} 18.4978 \\ -13.2737 \end{bmatrix}$$

$$\boldsymbol{n}_1^{\circ T} = [0.9983\ 0.0358\ 0.0205]$$

$$\boldsymbol{n}_2^{\circ T} = [0.9997\ -0.0144\ -0.0082]$$

$$\boldsymbol{n}_3^{\bullet T} = [0.6788\ -0.6745\ 0.6146]$$

Constructing the matrix $\boldsymbol{Q}^\circ$

$$\boldsymbol{Q}^\circ = \left[\begin{array}{rrr} 0.9983 & 0.9997 & 0.6788 \\ 0.0358 & 0.0144 & -0.6745 \\ 0.0205 & -0.0082 & 0.6146 \\ \hline 0.3891 & -0.1645 & 18.4978 \\ -0.2854 & 0.1194 & -13.2737 \end{array}\right] = \left[\begin{array}{c} \boldsymbol{P}^\circ \\ \hline \boldsymbol{R}^\circ \end{array}\right]$$

the control law parameters satisfying (52) are

$$\boldsymbol{K}^{\circ} = \begin{bmatrix} -0.0062 & -3.7944 & 25.9402 \\ 0.0036 & 2.6301 & -18.7151 \end{bmatrix}$$

$$\boldsymbol{K} = \begin{bmatrix} 22.1212 & 18.3483 & -3.7990 \\ -16.0707 & -13.4532 & 2.6211 \end{bmatrix}$$

It is possible to verify that closed-loop system matrix eigenvalues belongs to the desired one.

In the presented Fig. 1, 2 the example is shown of the unforced closed-loop system output response, as well as control actions, where nonzero initial state was considered.

7. CONCLUDING REMARKS

This paper provides a design method for memory-free controllers where the general problem of assigning the eigenstructure for state variable mode decoupling in state feedback control design is considered. The method exploits standard numerical optimization procedures to manipulate the system feedback gain matrix as a direct design variable. The manipulation is accomplished in a manner that produces desired system global performance by pole placement and output dynamics by modification of the mode observability.

With generalization of the known algorithms for pole assignment the modified exposition of the problem is presented here to handle the optimized structure of the left eigenvector set in state feedback control design. Presented method makes full use of the freedom provided by eigenstructure assignment to find a controller which stabilizes the closed-loop system. Therefore, the feedback control law has a clear physical meaning and provides a valid design method of the controller for real systems. It is shown by appropriately assigning closed-loop eigenstructure in state feedback control the overall stability is achieved. Finally the design methodology is illustrated by an example.

ACKNOWLEDGMENT

The work presented in this paper was supported by VEGA, Grant Agency of Ministry of Education and Academy of Science of Slovak Republic under Grant No. 1/0256/11, as well by Research & Development Operational Programme Grant No. 26220120030 realized in Development of Centre of Information and Communication Technologies for Knowledge Systems. These supports are very gratefully acknowledged.

REFERENCES

O. Bachelier, J. Bosche, and D. Mehdi. On pole placement via eigenstructure assignment Approach, *IEEE Transactions on Automatic Control*, 51:9, 1554-1558, 2006.

R.S. Burns. *Advanced Control Engineering*, Butterworth-Heinemann, Oxford, 2001.

B.N. Datta. *Numerical Methods for Linear Control Systems: Design and Analysis*, Elsevier, London, 2004.

G.H. Golub and C.F. Van Loan. *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.

A. Filasová. Robust control design for large-scale uncertain dynamic system, In *New Trends in Design of Control Systems*, STU, Bratislava, 247-432, 1997.

A. Filasová. Robust control design: An optimal control approach, In *Proceedings of the IEEE International Conference on Intelligent Engineering Systems INES'99*, Stará Lesná, Slovakia, 515-518, 1999.

A. Filasová and D. Krokavec. On sensor faults estimation using sliding mode observers, In *Conference on Control and Fault-Tolerant Systems SysTol'10*, Nice, France, 2010, 44-49.

A. Filasová and D. Krokavec. State estimate based control design using the unified algebraic approach, *Archives of Control Sciences*, 20:1, 5-18, 2010.

I.C.S. Ipsen. *Numerical Matrix Analysis. Linear Systems and Least Squares*, SIAM, Philadelphia, 2009.

J. Kautsky, N. Nichols, and P. Van Dooren. Robust pole assignment in linear state feedback, *International Journal of Control*, 41:5, 1129-1155, 1985.

P. Kocsis and D. Krokavec. State variables mode decoupling in state control design for linear MIMO systems, In *International Conference Cybernetics and Informatics 2008*, Ždiar, Slovak Republic, 44.1-44.8, 2008.

D. Krokavec and A. Filasová. *Dynamic Systems Diagnosis*, Elfa, Košice, 2007. (in Slovak)

A.S. Poznyak. *Advanced Mathematical Tools for Automatic Control Engineers: Deterministic Techniques*, Elsevier, London 2008.

K.M. Sobel and F.J. Lallman. Eigenstructure assignment for the control of highly augmented aircraft, *Journal of Guidance, Control and Dynamics*, 12:3, 318-324, 1989.

Q.G. Wang. *Decoupling Control*, Springer-Verlag, Berlin, 2003.

W.M. Wonham. *Linear Multivariable Control: A Geometric Approach*, Springer-Verlag, New York, 1985.

X.H. Xu and X.K. Xie. Eigenstructure assignment by output feedback in descriptor systems, *IMA Journal of Mathematical Control & Information*, 12, 127-132, 1995.

APPLICATION OF MATHEMATICAL MORPHOLOGY ON NANOSTRUCTURE IMAGE PROCESSING

J. Petrová * M. Mudrová * A. Procházka * J. Fojt **

* *Department of Computing and Control Engineering, ICT Prague,*
Technická 5, 166 28 Praha 6
fax : +420 220 445 053 and e-mail : petrovaj@vscht.cz
** *Department of Metals and Corrosion Engineering, ICT Prague,*
Technická 5, 166 28 Praha 6
fax : +420 220 444 400 and e-mail : fojtj@vscht.cz

Abstract: Mathematical morphology is a very effective tool on image segmentation. This paper presents selected methods of mathematical morphology that are applied to the microscopic images of nanomaterials. The first part briefly describes the mathematical background of the fundamental morphological methods, including their application on the test image. The second part contains an application both of binary and grayscale morphological methods on the nanostructure images. The influence of various structure elements is investigated, as well.

Keywords: binary morphology, nanostructure image processing, grayscale morphology

1. INTRODUCTION

Image processing is applied as an effective tool for various image analysis in many branches of science, Kuehn (1998), Wu (1995), Sorzanoa (2004). Presented paper is focused on analysis of nanomaterial images.

These nanomaterial seems to be one of the options of surface bioactivation of titanium implants that are used in medicine, Fojt (2010). Nanomaterial properties assessment is one of the essential issues in the process of material development while the tube diameter should be evaluated as well as a side thickness and other quantities. The material properties quantities could be based on the processing of microscopic images, Hodneland (2006), Bunes (2008), Kiang (1998), Gommes (2003).

A lot of various methods are used for segmentation and analysis of given microscopy images, such as a low-pass filtering, application of watershed transform, adaptive thresholding, mathematical morphology Dougherty (2003), Heijmans (1998), Horgan (1998), convex hull and others.

Methods of mathematical morphology could be used for processing both of binary and grayscale images, Gonzalez (2002), Hlavac (2002), Hanbury (2001). This paper looks out only a part of the complex algorithm of given data processing that concerns of an application of binary mathematical morphology methods.

2. MATHEMATICAL BACKGROUND

Mathematical morphology provides a tool for extraction of those image components that are useful in the representation and description of a region shape. The most common use of mathematical morphology is in image enhancement, segmentation and restoration, edge detection, texture and component analysis, curve filling, noise reduction etc. A grayscale digital image generally can be represented as a set of elements, values of which are vectors of the 3–D integer space Z^3. The first two components represent coordinates (x, y) of the pixel and the third one its discrete graylevel value. A binary image A can be considered as a set of n white pixels $A = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where vector $\{(x_i, y_i) \in Z^2\}$ corresponds to pixel coordinates.

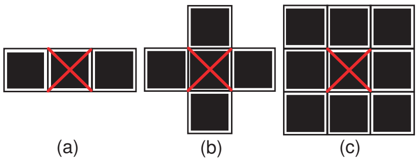


Figure 1. Examples of common structural elements:
(a) $B1 = \{(-1,0), (0,0), (1,0)\}$,
(b) $B2 = \{(-1,0), (0,-1), (0,0), (1,0), (0,1)\}$,
(c) $B3 = \{(-1,0), (0,-1), (0,0), (1,0), (0,1), (1,1), (1,-1), (-1,-1), (-1,1)\}$

Morphological operators are given by the relationships between a set of points of an original image A and another point set, which is called a structural element B. Various types of structural elements can be used. Examples of the most common structural elements are presented in Fig. 1.

Dilation, erosion, opening and closing are the basic morphological transformations.

2.1 Binary Morphology

Morphological operations can be defined with Minkowski operators, other description can be used, as well, Gonzalez (2002), Farouki (2001).

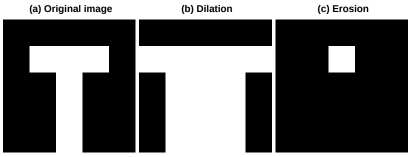


Figure 2. (a) Original image (point set A), (b) Dilation of A by structure element *B1*, (c) Erosion of A by *B1*

A *Dilation* operator of A by B, denoted $A \oplus B$ is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \tag{1}$$

where A is a set of points of the original image and set B is a selected structural element. $\hat{B}$ means a reflection of the set B. Dilation of A by B is the set of all displacements, z, such that $\hat{B}$ and A overlap by at least one element. A gaps bridging is one of common applications of dilation. Dilation with a suitable structural element use makes objects larger and so that it fills gaps in an image.

An *Erosion* operator of A by B, denoted $A \ominus B$ is defined as

$$A \ominus B = \{z \mid (B)_z \subseteq A\} \tag{2}$$

where A is a set of points of an original image and a set B is a structural element, again. Erosion of A by B is the set of all points z such that B, translated by z, is contained in A.

The simplest application of erosion is an elimination of small details in an image. Objects smaller than a structural element are removed. This property forms a base for a morphological operations. Example of a simple effect of dilation and erosion on the test image is presented in Fig. 2. The structural element $B1$ is used for test. Dilation expands objects and erosion shrinks them. A combination both of them changes the image but saves object size.

Opening of an image set point A with a structural element B, denoted $A \circ B$, is defined as an erosion followed by a dilation with the same structural element use

$$A \circ B = (A \ominus B) \oplus B. \tag{3}$$

Closing A with B, denoted $A \bullet B$, is defined as a dilation followed by an erosion

$$A \bullet B = (A \oplus B) \ominus B. \tag{4}$$

Both of these operations smooth the object contours. Opening separates objects connected with narrow isthmuses, closing eliminates small gaps and gets through narrow breaks. An example of simple effect of opening and closing is presented in Fig. 3. The same structural element as in the Fig. 2 has been used.

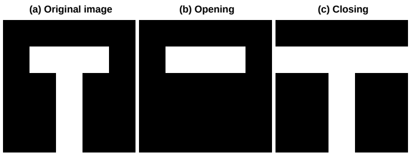


Figure 3. (a) Original image (point set A), (b) Opening A by structure element $B1$, (c) Closing A by $B1$

2.2 Grayscale Morphology

Basic binary morphological operation can be extended to grayscale images, Gonzalez (2002), Hanbury (2001). Operations deal with the input image function $f(x, y)$ and a structural element $b(x, y)$.

Grayscale dilation of f by b, $f \oplus b$, is defined as

$$\begin{aligned}(f \oplus b)(s,t) = max\{f(s-x, t-y) + b(x, y) \mid \\ \mid (s-x), (t-y) \in D_f; (x, y) \in D_b\}\end{aligned} \tag{5}$$

where D_f and D_b are the domains of f and b.

After application of dilation on a grayscale image, output image is brighter than an input one if all the values of the structural element are positive. Dark details are reduced or eliminated.

Grayscale erosion is based on the minimum value of $(f - b)$ choosing in the interval defined by structural element b. The Erosion of f by b, $f \ominus b$, is defined as

$$\begin{aligned}(f \ominus b)(s,t) = min\{f(s+x, t+y) - b(x, y) \mid \\ \mid (s+x), (t+y) \in D_f; (x, y) \in D_b\}\end{aligned} \tag{6}$$

where D_f and D_b are the domains of f and b.

Output image after grayscale erosion application is darker than input one if all elements of the structural element are positive. The bright details that are smaller than the structural element are reduced.

Grayscale opening and closing have the same form as their binary alternates. The opening of grayscale image f by the structural element b, $f \circ b$, is defined as

$$f \circ b = (f \ominus b) \oplus b \tag{7}$$

The grayscale closing of f by b, $f \circ b$, is defined as

$$f \bullet b = (f \oplus b) \ominus b \tag{8}$$

Top-hat transformation h of an image f is defined as

$$h = f - (f \circ b) \tag{9}$$

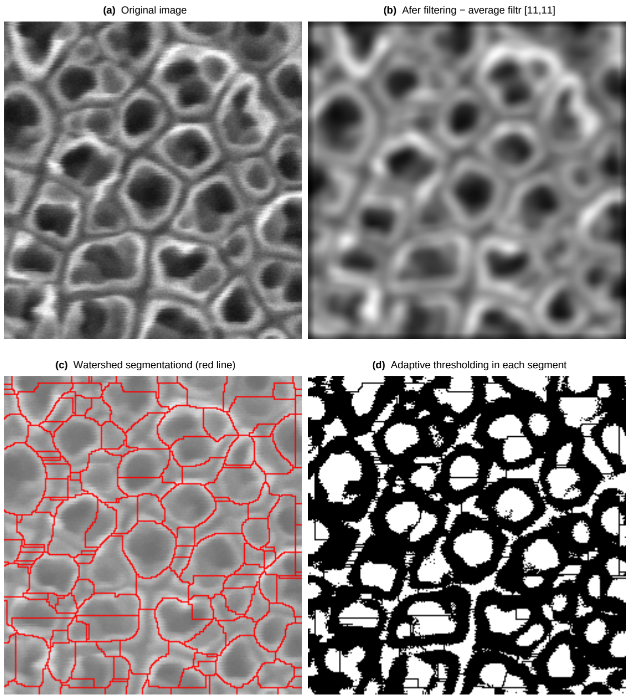
where f is an original image and b is a structural element. This transformation is useful for enhancing detail in background.

Morphological gradient, denoted g, highlight sharp grey-level transitions in the input image. It is defined as

$$g = (f \oplus b) - (f \ominus b) \tag{10}$$

3. NANOSTRUCTURE IMAGE PROCESSING

Nanostructure image processing features a very important part of classification process of nanomaterials quality. There is presented a selected cut of image of titanium dioxide tubes, Fojt (2010), in Fig. 4a. One of the important factor representing the material quality is an inner tube diameter. Methods of image processing seems to be a suitable tool to obtain the desired value. The essential step of the nanomaterial image processing is a separation of the inner area of tubes from the background. Preprocessing of these images based on digital filters application is



(a) Original image

(b) Afer filtering − average filtr [11,11]

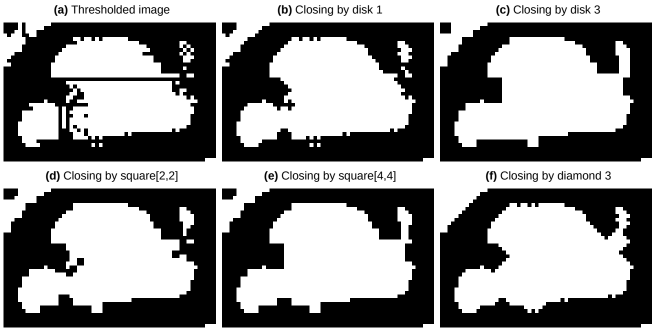
(c) Watershed segmentationd (red line)

(d) Adaptive thresholding in each segment

Figure 4. Image preprocessing before morphology use: (a) An original image, (b) Lowpass filter application, (c) Watershed segmentation, (d) Adaptive thresholding result

necessary, as well. The filter size and cutoff frequency of lowpass filter used dependent on image character. The filtration reduces a level of noise in the image and it improves results of segmentation in the following step, Fig. 4b.

In the next step the image is divided into segments by means of watershed transform Eq. (10), Fig. 4c. After that, an adaptive thresholding is performed inside of each segment. Adaptive thresholding is used because of very different grayscale level in each part of the image. Setting of one thresholding value for whole the image is unsuitable in this case. Although the adaptive thresholding application provides satisfying results, there are still undesired artifacts in the resulting image, Fig. 4d. Some of them imply from the inner area selection only, without segments border. Application of mathematical morphology seems



(a) Thresholded image

(b) Closing by disk 1

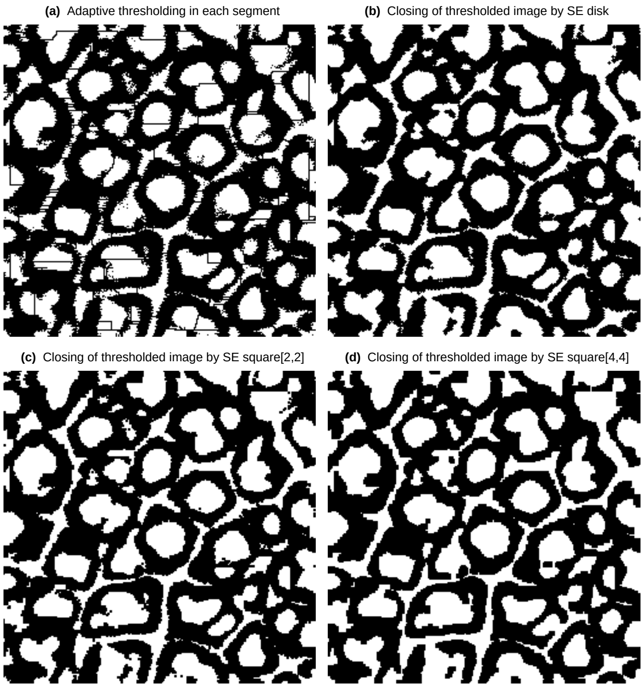
(c) Closing by disk 3

(d) Closing by square[2,2]

(e) Closing by square[4,4]

(f) Closing by diamond 3

Figure 5. Selected results of closing operation by means of various types of structural element in zoomed cut of an image: (a) Part of thresholded image, (b) Closing by disk size 1, (b) Closing by disk size 3, (d) Closing by square size [2,2], (e) Closing by square size [4,4], (f) Closing by diamond size 3



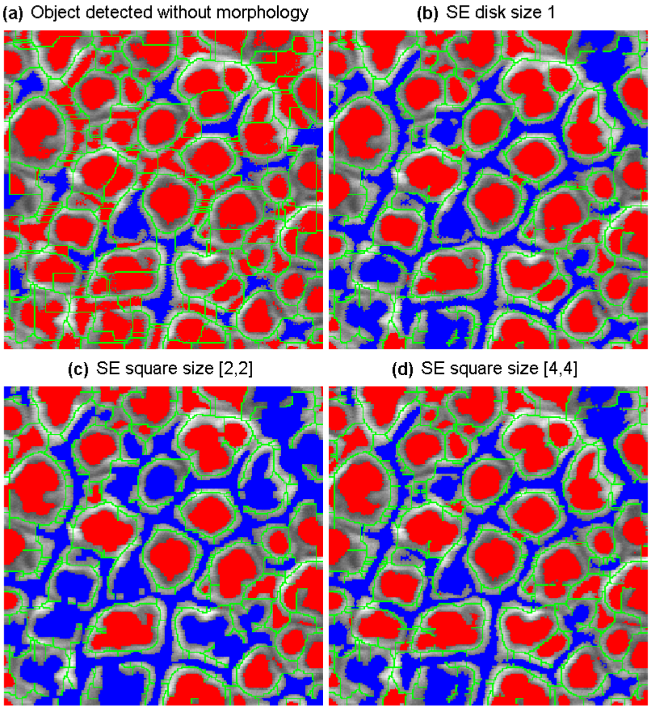
(a) Adaptive thresholding in each segment

(b) Closing of thresholded image by SE disk

(c) Closing of thresholded image by SE square[2,2]

(d) Closing of thresholded image by SE square[4,4]

Figure 6. Closing of thresholded image cut by various structural element: (a) Thresholded image, (b) Closing by disk 1, (b) Closing by structural element type disk size 1, (c) Closing by structural element type square size [2,2], (d) Closing by structural element type square size [4,4]



(a) Object detected without morphology

(b) SE disk size 1

(c) SE square size [2,2]

(d) SE square size [4,4]

Figure 7. Convex area (red) and non-convex area (blue) detected by various SE: (a) without morphology, (b) SE disk size 1, (c) SE square size [2,2], (d) SE square size [4,4]

to be an effectual tool for elimination of these undesired effects, Fig. 6. Closing operation can remove small holes inside the objects and it smooths out the objects contours. The result of this operation depends on the shape of structural element used, of course. Various structural elements B were tested on the real image set. A selected zoom cut processed by mean of various structural element is presented in Fig 5. Only the symmetric structural elements have been considered, as it is possible to suppose an isotropic material properties.

As the tubes are supposed to have a convex crosscut, nonconvex areas are excluded from the next processing and the area of convex segment is considered only, Fig. 7.

Results obtained with the application of various SE are presented in Tab. 1. Sample of selected structural element is Disk1=$\{(-1,0),(0,-1),(0,0),(1,0),(0,1)\}$ or Square[2,2]=$\{(0,0),(0,1),(1,0),(1,1)\}$.

The application of mathematical morphology methods implicates reduction of a number of objects found in the image, see Tab. 1. Results obtained by application of various structural elements are comparable. It's not possible to appoint the best structural element explicitly. The elements in form of disk and square seems to be useful for this type of images.

Table 1. Results obtained with various structural elements use

Structural Element	The Number of Tubes	Tube Area Percentage of the Image	Average Tube Area (px)	Standard Deviation
–	248	32,21	85,11	147,33
disk 1	100	27,95	183,16	254,08
disk 3	50	21,27	278,84	304,34
square [2,2]	83	27,49	217,06	262,94
square [4,4]	58	25,52	288,34	294,39
diamond 3	61	21,51	231,13	293,58

4. CONCLUSION

Nanostructure image processing is a very important part of classification process of nanomaterials quality. Application of mathematical morphology methods can improve object detection in these images. The best structural element for this detection cannot be exactly defined, as the image resolution seems to be a basic factor for structural element size and shape selection, too. Obtained results depend on an original image quality, as well. Based on testing, structural element in the form disk and square seems to suitable for processing of the given images.

Used images originate from electron microscope TESCAN VEGA 3 SBU. The algorithm was developed in MATLAB environment, ver. 2010a, and it was tested on computer with following parameters: processor Intel(R) Core(T2) 2CPU, 2,40 GHz, RAM 4GB, OS Win7 32bit.

Application of various type of a structural element has similar demands concerning a runtime. There is a comparison of runtime values for selected structural elements use, Tab. 2.

Table 2. Runtime obtained with an application of various structural elements

	Disk 1	Disk 3	Square[2,2]	Square[4,4]	Diamond3
Runtime[s]	0,118	0,234	0,031	0,027	0,066

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Education of the Czech Republic (programs No. MSM 6046137306 and MSM 6046137302). Financial support from specific university research (MSMT No 21/2011) is acknowledged, as well.

REFERENCES

Gonzalez R.C. and Woods R.E. *Digital Image Processing.* Prentice Hall, Upper Saddle River, New Jersey 07458, 2002.

Dougherty E.R. and Lotufo R.A. *Hands-on Morphological Image Processing.* SPIE, Bellingham, Washington 98227-0010 USA, 2003.

Fojt J., Moravec H. and Joska L. Nanostrukturování slitiny titan-hliník-vanad. *Koroze a ochrana materiálů*, 2010,volume 54 (4), pages 154–154.

Heijmans H.J.A.M. and Roerdink J.B.T.M. *Mathematical Morphology and its Applications to Image and Signal Processing.* Kluwer Academic Publisher, P.O.Box 17, 3300 AA Dordrecht, The Netherlands, 1998.

Hlaváč V. and Sedláček M. *Zpracování signalů a obrazů.* Vydavatelství ČVUT, Praha, 2000.

Wu K., Gauthier D. and Levine M.D. Live Cell Image Segmentation. *IEEE Transactions on Biomedical Engineering*, Jan. 1995,volume 4, No. 1.

Kuehn M.,Hausner M., Bungartz H.-J., Wagner M.,Wilderer P.A. and Wuertz S. Automated Confocal Laser Scanning Microscopy and Semiautomated Image Processing for Analysis of Biofilms. *Applied and Enviromental Microbiology*, Nov. 1998, volume 64, No. 11.

Sorzanoa C.O.S., Marabinia R., Velazquez-Muriela J., Bilbao-Castroa J.R., Scheresa S.H.W., Carazoa J.M. and Pascual-Montanoa A. XMIPP: a new generation of an open-source image processing package for electron microscopy. *Journal od Structural Biology*, 2004, vol. 148, pages 194–204.

Hodneland E., Lundervold A., Gurke S., Tai X.-Ch., Rustom A. and Gerdes H.-H. Automated Detection of Tunneling Nanotubes in 3D Images. *International Society for Analytical Cytology*, 2006.

Farouki R.T., Moon H.P. and Ravani B. Minkowaki Geometric Algebra of Complex Sets. *Geometriae Dedicata*, 2001, vol. 85, pages 283–315.

Horgan G.W. Mathematical Morphology for Analysing Soil Structure Images. *European Journal of Soil Science*, June 1998, vol. 49, pages 161–173.

Hanbury A.G. and Serra J. Morphological Operators on the Unit Circle. *IEEE Transactions on Biomedical Engineering*, Dec. 2001, vol. 10, No. 12.

Bunes B.R., Catravas P.E. and Hagerman, M.E. Image Processing Algorithm for Analyzing Chirality in Carbon Nanotubes. 8th IEEE Conference on Nanotechnology, 2008.

Kiang C.-H., Endo M., Ajayan P.M., Dresselhaus G. and Dresselhaus M.S. Size Effects in Carbon Nanotubes. *Phys. Rev. Lett.*, 1998, vol. 81, pages 1869–1872.

Gommes C., Blacher S., Masenelli-Varlot K., Bossuot Ch., McRae E., Fonseca A., Nagy J.-B., Pirard J.-P. Image analysis characterization of multi-walled carbon nanotubes. *Carbon*, 2003, vol. 41, pages 2561–2572.

Safety Verification of Rule-Based Controllers

Michal Paulovič [*], **Michal Kvasnica** [*], **Alexander Szücs** [*], and **Miroslav Fikar** [*]

[*] *Institute of Information Engineering, Automation, and Mathematics*
Faculty of Chemical and Food Technology
Slovak University of Technology in Bratislava
Radlinského 9, 812 37 Bratislava, Slovakia
(Tel: +421 259 325 352; e-mail: michal.kvasnica@stuba.sk)

Abstract: This paper proposes how to transform a control algorithm, written in MATLAB, into a hybrid system in order to verify its stability properties. The procedure first converts the code into a corresponding HYSDEL equivalent, which is then used to generate a suitable mathematical model. Safety verification is then formulated as a mixed integer linear program with feasibility objective.

Keywords: hybrid systems, safety verification, reachability analysis, MATLAB, HYSDEL

1. INTRODUCTION

Processes that evolve according to dynamic equations and logic rules can be described by hybrid models (Bemporad and Morari, 1999). Typical examples are real-time systems, where physical plants are governed by embedded rule-based controllers. When such systems are designed, it is important to provide a certificate that they will always operate in a safe manner, e.g. that the control rules never drive the plant into an "unsafe" area. Such a certificate can be provided by performing *reachability analysis* (Lygeros et al., 1999; Torrisi, 2003), which answers the following question: given a set of initial conditions $\mathcal{X}_0$, find the initial condition $x(0) \in \mathcal{X}_0$ for which the plant enters a set of unsafe states $\mathcal{X}_f$ in a finite number of steps T. If the reachability problem is infeasible, there is a guarantee that no such "unsafe" initial condition exists, hence providing the required safety certificate.

In this paper we propose how to solve the reachability problem when the control rules are implemented as a standard MATLAB function, composed of several IF-THEN-ELSE logic rules. First, the code of the function is converted into the HYSDEL (Torrisi, 2002) language, which is a high-level language tailored for describing behavior of hybrid systems. The translation process creates a one-to-one equivalent of the MATLAB control loop from which a suitable mathematical description is derived. The model then captures all interconnections between continuous plant dynamics and logic-based control rules. Once the model is available, the reachability problem is formulated as a mixed-integer linear program (MILP) with a pure feasibility objective.

The paper is structured as follows. First we introduce basic notion of hybrid systems and review most popular mathematical abstractions of such systems. Then, in Section 3 we describe the translation process in details. Reachability problems are then formulated in Section 4 and illustrated on a concrete example in Section 5. The paper is wrapped up by concluding remarks.

2. HYBRID SYSTEMS

Hybrid systems represent a compact framework which captures behavior of systems where continuous dynamics is coupled with discrete logic. Examples include, but are not limited to, systems with discrete-valued actuators (such as on/off switches), piecewise linear nonlinearities, and finite state machines. Mathematically, hybrid systems can be described by Piecewise Affine (PWA) models (Sontag, 1981), Mixed Logical Dynamical (MLD) systems (Bemporad and Morari, 1999), Linear Complementarity systems (Heemels et al., 2000) and max-min-plus-scaling models (De Schutter and Van den Boom, 2001). Under mild assumptions, all these frameworks are equivalent to each other (Heemels et al., 2001). In the sequel we review PWA and MLD approaches to modeling of hybrid systems. Since the aim of the paper is on verifying safety properties of closed-loop systems where the plant is governed by a set of internal IF-THEN-ELSE rules, only autonomous systems are considered.

2.1 Piecewise Affine (PWA) Systems

Autonomous PWA systems are defined by partitioning the space into polyhedral regions,and associating each region with a different linear (or affine) state-update equation:

$$x(k+1) = \begin{cases} A_1 x(k) + f_1 \text{ if } x(k) \in \mathcal{R}_1 \\ \vdots \\ A_n x(k) + f_n \text{ if } x(k) \in \mathcal{R}_n. \end{cases} \tag{1}$$

Here, $x(k) \in \mathbb{R}^{n_x}$ is the state vector at time instance k, $x(k+1)$ is the successor state at the next sampling instance, $\mathcal{R}_i \subseteq \mathbb{R}^{n_x}$, $i = 1, \ldots, n$ are polyhedral regions of the joint state-input space, and n is the number of individual affine dynamics. PWA systems arise naturally when nonlinear plants are approximated by the technique of multiple linearizations.

2.2 Mixed Logical Dynamical (MLD) Systems

MLD systems represent systems governed by discrete logic by a system of linear inequalities involving binary variables, which can be derived using so-called *big-M* formulation (Williams, 1993). To illustrate the procedure, consider a logic statement of the following form

$$\delta = \begin{cases} 1 \text{ if } a^T x \leq b \\ 0 \text{ if otherwise} \end{cases} \tag{2}$$

which connects the truth value of binary variable δ to satisfaction of the linear inequality $a^T x \leq b$ (which involves a real-valued variable $x \in \mathbb{R}^{n_x}$) via a logic equivalence relation. Let M and m denote, respectively, the maximum and minimum values which the linear expression $a^T x - b$ attains over the domain $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, i.e.

$$M = \max_{x \in \mathcal{X}} a^T x - b, \tag{3a}$$

$$m = \min_{x \in \mathcal{X}} a^T x - b. \tag{3b}$$

Then the IF-THEN-ELSE rule (2) is equivalent to satisfaction of the following system of linear inequalities:

$$a^T x - b \leq M(1 - \delta), \tag{4a}$$

$$a^T x - b \geq \epsilon + m\delta. \tag{4b}$$

Here, ϵ is a small constant, typically the machine precision, used to convert a strict inequality into a non-strict form. More complex logic expressions involving e.g. one-way implications ($\Leftarrow$ or $\Rightarrow$) and logic operations (and, or, negation) can be translated in a similar fashion, see e.g. (Williams, 1993; Bemporad and Morari, 1999).

In the most general form, autonomous MLD systems are described by

$$x(k+1) = Ax(k) + B_\delta \delta(k) + B_z z(k) + B_0, \tag{5a}$$

$$E_x x(k) + E_\delta \delta(k) + E_z z(k) \leq E_0, \tag{5b}$$

where $x \in \mathbb{R}^{n_x}$ is the vector of states, $\delta \in \{0, 1\}^{n_\delta}$ is the vector of binary variables, $z \in \mathbb{R}^{n_z}$ is the vector of auxiliary real variables, and A, B_δ, B_z, B_0, E_x, E_δ, E_z, E_0 are matrices (or vectors) of appropriate dimensions. Given a value of $x(k)$, the state update $x(k+1)$ can be computed by solving a feasibility problem, i.e. by finding a compatible combination of binary $\delta(k)$ and real $z(k)$ variables which satisfy constraints (5b).

2.3 HYSDEL

Modeling of hybrid systems involves finding parameters of the corresponding mathematical model. In the PWA case (1), this boils down to finding matrices A_i, B_i, and the regions $\mathcal{R}_i$. In the MLD case (5), one needs to apply the big-M procedure to find matrices A, B_δ, B_z, B_0, E_x, E_δ, E_z, and E_0. Clearly, as the system to be described becomes more complex, such a “manual” approach to modeling can become cumbersome and error prone.

To accelerate development of hybrid models, HYSDEL (Hybrid Systems Description Language) was developed (Torrisi, 2002). It features a high-level modeling language which allows to describe behavior of hybrid systems using a custom syntax, which is similar to the C language. Once the system is described, the HYSDEL compiler parses the model and converts it into the MLD description (5) using the big-M technique. The MLD model can subsequently be converted into the PWA form.

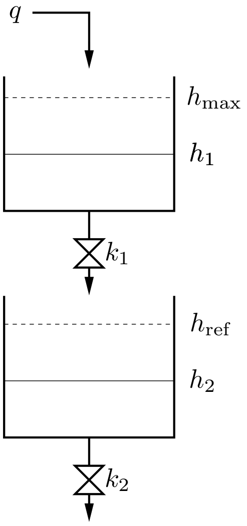


Fig. 1. The two-tanks arrangement.

Although HYSDEL significantly simplifies synthesis of mathematical representations of hybrid systems, it requires the user to learn its syntax. Therefore it is not directly applicable to verify control algorithms written in standard languages, such as in MATLAB or in C. To bridge this gap between standard control engineering tools and HYSDEL, we have developed a novel tool which automatically translates a MATLAB code to a corresponding HYSDEL model.

3. THE MATLAB-TO-HYSDEL TRANSLATOR

This section describes the process of translating a control algorithm written in MATLAB into the HYSDEL form. The translator consists of a *lexer*, which cleans the MATLAB code and identifies its key components. Operation of this phase is reported as Algorithm 1. Following components are being identified:

- operators: +, -, *, ^, /, &, |, ~, =, >, <
- keywords: `if`, `else`, `end`, `function`, `global`
- names of variables

The cleaned-up code is subsequently processed by a *parser*, which operates according to Algorithm 2. The parser first creates declarations of state, input and output variables, which serve as an interface between the control algorithm and the outside world. Subsequently, each line of the MATLAB source file is converted into its HYSDEL equivalent. Since HYSDEL only supports linear expressions, only a subset of valid MATLAB expressions can be converted.

To illustrate the translation, consider the following example. Given is a system composed of two liquid tanks, situated above each other, as shown in Figure 1. The linearized mathematical model of such a system is given by

$$h_1(k+\Delta_t) = h_1(k) + {}^{\Delta_t}\!/_{F_1}(q(k) - s_1 h_1(k)), \tag{6a}$$

$$h_2(k+\Delta_t) = h_2(k) + {}^{\Delta_t}\!/_{F_2}(s_1 h_1(k) - s_2 h_2(k)), \tag{6b}$$

where h_1 and h_2 are the liquid levels in the corresponding tanks, Δ_t is the sampling period, F_1, F_2 are the tanks' cross-section areas, $q(k)$ is the liquid inflow to the first tank (which is the control input), and s_1 and s_2 are linearization coefficients. The system is to be controlled by a rule-based controller:

$$q(k) = \begin{cases} q_{\max} \text{ if } h_2(k) \leq h_{\text{ref}} \text{ and } h_1(k) \leq h_{\max} \\ 0 \text{ otherwise.} \end{cases} \tag{7}$$

The rules are such that the liquid inflow is set to a non-zero value $q_{\max}$ whenever the liquid level in the bottom tank is below its reference h_{ref} and the upper tank is not overflowing. Otherwise the control input is set to zero. Important to notice is that due to accumulation of the liquid in the upper tank, liquid in the lower tank may continue to rise even after the control input is set to zero. In the next section we will show how to verify suitability of such a control scheme (i.e. that it guarantees a safe operation of the equipment where none of the tanks will overflow) by employing reachability analysis.

To illustrate the automatic code translator, suppose that the closed-loop system is described by the following MATLAB code:

```
function closed_loop

% declaration of internal states
global h1 h2

% declaration of parameters
F1 = 31.8319;
F2 = 31.8319;
s1 = 1;
s2 = 0.9;
dT = 5;
href = 76;  % reference level in centimeters
hmax = 100; % safety limit for the upper tank
qmax = 100; % default flow rate to upper tank

% control rules
if ( h2 <= href ) & ( h1 <= hmax )
 q = qmax;
else
 q = 0;
end

% dynamical system
h1 = h1 + dT/F1*( q - s1*h1 );
h2 = h2 + dT/F2*( s1*h1 - s2*h2 );
```

The code supported by the translator is structured as follows. First line always contains definition of the MATLAB function. State variables are represented as global variables, since they constitute an internal storage which needs to be updated between consecutive executions of the code. Concrete numerical values of parameters are provided next, followed by definition of switching control rules (7). The rules can contains logic operators such as `and` (&), `or` (|), and negation (∼). Multiple rules can be used and they can be interconnected using ELSEIF statements. The computed control action (denoted by the `q` variable in the code), is then used to update the internal state variables according to (6).

Applying the translator to such a MATLAB code produces its HYSDEL equivalent, reported next.

```
SYSTEM closed_loop {
 INTERFACE {
   STATE {
     REAL h1, h2;
   }
   PARAMETER {
   REAL F1 = 31.8319, F2 = 31.8319;
   REAL s1 = 1, s2 = 0.9;
   REAL dT = 5, href = 76;
   REAL hmax = 100, qmax = 100;
   }
 }
 IMPLEMENTATION {
   AUX {
     REAL q;
     BOOL delta1, delta2;
   }
   AD {
     delta1 = (h2 <= href);
     delta2 = (h1 <= hmax);
   }
   DA {
     q = {IF (delta1 & delta2) THEN qmax ELSE 0};
   }
   CONTINUOUS {
     h1 = h1 + dT/F1*( q - s1*h1 );
     h2 = h2 + dT/F2*( s1*h1 - s2*h2 );
   }
 }
}
```

Applying the HYSDEL compiler to the generated model, matrices of the MLD model (5) will be generated and saved to MATLAB. The MLD model can be subsequently converted to the PWA model (1) e.g. by using the `mpt_sys` function of the Multi-Parametric Toolbox (Kvasnica et al., 2004). The translator implements Algorithms 1 and 2 in the PHP language and is provided as a free web-based service available at `http://necron.sk/xant/`. Notice that the translator is under an active development and is subject to frequent changes in the following months.

4. SAFETY VERIFICATION VIA REACHABILITY ANALYSIS

To verify safety properties of closed-loop systems described as hybrid systems, one can solve the following problem.

Problem 4.1. Given is a hybrid system either in PWA or MLD form, a polyhedral set of initial conditions $\mathcal{X}_0$, a time horizon T, and a polyhedral set of "unsafe" states $\mathcal{X}_f$. Find an initial condition $x(0) \in \mathcal{X}_0$ for which the evolution of states reaches $\mathcal{X}_f$ in, at most, T steps, or determine that no such initial condition exists.

A feasible solution to Problem 4.1 constitutes at least one "unsafe" initial condition for which the control rules fail to meet a given safety goal. Infeasibility of Problem 4.1, on the other hand, provides a certificate that the system will always evolve in a safe manner.

Remark 4.2. Problem 4.1 can be easily extended to cover cases where the set of "unsafe" states $\mathcal{X}_f$ is a non-convex set represented by a finite number of polyhedra. Moreover, instead of verifying safety with respect to a fixed horizon

Algorithm 1 Lexer algorithm

```
INPUT: MATLAB code
OUTPUT: Cleaned MATLAB code with identified tokens
take MATLAB code as string
for each operator OR keyword in string do
    if operator then
        put space before and after operator
    end if
    if keyword then
        put semicolon before and after keyword
    end if
end for
for each character in string do
    if it is TAB then
        replace it with space
    end if
    if it is carriage return OR line feed then
        replace it with semicolon
    end if
    if it is per cent sign then
        while it is NOT (semicolon OR carriage return OR line feed) do
            shift to the next character
        end while
    end if
    if it is space as previous character then
        shift to the next character
    end if
    if it is semicolon as previous character then
        shift to the next character
    end if
end for
for each semicolon do
    break the rest of the string into new line
end for
```

Algorithm 2 Parser algorithm

```
INPUT: MATLAB tokens
OUTPUT: HYSDEL code
declare state, input and output variables
for each line do
    if "if" found then
        parse next line
        store condition(s)
        parse next line
        store "if" value
        parse next line
        if "else" found then
            store "else" value
            parse next line
        end if
    else
        store parameter
    end if
end for
for each condition do
    store auxiliary variable
    negate condition
    if negated condition found then
        store negated aux var
    end if
end for
create HYSDEL pattern
fill pattern with stored strings
generate HYSDEL code
```

T, one can look for the minimal value of T for which the system violates safety conditions. This can be achieved e.g. by employing bisection in conjunction with Problem 4.1.

If the hybrid system to be verified is given in the MLD form (5), Problem 4.1 can be approached by solving a feasibility mixed-integer linear program:

$$
\begin{aligned}
&\text{find } x(0) && \text{(8a)}\\
&\text{s.t. } x(0) \in \mathcal{X}_0, && \text{(8b)}\\
&\quad x(k+1) = Ax(k) + B_\delta \delta(k) + B_z z(k) + B_0, && \text{(8c)}\\
&\quad E_x x(k) + E_\delta \delta(k) + E_z z(k) \leq E_0, && \text{(8d)}\\
&\quad x(T) \in \mathcal{X}_f. && \text{(8e)}
\end{aligned}
$$

Here, constraints (8c)–(8d), which are defined for $k = 0, \ldots, T-1$, describe evolution of the MLD system on horizon T, cf. (5). Under the assumption that $\mathcal{X}_0$ and $\mathcal{X}_f$ are polyhedral sets, they can be described by

$$
\begin{aligned}
\mathcal{X}_0 &= \{x \mid H_0 x \leq K_0\}, && \text{(9a)}\\
\mathcal{X}_f &= \{x \mid H_f x \leq K_f\}, && \text{(9b)}
\end{aligned}
$$

where H_0, H_f, K_0, K_f are matrices which represent the half-space representation of such sets. Therefore all constraints in (8) are linear in the decision variables $x(k)$, $\delta(k)$, and $z(k)$, for $k = 0, \ldots, T$. Since $\delta(k)$ are vectors of binary variables while $x(k)$ and $z(k)$ are real-valued vectors, it follows that problem (8) is a mixed-integer linear program (MILP) with a pure feasibility objective. Such MILP problems can be formulated e.g. by YALMIP (Löfberg, 2004) and solved efficiently using state-of-the-art solvers, such as with GLPK (Makhorin, 2001) or CPLEX (ILOG, Inc., 2003).

If the hybrid system is given in its PWA form (1), the corresponding reachability problem is formulated as follows

$$
\begin{aligned}
&\text{find } x(0) && \text{(10a)}\\
&\text{s.t. } x(0) \in \mathcal{X}_0, && \text{(10b)}\\
&\quad \delta_i(k) \Leftrightarrow x(k) \in \mathcal{R}_i,\ i = 1, \ldots, n, && \text{(10c)}\\
&\quad \delta_i(k) \Leftrightarrow x(k+1) = A_i x(k) + f_i,\ i = 1, \ldots, n, && \text{(10d)}\\
&\quad \sum_{i=1}^{n} \delta_i(k) = 1, && \text{(10e)}\\
&\quad x(T) \in \mathcal{X}_f. && \text{(10f)}
\end{aligned}
$$

Here, $\delta_i(k)$, $i = 1, \ldots, n$ (where n is the number of PWA regions) are binary selectors which take the value of 1 if and only if the state $x(k)$ is contained in the i-th polyhedral region $\mathcal{R}_i$, cf. (10c). The truth value of the corresponding binary selector then activates a particular state-update equation in (10d). The logic equivalence ($\Leftrightarrow$) rules can again be translated into mixed-integer inequalities using the big-M method, as shown in Section 2.2. Finally, (10e) is an exclusive-or condition which only allows the state to reside in a single polyhedral region. Again, problem (10) can be readily cast as a feasibility MILP with binary variables $\delta_i(k)$, and real variables $x(k)$, $k = 0, \ldots, T$.

Due to equivalence between PWA and MLD systems (Heemels et al., 2001), Problem 4.1 can be answered either by solving (8) or (10). The particular selection depends on the number of binary variables induced by a particular choice of the hybrid model (either PWA or MLD). In the next section we apply the described safety verification procedure to an illustrative example.

5. EXAMPLE

We revisit the two-tanks example discussed in the previous section. We reiterate that the aim is to verify that the rules-based controller (7), connected with the system (6), always meets a certain safety goal. In this example we want to verify that the liquid level in the bottom tank (h_2) stays below a pre-defined threshold, say h_{unsafe}. The set of unsafe states is hence $\mathcal{X}_f = \{h_2 \mid h_2 > h_{\text{unsafe}}\}$ with $h_{\text{unsafe}} = 84$ (all levels are expressed in centimeters). The set of initial states is $\mathcal{X}_0 = \{h_1, h_2 \mid 0 \leq h_1 \leq a,\ 0 \leq h_2 \leq a\}$. Two scenarios were considered: one with $a = 30$ and the other one with $a = 70$.

To formulate the verification problem, the HYSDEL model of the closed-loop system is first compiled by the HYSDEL compiler, which generates the corresponding MLD model description. This model was subsequently used to formulate the MILP problem (8) using YALMIP (Löfberg, 2004):

```
 1 % definition of decision variables
 2 x = {}; d = {}; z = {};
 3 for k = 1:T+1
 4  x{k} = sdpvar(nx, 1);
 5  z{k} = sdpvar(nz, 1);
 6  d{k} = binvar(nd, 1);
 7 end
 8
 9 % constraint on the initial condition
10 CON = [ 0 <= x{1}(1) <= a; 0 <= x{1}(2) <= a ];
11
12 % constraint on the final state in the unsafe set
13 CON = CON + [ x{end}(2) > 84 ];
14
15 % time evolution of the MLD model
16 for k = 1:T
17  CON = CON + [ x{k+1} == A*x{k}+Bd*d{k}+Bz*z{k}+B0 ];
18  CON = CON + [ Ex*x{k} + Ed*d{k} + Ez*z{k} <= E0 ];
19 end
20
21 % solve the fesibility problem
22 solution = solvesdp(CON, []);
23
24 % return the "unsafe" initial condition if it exists
25 if solution.problem == 0
26  xunsafe = double(x{1});
27 end
```

Here, the decision variables $x(t)$, $\delta(t)$, $z(t)$ are first defined on lines 2–7 for $t = 0, \ldots, T$. Notice that variables $\delta(t)$ are declared as binary variables on Line 6. Line 10 then specifies the set of initial conditions for both state variables (the index in $\{\cdot\}$ denotes the time step k, while the index in $(\cdot)$ specifies position of the particular element in the state vector). Similarly, the set of unsafe states is defined on Line 13. Then, constraints (8c)–(8d) are repeated for $k = 0, \ldots, T-1$ on Lines 17 and 18. Finally, the formulated verification problem is solved by calling the `solvesdp` command. If the problem is feasible for some value of T, (cf. Lines 25–27), value of the “unsafe” initial condition is returned. If the problem is infeasible for all $T \leq T_{\max}$, then there is no such unsafe starting point and therefore the controller always operates in a safe manner within of $T_{\max}$ time steps.

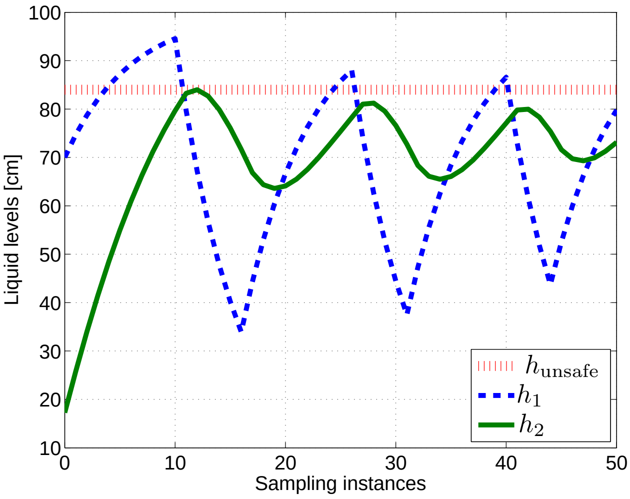


Fig. 2. Simulation scenario for an unsafe initial condition.

For $a = 30$ we have solved problem (8) for $T = 0, \ldots, 50$ (which corresponds to 250 seconds) using the GLPK MILP solver. The problem was infeasible for any value of $T \leq T_{\max}$, which certifies a safe behavior of the control system (7) for any initial condition bounded by $0 \leq h_i \leq a$, $i = 1, 2$. However, for $a = 70$ the safety verification problem was feasible for $T = 12$, which resulted into the unsafe initial condition $h_1(0) = 70$ and $h_2(0) = 17.2606$. Simulation of the closed-loop system starting from this initial condition is shown in Figure 2, which indeed confirms that the safety barrier $h_2 > 84$ is violated after 12 sampling instances.

6. CONCLUSIONS

In this paper we have proposed how to verify safety properties of logic-based control laws written in MATLAB. First, the MATLAB code was converted into its HYSDEL equivalent by means of an automated lexing and parsing procedure. The HYSDEL model was subsequently converted into a mathematical form, represented either by a PWA or by an MLD model. Finally, the verification was performed by solving a mixed-integer linear program. A motivating example was provided to illustrate individual steps. The main benefit of this work is that it allows theoretical verification algorithms to be applied to a subset of ordinary computer code, in this case represented by MATLAB. However, the translator can be easily modified to support other programming languages as well, for instance the C language or Java.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07.

REFERENCES

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

De Schutter, B. and Van den Boom, T. (2001). On model predictive control for max-min-plus-scaling discrete event systems. *Automatica*, 37(7), 1049–1056.

Heemels, W.P.M., De Schutter, B., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.

Heemels, W., Schumacher, J., and Weiland, S. (2000). Linear complementarity systems. *SIAM Journal on Applied Mathematics*, 60(4), 1234–1269.

ILOG, Inc. (2003). *CPLEX User Manual*. Gentilly Cedex, France. http://www.ilog.fr/products/cplex/.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/~mpt/.

Löfberg, J. (2004). YALMIP. Available from http://users.isy.liu.se/johanl/yalmip/.

Lygeros, J., Tomlin, C., and Sastry, S. (1999). Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3), 349–370.

Makhorin, A. (2001). *GLPK - GNU Linear Programming Kit*. http://www.gnu.org/directory/libs/glpk.html.

Sontag, E.D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2), 346–358.

Torrisi, F.D. (2002). Hybrid System DEscription Language (HYSDEL). Available from http://control.ee.ethz.ch/~hybrid/hysdel/.

Torrisi, F. (2003). *Modeling and Reach-Set Computation for Analysis and Optimal Control of Discrete Hybrid Automata*. Ph.D. thesis, ETH Zurich.

Williams, H. (1993). *Model Building in Mathematical Programming*. John Wiley & Sons, Third Edition.

The Empirical Mode Decomposition in Real-Time

P. Trnka* M. Hofreiter**

** Czech Technical University in Prague, Faculty of Mechanical Engineering,
Department of Instrumentation and Control Engineering
(Tel: +420 602 286 781; e-mail: Pavel.Trnka@fs.cvut.cz)
** Czech Technical University in Prague, Faculty of Mechanical Engineering,
Department of Instrumentation and Control Engineering
(Tel: +420 22435 2566; e-mail: Milan.Hofreiter@fs.cvut.cz)*

Abstract: The paper devotes analysis of environmental time series by using the on-line empirical mode decomposition (OEMD). The environmental data were measured by meteorological stations which are deployed in the southern part of Czech Republic. The EMD algorithm was modified for the possibility of the on-line analysis of environmental time series.

1. INTRODUCTION

During processing of environmental data to consider the state of the ecosystem, the stationarity or periodicity of the measured data is usually assumed. In fact, the observed data reflect the characteristics of the ecosystem, which is generally nonlinear, stochastic and nonstationary. The results obtained, given the very simplistic assumptions might therefore lead to incorrect conclusions and to obtain distorted characteristics either in time or frequency domain. Since similar problems encountered at each analyzing nonstationary stochastic systems, the EMD (Empirical Mode Decomposition) algorithm, developed by N. E. Huang in 1998 for NASA (Huang, et al. 1998), attracted much attention. Huang combined EMD algorithm with the algorithm for the Hilbert spectral analysis and created the so-called Hilbert-Huang Transformation (HHT), which is applicable for analysis of nonlinear, stochastic and nonstationary processes.

2. EMPIRICAL MODE DECOMPOSITION

An EMD algorithm decomposes adaptively the signal $x(t)$ into intrinsic mode functions $c_i(t)$, i = 1, 2, …, n and into residue $r(t)$:

$$x(t) = \sum_{i=1}^{n} c_i(t) + r(t), \tag{1}$$

where n means the number of IMF functions. Residue $r(t)$ reflects the average trend of a signal $x(t)$ or a constant value. Intrinsic mode functions (IMF) are signals with following characteristics:

In the whole dataset, the number of extremes (minima and maxima) and the number of zero-crossings must either equal or must differ by a maximum of one.

Each point, that is defined as mean value of envelopes defined by local maxima and local minima is zero.

The algorithm for searching of intrinsic mode functions is based on a procedure called “sifting”, described e.g. in (Zhaohua 2009) and (Zhaohua 2010).

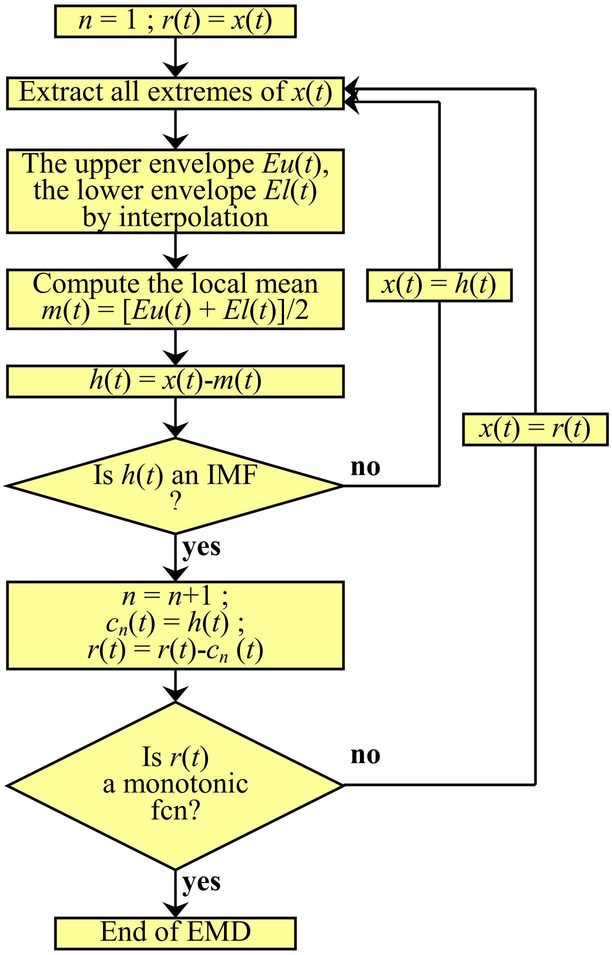

n = 1 ; $r(t)$ = $x(t)$
Extract all extremes of $x(t)$
The upper envelope $Eu(t)$, the lower envelope $El(t)$ by interpolation
Compute the local mean $m(t)$ = [$Eu(t)$ + $El(t)$]/2
$h(t)$ = $x(t)$-$m(t)$
Is $h(t)$ an IMF ?
no
$x(t)$ = $h(t)$
yes
n = n+1 ; $c_n(t)$ = $h(t)$; $r(t)$ = $r(t)$-c_n (t)
Is $r(t)$ a monotonic fcn?
no
$x(t)$ = $r(t)$
yes
End of EMD

Fig 1.Flowchart of Empirical Mode Decomposition algorithm

The algorithm proceeds in the following steps (see Fig. 1):

1. Create upper envelope $E_u(t)$ by local maxima and lower envelope $E_l(t)$ by local minima of data $x(t)$.

2. Calculate the mean of upper and lower envelope
$$m_1(t)=\frac{E_{u1}(t)+E_{l1}(t)}{2}. \tag{2}$$

3. Subtract the mean from original data
$$h_1(t)=x(t)-m_1(t). \tag{3}$$

4. Verify that $h_1(t)$ satisfies conditions for IMFs. Repeat steps 1 to 4 with $h_1(t)$, until it is an IMF.

5. Get first IMF (after k iterations)
$$c_1(t)=h_{1(k-1)}(t)-m_{1k}(t). \tag{4}$$

6. Calculate first residue
$$r_1(t)=x(t)-c_1(t) \tag{5}$$

7. Repeat whole algorithm with $r_1(t)$, $r_2(t)$, … until residue is monotonic function.

8. After n iterations $x(t)$ is decomposed according to equation (1).

3. ON-LINE ANALYSIS

The algorithm described in chapter 2. is calculated off-line over the entire measured data range. Since the dataset can be very large, the sifting process can be time-consuming and computationally very demanding. Therefore, an algorithm that processes the data gradually, by moving time windows, was created. The advantage of floating time windows is mainly significant in accelerating of decomposition into IMF functions.

For off-line decomposition, the computation time is not so much restricted and the interval edges, that might be distorted, can be omitted from the analysis. But these two problems become serious, when the EMD algorithm for on-line analysis is used. The following procedure was used to overcome these problems:

- The floating time window is created. The window range:
$$\langle (t-T_w);t\rangle, \tag{6}$$
where t is actual time point and a T_w is length of window. The decomposition process is at any point of time evaluated only in the appropriate window range. So the time needed for compute IMFs does not grow with simulation time.

- To reduce distortion of decomposition at the end of measurement interval, the currently known courses of IMFs are used to estimate their future course and to estimate their future local extremes needed for EMD algorithm.

The size of the time window affects frequencies, which ones will be detected during sifting in the IMFs and which ones will be included within residue. Generally, the longer the window, the lower frequencies (and therefore the longer periods) will appear in IMFs.

The practical realization of algorithm is implemented in the software MATLAB and its simulation toolbox SIMULINK. As a basis for the algorithm is used modified Zhaohua MATLAB function „eemd()“ – see (Zhaohua 2010). The modification consists in replacing the interpolation functions for creating of envelopes with functions that allow also interpolation. The modified eemd() function is built into the s-function „s-emd()“. The main tasks of the s-function are:

- To create and to maintain moving time window during simulation.

- To maintain dynamic global variables of type one-, two- and three-dimensional array of variable length. These variables store the source data, residue and IMFs with their variances.

4. IMPLEMENTATION

Modified On-line Empirical Mode Decomposition (OEMD) algorithm described in chapter 3 is demonstrated on soil temperature measurements from the meteorological station by the environmental project TOKENELEK. (Fig. 2) shows the temperature ϑ during September 2010, the sampling period is 10 minutes.

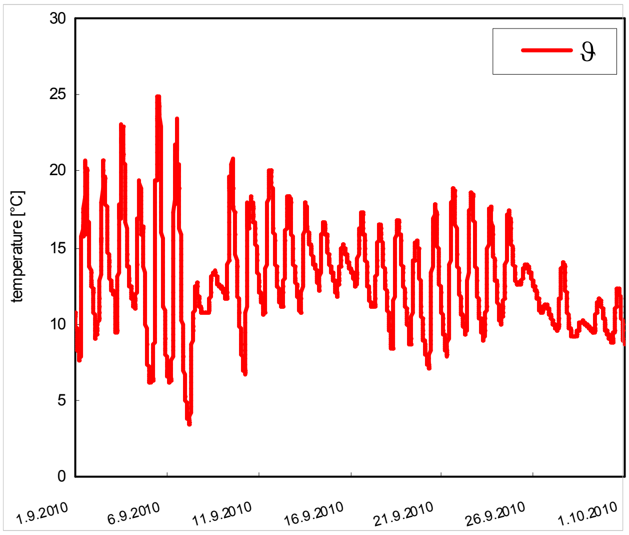


Fig. 2 The time range of temperature ϑ.

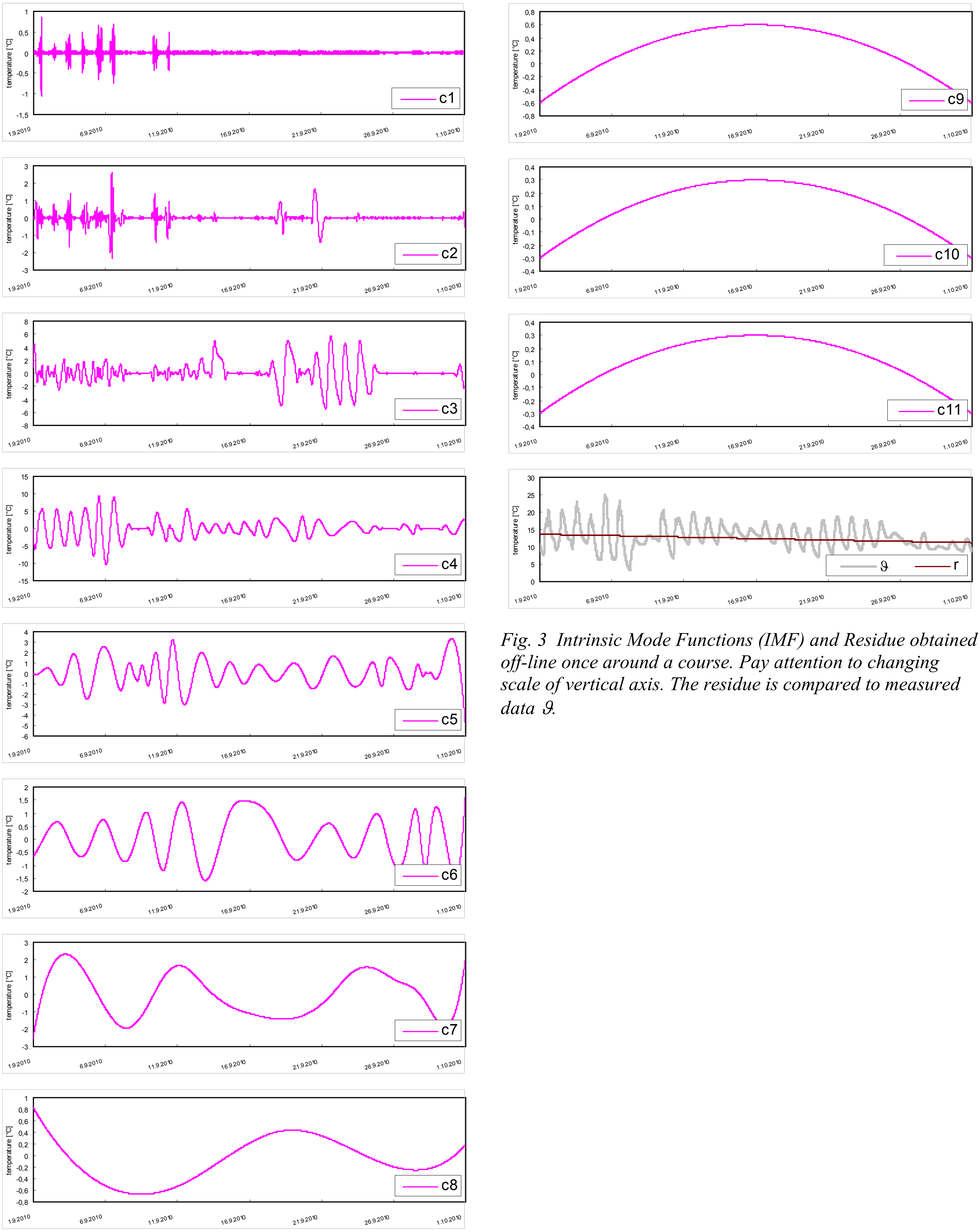


Fig. 3 Intrinsic Mode Functions (IMF) and Residue obtained off-line once around a course. Pay attention to changing scale of vertical axis. The residue is compared to measured data ϑ.

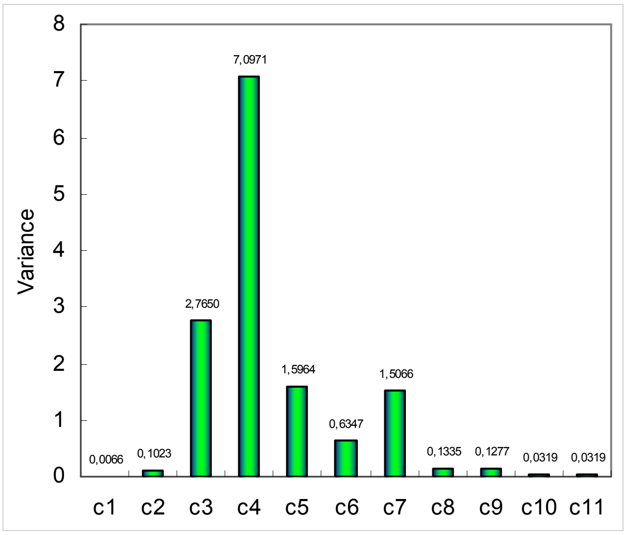


Fig. 4 Comparison of variances of (off-line) IMF. The larger variance, the more important component it represents.

The experimental analysis has been divided into two parts. An off-line EMD analysis was carried out in the first part. The resulting functions are shown in the graph (Fig. 3) and serve as reference samples of IMF functions. Variance was also found for every IMF. Variances were used as a simple benchmark to determine how significant component of the initial data each IMF represents. As shown in (Fig. 4), the most important component is c_4.

(Fig. 5) compares original data ϑ with filtered temperature ϑ_f, that is declared by formula:

$$\vartheta_f(t) = \sum_{i=2}^{9} c_i(t) + r(t). \qquad (6)$$

The same source data (Fig. 2) has been processed by on-line EMD algorithm in the second part of the experiment. The size of the time window has been set at 1008, which given the sampling density represented 7 days.

The rectangle shown in graph (Fig. 6) represents the time window during the simulation. One set of IMF functions that has been generated inside the time window is shown in the graph (Fig. 7). It is obvious that the residue is more curved in comparison to off-line decomposition and also some IMFs have higher variance. This phenomenon is a necessary and expected consequence of a shorter time range of analyzed data.

Along with finding the IMF functions their variance were estimated. On the basis of the variances it was decided which IMFs will be included into partial filtered function $\vartheta_f(t)$.

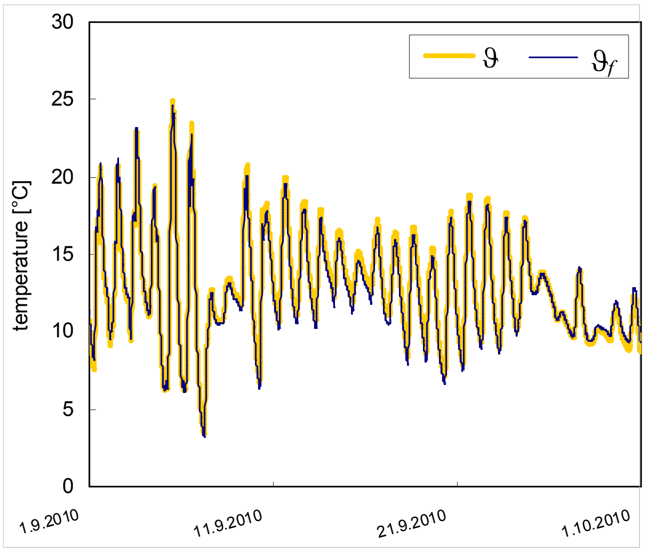


Fig. 5 Comparison of temperature trends ϑ and ϑ_f. The thick line represents the original measured data; the thin line represents the filtered function obtained by adding the selected major IMFs and residue.

(Fig. 8) shows the course of several partial filtered functions compared to the original source data. Thick light line represents the source data; thin gray scaled lines show partial filtered functions from selected time windows.

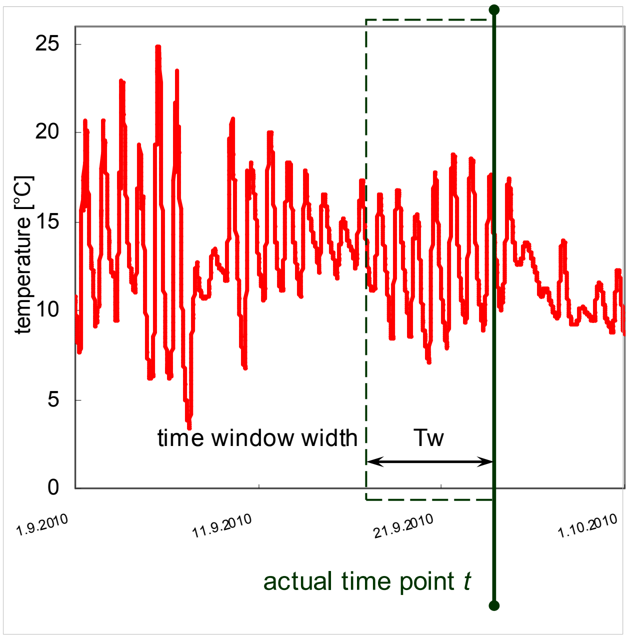


Fig. 6 The symbolic representation of moving time window.

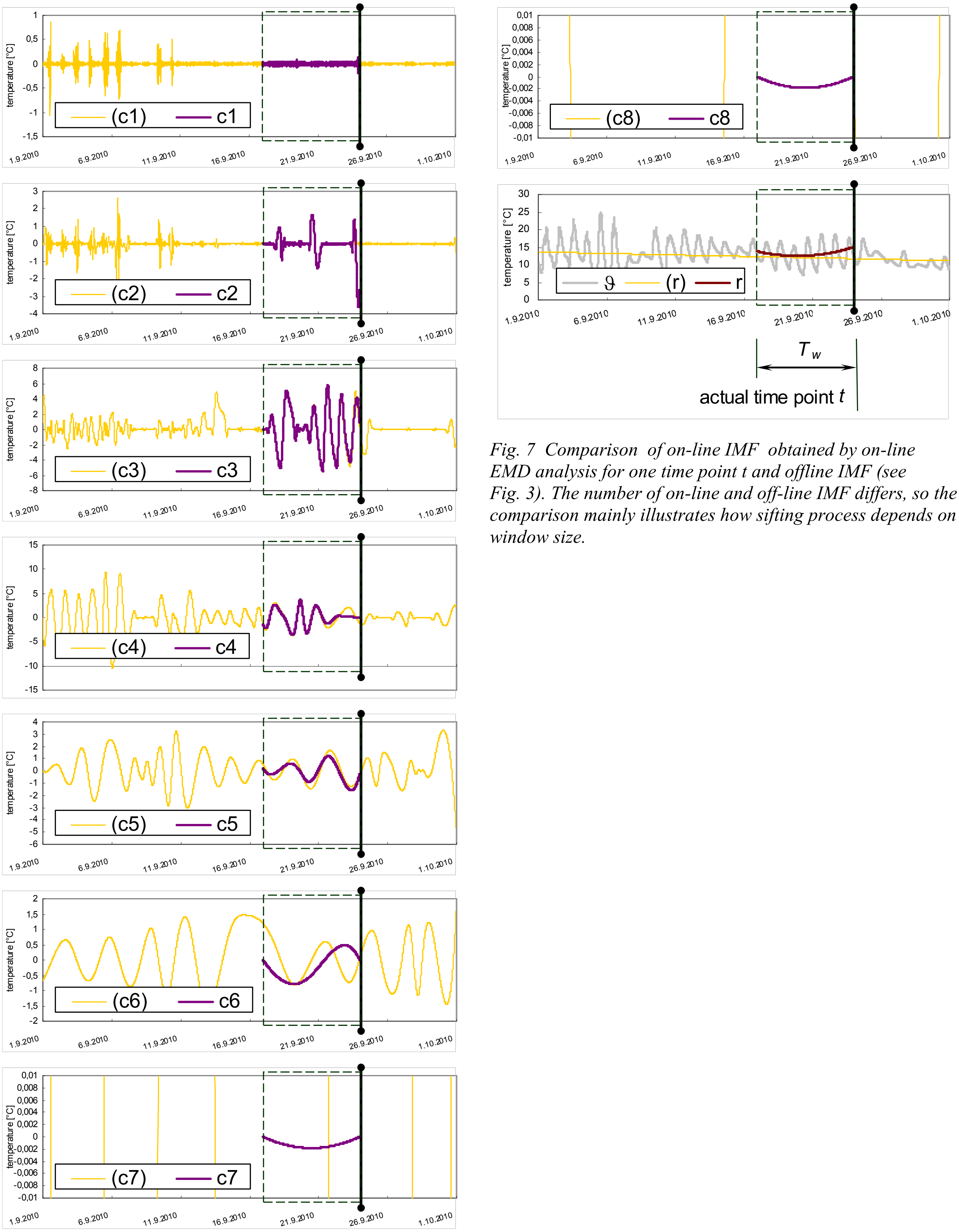


Fig. 7 Comparison of on-line IMF obtained by on-line EMD analysis for one time point t and offline IMF (see Fig. 3). The number of on-line and off-line IMF differs, so the comparison mainly illustrates how sifting process depends on window size.

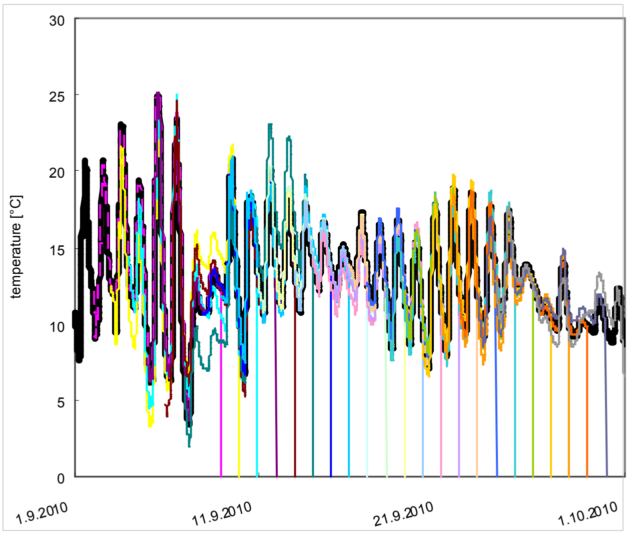


Fig. 8 Comparison of measured temperature ϑ and (on-line) partial filtered functions ϑ_{fi}. The measured temperature is represented by thick light line; the courses of filtered temperatures for selected time windows are represented by thin lines with varying greyscales. The intervals between displayed partial functions are 1 day; the length of time window is T_w=7 days.

5. CONCLUSION

Empirical Mode Decomposition (EMD) is a progressive method that combines the signal analysis in time even in frequency domain. In field of environmental non-stationary data streams, almost repetitive sequences with very various periodicities often appear, but exact repetition of events occurs rarely. For this reason, the EMD analysis of these systems is very convenient. Real-time data processing provides results qualitatively similar to the offline analysis. Comparing of both methods shows that on-line analysis with a moving time window is much faster and significantly reduces the computational complexity. Off-line analysis on the contrary provides a slightly more detailed decomposition because it captures even very low frequencies. Data obtained from the experiments provided a number of suggestions for further work, with a wide range stems mainly from the short-term prediction of the internal mode functions for more precise decomposition in real time.

ACKNOWLEDGMENTS

This work has been supported by the
Grant Agency of the Czech Technical University in Prague, grant No. SGS10/252/OHK2/3T/12
and by the National development program in project TOKENELEK, grant No. 2B06023.

REFERENCES

Huang, et al. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc. R. Soc. Lond. A (1998) 454, p. 903–995, [cit. 2010-10-06], Online: <http://keck.ucsf.edu/~schenk/Huang_etal98.pdf>

Huang, N. E., Shen, Z., Long, R. S. (1999). A New View of Nonlinear Water Waves—The Hilbert Spectrum. In: Ann. Rev. Fluid Mech. 31, p. 417-457.

Kokeš, Josef (2009). Application of Hilbert-Huang Transform in an Expert System. In: Nové metody a postupy v oblasti přístrojové techniky, automatického řízení a informatiky: odborný seminář. Jindřichův Hradec: Ústav přístrojové a řídicí techniky ČVUT v Praze. (in Czech language)

Zhaohua, W., Huang, N. E. (2009). Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method. In: Advances in Adaptive Data Analysis, Vol. 1, No. 1, p. 1–41, World Scientific Publishing Company.

Zhaohua, Wu (2010). HHT MATLAB Program. [cit. 2010-10-06],
Online:
<http://rcada.ncu.edu.tw/research1_clip_program.htm>.

Hofreiter, M. (2010). The Application of Hilbert-Huang Transform to Non-Stationary Environmental Data Sets. In: *TMT 2010*. Zenica: Faculty of Mechanical Engineering in Zenica, 2010, p. 309-312. ISSN 1840-4944.

CONTROL OF THE LABORATORY HELICOPTER SIMULATOR

J. Macháček, L. Havlíček

University of Pardubice, Faculty of Electrical Engineering and Informatics
Department of Process Control, Nám. Čs. Legií 565, 532 10 Pardubice
Phone +420 466 037 122, e-mail jiri.machacek@upce.cz

Abstract: The laboratory helicopter simulator is a nonlinear two inputs - two outputs system with significant cross-coupling. The papers deals with control of vertical angle, where the controlled variable was the position angle and the manipulated variable was main motor voltage. The methods of control with PID controller, IMC controller and self-tuning adaptive controller were designed and tested. The control algorithms were used for tracking of the reference values and rejection of disturbances (moving of the tail rotor).

Keywords: Helicopter simulator, PID controller, IMC controller, self-tuning controller

1 INTRODUCTION

The real laboratory physical models are significant part of control engineering education. Control of real physical models opens many problems, which are hidden by computer simulation (static and dynamic properties sensors or actuators, immeasurable disturbances, hardware and software tools for the connection between system and computer, problems with sampling by real-time experiments, etc). Those real physical models are then closer to real industrial systems.

Laboratory system with twin rotor (helicopter simulator) is often used for laboratory education. This laboratory equipment produce *e.g.* firms Humusoft, Feedback or Bytronic. The system described in this paper was designed and realized in the Department of Process Control, University of Pardubice.

The helicopter is a nonlinear two inputs - two outputs system with significant cross-coupling. Many tasks may be realized on this system from the easy measurement of both static and dynamic characteristics to the multivariable control.

The aim of this paper is to give some suggestions for simple student's laboratory tasks on the one-dimensional control of vertical angle with fixed horizontal position. The paper includes the measurement of both step and frequency responses, the model building from identification results and controller design. The system was controlled by PID, IMC and adaptive discrete STC controllers.

2. SYSTEM DESCRIPTION

The system consists of two propellers driven by DC motors (Fig.1). The movable part has two degrees of freedom. The axes of rotation are perpendicular. The position angles (elevation and azimuth) are influenced by the rotation of propellers. The system can freely rotate around the vertical axis by about 215 degrees and the horizontal axis about 90 degrees. Both angles are measured by sensors and the angular velocities of the rotors by tacho-generators. DC motors are driven by power amplifiers with voltage in the range from 0 to 5 V. The main motor rotation is possible in one direction only, whereas the tail rotor may rotate in both directions. The model is connected with computer by NI USB – 6009 data acquisition device. The PC is equipped with the MATLAB and SIMULINK software along with the tools to perform measurements on the system and to implement controllers in real-time. Detailed description of the system can be found in Havlíček (2010). The paper deals with the control of vertical angle, where the controlled variables was the position angle and the manipulated variable was main motor voltage.

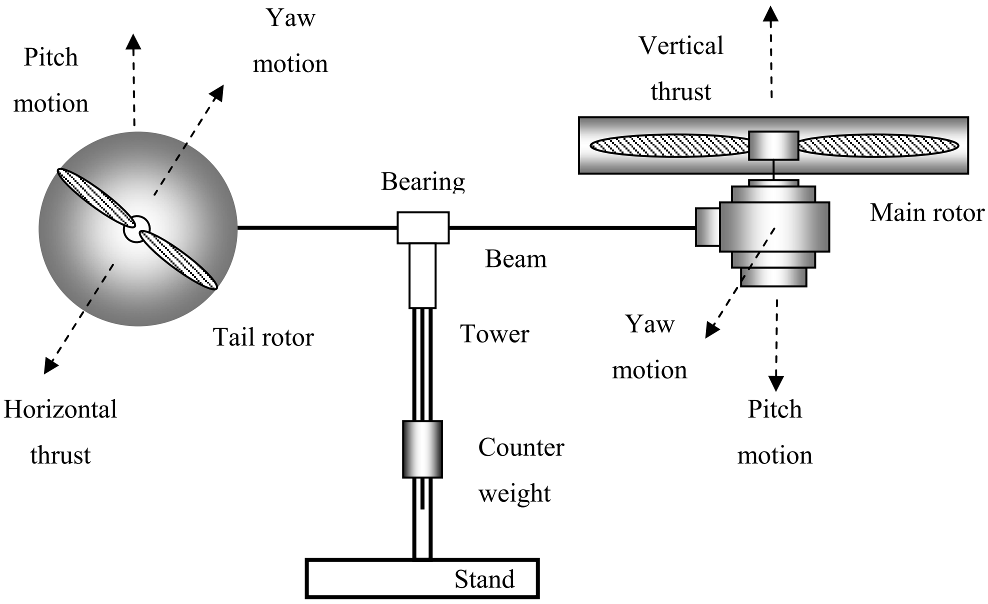


Fig. 1. - Twin rotor system

3. PROPERTIES OF MAIN ROTOR SYSTEM

Input of the main rotor system is the voltage on motor in range from 0 to 5 V and output is vertical angel in range from 0 to 90°. The voltage 2.5 V was taken as a nominal input which corresponds to nominal output angle 29°. Next the values of the input and output variables were taken as deviations from these nominal points. The disturbance variable is presented by the rotation of the tail propeller. Following measurements on the system were performed: Static characteristic, step responses, frequency characteristic and experimental identification.

3.1 Static characteristic

The course of the measured static characteristic is given in Fig. 2. The system is nonlinear and its gain increase with input voltage.

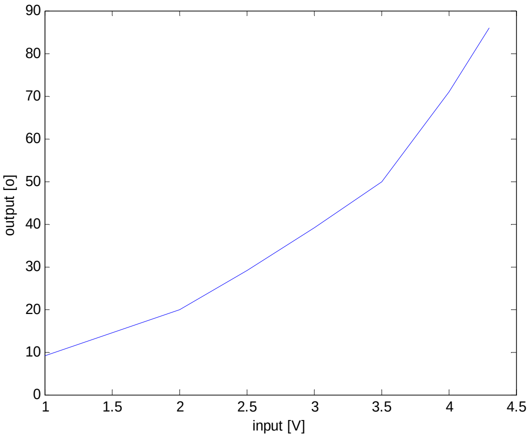


Fig. 2. Static characteristic

3.2 Step responses

The response on step voltage change was measured in three operating points – round about nominal value (step from 2 to 3 V), step for small voltage (from 1 to 2 V) and step for higher voltage (from 3 to 4 V). The all responses are shown in Fig. 3. The step responses change with operating points in accordance with course of the static characteristic.

3.3 Frequency response

The frequency response was measured for the nominal operating point and it is presented in Fig. 4. Maximum gain occurs by the frequency $\omega = 1.8\, s^{-1}$ where phase angle is –π.

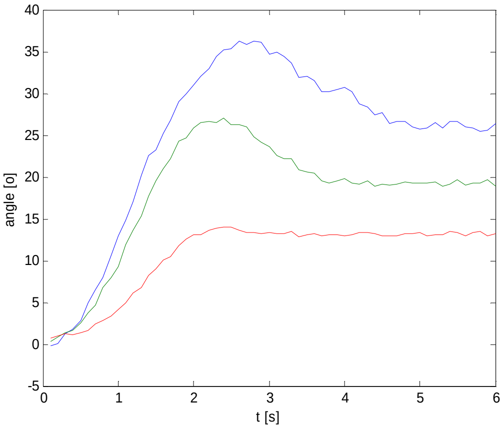


Fig. 3. Comparison of step responses (without time delay 1 s)

(red – step from 1 to 2 V, green – step from 2 to 3 V, blue – step from 3 to 4 V)

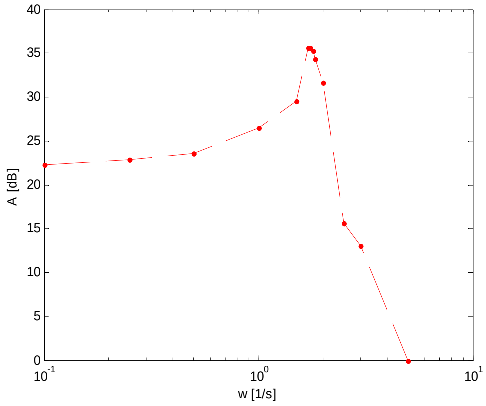


Fig. 4. Amplitude frequency response

3.4 Model from experimental identification

The sampling period was proposed with regard to speed of the data acquisition device during connection with MATLAB – Simulink. The model of STC controller is relatively complicated and the sampling period was hence chosen $T = 1$ s.

Structure of the model was proposed as the system of the second order with time delay 1 s. It is the simplest model which can ensures the overshoot of step response. The difference equation has the form

$$\begin{aligned} &y(k) + a_1 y(k-1) + a_2 y(k-2) = \\ &= z^{-1}\left[b_1 u(k-1) + b_2 u(k-2)\right] \end{aligned} \tag{1}$$

and its parameters were computed by the least-squares method, see *e.g.* Drábek *et al.* (1987). Several step responses close to the nominal values were evaluated and the following parameters were obtained:

$$\begin{aligned} a_1 &= -1.2784 \\ a_2 &= \ \ 0.5460 \\ b_1 &= \ \ 6.3608 \\ b_2 &= -0.6645 \end{aligned} \tag{2}$$

The model (1) with these parameters is the good approximation of dynamic behaviour of the identified system in given area.

4. SYSTEM CONTROL

Several methods were used for controller design. The controllers were realized in MATLAB – Simulink. All controllers were verified for reference tracking and rejection of disturbances (moving of the tail rotor).

4.1 PID controller

The PID controller in continuous form has transfer function

$$G_R(s) = r_0\left[1 + \frac{1}{sT_i} + sT_d\right] \tag{3}$$

and its parameters were designed according to frequency Ziegler-Nichols method - Ziegler *et all.* (1942). The ultimate (critical) values were first measured for the sampling period $T = 0.1$ s: $r_{0k} = 0.0172$ and $T_k = 3.5\,s$. Hence controller parameters are

$$\begin{aligned} r_0 &= 0.6\,r_{0k} = 0.0103 \\ T_i &= 0.5\,T_k = 1.75\,s \\ T_d &= 0.125\,T_k = 0.4375\,s \end{aligned}$$

Response on the step changes of reference and on the influence of tail rotor moving (voltage 4.5 V in time 220 s) is given in Fig. 5.

The ultimate values changed when the sampling period is increased. For example the ultimate gain was r_{0k} = 0.0455 for T = 1 s and the ultimate period was $T_k = 6\,s$. But the PID controller designed from this values was not acceptable, as output variable oscillated. It was caused the nonlinear behaviour of plant. The ultimate values for the other operating point (u = 3.5 V; r_{0k} = 0.0212; T_k =5 s) gave good response (see Fig.6).

4.2 IMC controller

Internal Model Control (IMC) is an effective method of controller design, which requires limited computation - Morari *et al.* (1998), Macháček *et al.* (2004). The block diagram of IMC is shown in Fig. 7, where G is controlled process, G_M is process model and G_RI is controller.

The process model must be factorized on invertible G^- and noninvertible G^+ part:

$$G_M(s) = G^- G^+ \tag{4}$$

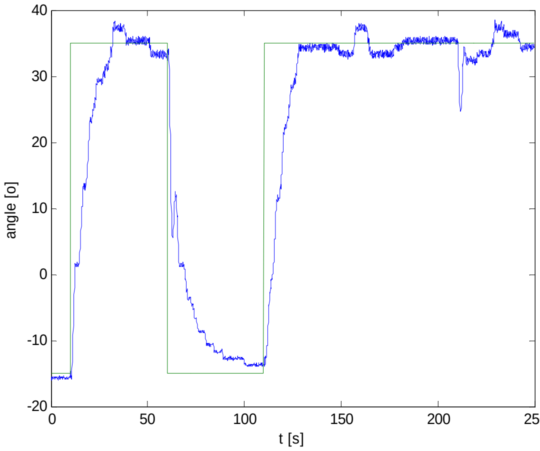


Fig. 5. Control process with PID controller (T = 0.1 s)

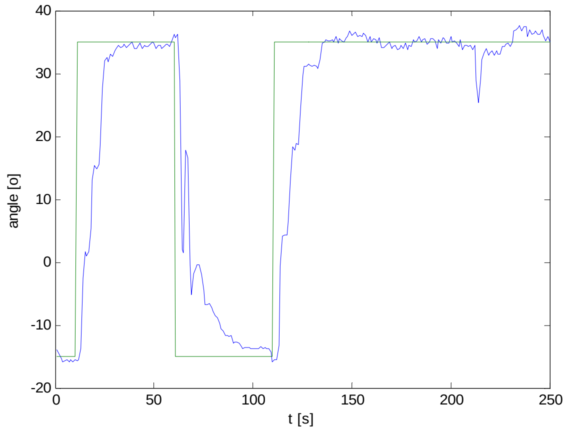


Fig. 6. Control process with PID controller ($T = 1$ s)

The controller includes the invertible part of process model. The noninvertible parts are time delay and the factors with right-half-plane zeros, which stay in closed loop transfer function. A linear filter can be added to make possible the controller realization:

$$G_f(s)=\frac{1}{(\tau_f s+1)^r} \qquad (5)$$

where τ_f is the select parameter for adjustment of the closed-loop dynamics and r is chosen according to model order. The controller is then in the form:

$$G_{RI}(s)=\frac{G_f(s)}{G^-(s)} \qquad (6)$$

The all models with the exception of time delay may be inverted in our case. The filter (5) of second order with $\tau_f = 2\,s$ was chosen and its discrete transfer function for $T = 1$ s was

$$G_f(z)=\frac{0.0902z+0.06446}{z^2-1.213z+0.3679} \qquad (7)$$

The course of experiment was the same as for the PID controller. The measured response is given in Fig. 8.

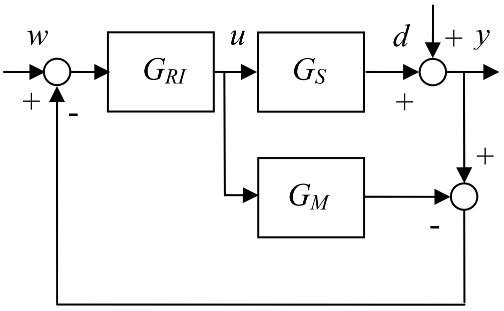


Fig. 7. Block diagram of IMC

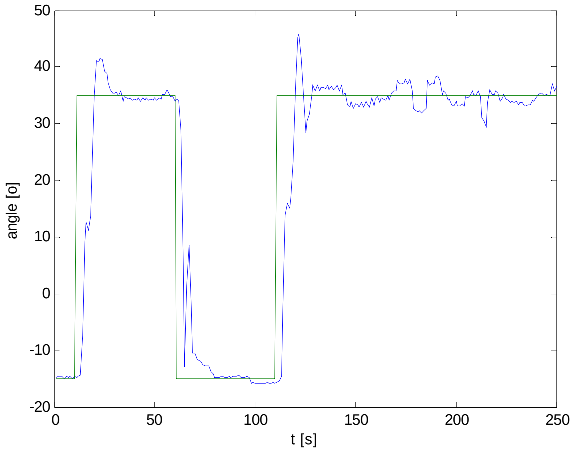


Fig. 8. Control process with IMC controller ($T = 1$ s)

The IMC controller is sensitive on the accuracy of model and control process quality was worse for great reference steps.

4.3 Self-tuning controller

Controllers with fixed parameters are often unsuited to nonlinear processes because their parameters change with operating conditions. One possible alternative for improving the quality of control for such processes is the use of adaptive control system. The approach to adaptive control, called self-tuning controller (STC), is based on the recursive identification of controlled system and subsequently on the design of optimal controller from identified parameters. The controller is digital and works with fixed sampling period *T*. More information about STC can be found *e.g.* in Bobál *et al.* (2005).

The recursive least squares method together with a forgetting strategy is used to estimate the process model parameters, as the part of the general control algorithm. The parameter vector $\boldsymbol{\Theta}$ for ($k+1$)-th time interval is estimated using the following recursive equations:

$$\hat{\boldsymbol{\Theta}}(k+1)=\hat{\boldsymbol{\Theta}}(k)+\boldsymbol{m}(k)[y(k+1)-\boldsymbol{\Phi}^T(k+1)\hat{\boldsymbol{\Theta}}(k)] \qquad (8)$$

$$\boldsymbol{m}(k)=\boldsymbol{C}(k)\boldsymbol{\Phi}(k+1)[\varphi+\boldsymbol{\Phi}^T(k+1)\boldsymbol{C}(k)\boldsymbol{\Phi}(k+1)]^{-1} \qquad (9)$$

$$\boldsymbol{C}(k+1)=\frac{1}{\varphi}[\boldsymbol{C}(k)-\boldsymbol{m}(k)\boldsymbol{\Phi}^T(k+1)\boldsymbol{C}(k)] \qquad (10)$$

where φ is the forgetting factor $\varphi \le 1$.

The data vector $\boldsymbol{\Phi}$ has for the model of second order with time delay and offset form

$$\boldsymbol{\Phi}^T(k+1)=$$
$$=[-y(k) \quad -y(k-1) \quad u(k-d) \quad u(k-d-1) \quad 1\,]$$

and the parameter vector is then

$$\Theta^T(k+1) = [a_1 \quad a_2 \quad b_1 \quad b_2 \quad \text{c}]$$

The algorithm begins with diagonal matrix ***C***, which has the same values on its main diagonal (chosen 100) and an arbitrary initial parameter vector (chosen all 1).

The IMC controller was chosen as the control part of the adaptive algorithm. The on-line identification and IMC controller design are repeated in every sampling time. The model from experimental identification is used for controller design. The transfer functions of model G_M (from Eq. 1), controller G_{RI} (Eq. 6) and filter G_f (Eq. 7) are taken in discrete form with sampling period $T = 1$ s.

The controller was realized directly in Simulink, which can to work (from the version 4) with signals type matrix - Dušek *et al.* (2004). Real time toolbox was then not need to use. The blocks from Simulink Library as *Selector* (selects or reorders specified elements of multidimensional signal) or *Reshape* (changes the dimensions of a vector or matrix signal) were used for modelling identification and control algorithms. Block scheme of the whole adaptive controller is in Fig.9.

Response of the system with STC IMC regulator on the same signals as for the PID and IMC controllers is given in Fig. 10.

5. CONCLUSIONS

The control of the vertical angle was realised with several controllers. The control algorithms were used for tracking of the reference values and rejection of disturbances (moving of the tail rotor). Direct comparison of used methods by some numerical criterion is not suitable. Firstly the controlled system is nonlinear and control process quality depends on operating point. The STC controller is the best method from this objective. In the second place the disturbances are larger then differences among methods.

All controllers gave relatively good control process. The influence of the tail rotor moving on the process control was small. Self tuning controller had better tracking of reference, but worse rejection of disturbances. The results of this work will be used for laboratory education.

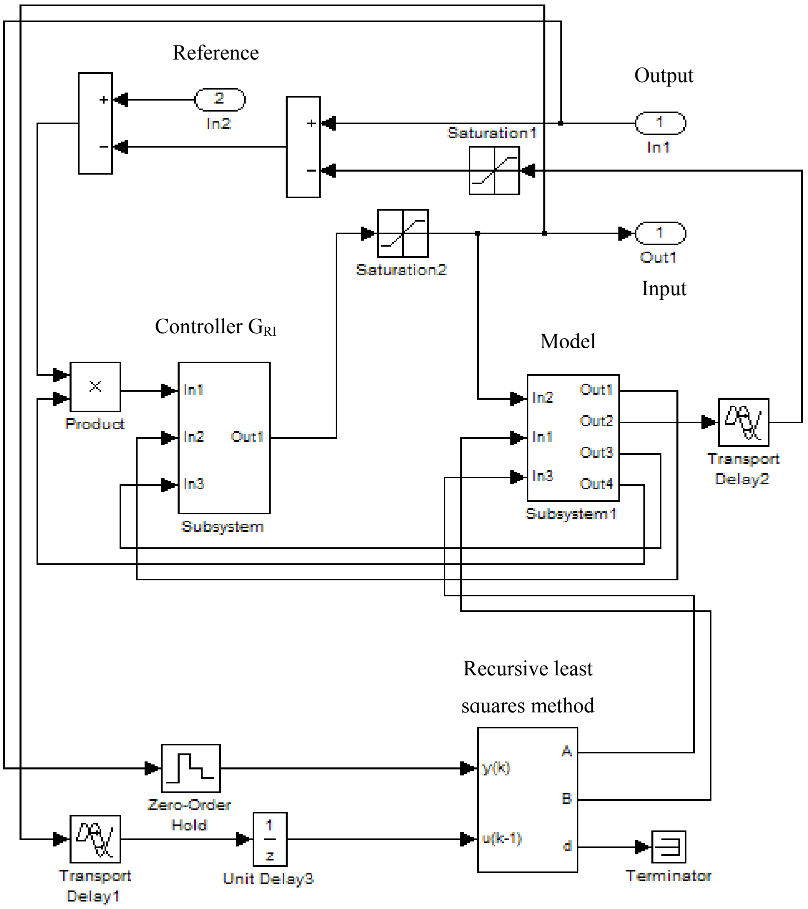


Fig. 9. Block scheme the whole adaptive controller

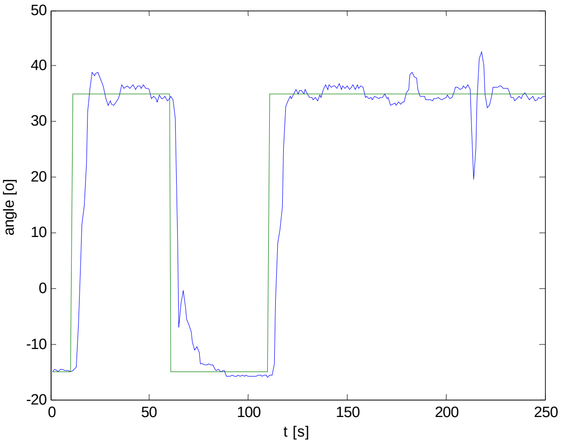


Fig. 10. Control process with STC controller

ACKNOWLEDGMENTS

The work has been supported by the grant MSM 002 162 7505. This support is very gratefully acknowledged.

REFERENCES

V. Bobál, J. Böhm, J. Fessl and J. Macháček. *Digital Self-tuning Controllers*. Springer –Verlag, London, 2005.

O. Drábek, J. Macháček. *Experimentální identifikace*. VŠCHT Pardubice, 1987.

D. Honc, F. Dušek. Maticové operace v SIMULINKu verze 4. In: *10. Konference MATLAB 2002*, ČVUT Praha, s. 152-161, 2002.

L. Havlíček. Modelování a řízení vícerozměrové soustavy. Disertační práce, Univerzita Pardubice, 2010.

J. Macháček, L. Havlíček. Metoda IMC pro nastavování PID regulátorů. *Automatizace*, 47, 123-124, 2004.

M. Morari, E. Zafirou. *Robust Process Control*. Preantice-Hall, Englewood Cliffs, NJ, 1989.

J.G. Ziegler, N.B. Nichols. Optimum Settings for Automatic Controllers, *Trans. ASME*, 64, 759-764, 1942.

Piecewise-Linear Neural Models for Process Control

P. Doležel* I. Taufer J. Mareš*****

** University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control,
Nám. Čs. legií 565, 532 10 Pardubice
(Tel: +420 466 037 504; e-mail:petr.dolezel@upce.cz.)
** University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control,
Nám. Čs. legií 565, 532 10 Pardubice
(e-mail:ivan.taufer@upce.cz.)
*** Institute of Chemical Technology, Department of Computing and Control Engineering,
Technická 5, 166 28 Prague 6
(e-mail: jan.mares@vscht.cz)*

Abstract: There is introduced an algorithm which provides piecewise-linear model of nonlinear plant using artificial neural networks, in this paper. That piecewise-linear model is precise and each linear submodel is valid in some neighbourhood of actual plant state. This model can be used for plant control design. There is presented an example at the end of this paper, where defined nonlinear plant is controlled via Pole Assignment technique using piecewise-linear neural model and control response is compared to data obtained by common PID controller.

1. INTRODUCTION

Artificial Neural Network (ANN) is a popular methodology nowadays with lots of practical and industrial applications. As introduction it is necessary to mention applications as mathematical modelling of bioprocesses in Montague et al. (1994), Teixeira et al. (2005), prediction models and control of boilers, furnaces and turbines in Lichota et al. (2010) or industrial ANN control of calcinations processes and iron ore processes in Dwarapudi, et al. (2007).

Therefore, the aim of the contribution is to explain how to use ANN with piecewise-linear activation functions in hidden layer in process control. To be more specific, there is described technique of controlled plant linearization using ANN nonlinear model. Obtained linearized model is in a shape of linear difference equation.

2. ANN FOR APPROXIMATION

According to Kolmogorov's Superposition Theorem, any real continuous multidimensional function can be evaluated by sum of real continuous one-dimensional functions, see Hecht-Nielsen (1987). If the theorem is applied to ANN, it can be said that any real continuous multidimensional function can be approximated by certain three-layered ANN with arbitrary precision. Topology of that ANN is depictured in Fig. 1. Input layer brings external inputs x_1, x_2, …, x_P into ANN. Hidden layer contains S neurons, which process sums of weighted inputs using continuous, bounded and monotonic activation function. Output layer contains one neuron, which processes sum of weighted outputs from hidden neurons. Its activation function has to be continuous and monotonic.

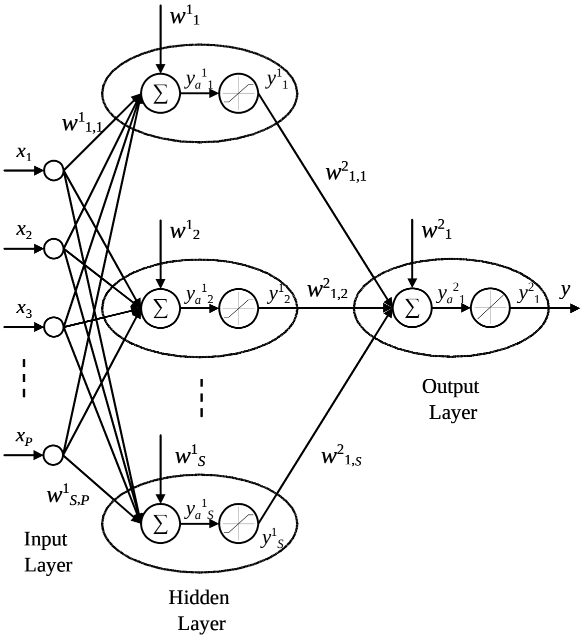


Fig. 1. Three-layered ANN

So ANN in Fig. 1 takes P inputs, those inputs are processed by S neurons in hidden layer and then by one output neuron. Dataflow between input i and hidden neuron j is gained by weight $w^1{}_{j,i}$. Dataflow between hidden neuron k and output neuron is gained by weight $w^2{}_{1,k}$. Output of the network can be expressed by following equations.

$$y_a{}^1{}_j = \sum_{i=1}^{P} w^1{}_{j,i} \cdot x_i + w^1{}_j \qquad (1)$$

$$y^1{}_j = \varphi^1\left(y_a{}^1{}_j\right) \qquad (2)$$

$$y_a{}^2{}_1 = \sum_{i=1}^{S} w^2{}_{1,i} \cdot y^1{}_i + w^2{}_1 \tag{3}$$

$$y = \varphi^2\left(y_a{}^2{}_1\right) \tag{4}$$

In equations above, $\varphi^1(.)$ means activation functions of hidden neurons and $\varphi^2(.)$ means output neuron activation function.

As it is mentioned above, there are some conditions applicable for activation functions. To satisfy those conditions, there is used mostly hyperbolic tangent activation function (eq. 5) for neurons in hidden layer and identical activation function (eq. 6) for output neuron.

$$y^1{}_j = \tanh\left(y_a{}^1{}_j\right) \tag{5}$$

$$y = y_a{}^2{}_1 \tag{6}$$

Mentioned theorem does not define how to set number of hidden neurons or how to tune weights. However, there have been published many papers which are focused especially on gradient training methods (Back-Propagation Gradient Descend Alg.) or derived methods (Levenberg-Marquardt Alg.) – see Haykin (1994).

3. SYSTEM IDENTIFICATION BY ANN

System identification means especially a procedure which leads to dynamic model of the system. ANN has traditionally enjoyed considerable attention in system identification because of its outstanding approximation qualities. There are several ways to use ANN for system identification. One of them assumes that the system to be identified (with input u and output y_S) is determined by the following nonlinear discrete-time difference equation.

$$\begin{aligned} y_S(k) = \psi[&y_S(k-1), \ldots, y_S(k-n), \\ &u(k-1), \ldots, u(k-m)],\ m \le n \end{aligned} \tag{7}$$

In equation above, $\psi(.)$ is nonlinear function, k is discrete time and n is difference equation order.

The aim of the identification is to design ANN which approximates nonlinear function $\psi(.)$. Then, neural model can be expressed by (eq. 8).

$$\begin{aligned} y_M(k) = \hat{\psi}[&y_M(k-1), \ldots, y_M(k-n), \\ &u(k-1), \ldots, u(k-m)],\ m \le n \end{aligned} \tag{8}$$

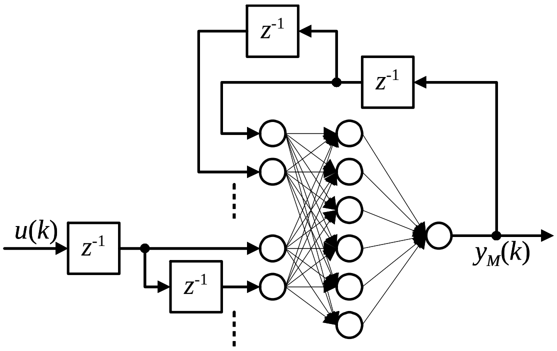


Fig. 2. Neural model

In (eq. 8), $\hat{\psi}$ represents well trained ANN and y_M is its output. Formal scheme of neural model is shown in Fig. 2. It is obvious that ANN in Fig. 2 has to be trained to provide y_M as close to y_S as possible. Existence of such a neural network is guaranteed by Kolmogorov's Superposition Theorem and whole process of neural model design is described in detail in Haykin (1994) or Taufer et al. (2008).

4. PIECEWISE-LINEAR MODEL

As mentioned in section 2, there is recommended to use hyperbolic tangent activation function for neurons in hidden layer and identical activation function for output neuron in ANN used in neural model. However, if linear saturated activation function (eq. 9) is used instead, ANN features stay similar because of resembling courses of both activation functions (see Fig. 3).

$$y^1{}_j = \begin{cases} 1 & \text{for} \quad y_a{}^1{}_j > 1 \\ y_a{}^1{}_j & \text{for} \quad -1 \le y_a{}^1{}_j \le 1 \\ -1 & \text{for} \quad y_a{}^1{}_j < -1 \end{cases} \tag{9}$$

The output of linear saturated activation function is either constant or equal to input so neural model which uses ANN with linear saturated activation functions in hidden neurons acts as piecewise-linear model. One linear submodel turns to another when any hidden neuron becomes saturated or becomes not saturated.

Let us presume an existence of some dynamic neural model which uses ANN with linear saturated activation functions in hidden neurons and identic activation function in output neuron – see Fig. 4. Let us also presume $m = n = 2$ for making process easier. ANN output can be computed using eqs. (1), (2), (3), (4). However, another way for ANN output computing is useful. Let us define saturation vector $\mathbf{z}$ of S elements. This vector indicates saturation states of hidden neurons – see (eq. 10).

$$z_i = \begin{cases} 1 & \text{for} \quad y^1{}_i > 1 \\ 0 & \text{for} \quad -1 \le y^1{}_i \le 1 \\ -1 & \text{for} \quad y^1{}_i < -1 \end{cases} \tag{10}$$

Then, ANN output can be expressed by (eq. 11).

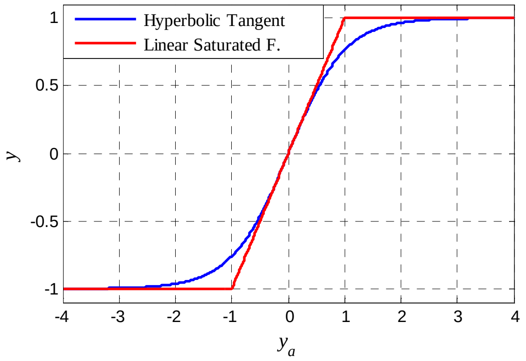


Fig. 3. Activation functions comparison

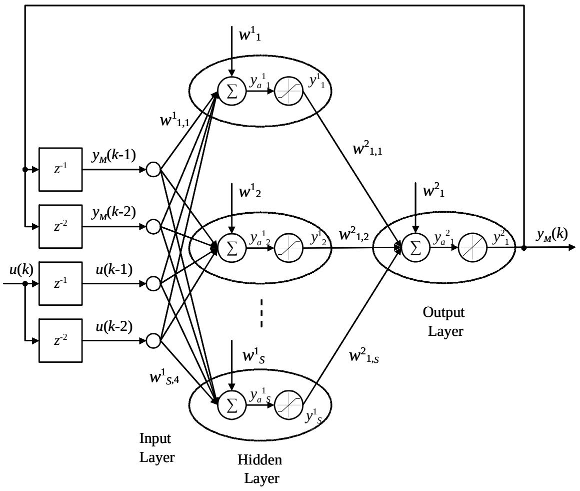


Fig. 4. Piecewise-linear neural model

$$y_M(k) = -a_1 \cdot y_M(k-1) - a_2 \cdot y_M(k-2) + \\ + b_1 \cdot u(k-1) + b_2 \cdot u(k-2) + c \tag{11}$$

where:

$$a_1 = -\sum_{i=1}^{S} w^2{}_{1,i} \cdot \left(1-|z_i|\right) \cdot w^1{}_{i,1}$$

$$a_2 = -\sum_{i=1}^{S} w^2{}_{1,i} \cdot \left(1-|z_i|\right) \cdot w^1{}_{i,2}$$

$$b_1 = \sum_{i=1}^{S} w^2{}_{1,i} \cdot \left(1-|z_i|\right) \cdot w^1{}_{i,3}$$

$$b_2 = \sum_{i=1}^{S} w^2{}_{1,i} \cdot \left(1-|z_i|\right) \cdot w^1{}_{i,4}$$

$$c = w^2{}_1 + \sum_{i=1}^{S} \left(w^2{}_{1,i} \cdot z_i + \left(1-|z_i|\right) \cdot w^2{}_{1,i} \cdot w^1{}_i\right)$$

Thus, difference equation (11) defines ANN output and it is linear in some neighbourhood of actual state (in that neighbourhood, where saturation vector **z** stays constant). Difference equation (11) can be clearly extended into any order.

In other words, if it is designed neural model of any nonlinear system in form of Fig. 4, then it is simple to determine parameters of linear difference equation which approximates system behaviour in some neighbourhood of actual state. This difference equation can be used then to the actual control action setting due to any of classical or modern control techniques.

If chosen control technique requires model in form of difference equation with no constant term (c = 0), (eq. 11) can be transformed in following way. Let us define

$$\tilde{u}(k) = u(k) - u_0 \tag{12}$$

where u_0 is constant. Then, (eq. 11) turns into

$$y_M(k) = -a_1 \cdot y_M(k-1) - a_2 \cdot y_M(k-2) + \\ + b_1 \cdot \tilde{u}(k-1) + b_2 \cdot \tilde{u}(k-2) + c + \\ + (b_1 + b_2) \cdot u_0 \tag{13}$$

Equation (13) becomes constant term free, if (eq. 14) will be satisfied.

$$u_0 = -\frac{c}{b_1 + b_2} \tag{14}$$

It is obvious that mentioned procedure can be extended into any order of difference equation.

Whole algorithm of piecewise-linear neural model usage in process control is summarized in following terms.

1. Create neural model of controlled plant in form of Fig. 4.
2. Set k = 0.
3. Measure plant output $y_S(k)$.
4. Determine parameters a_i, b_i and c of difference equation (11).
5. Transform (eq. 11) into (eq. 13).
6. Determine $\tilde{u}(k)$ according to some chosen control technique using linear plant model in form (eq. 13).
7. Transform $\tilde{u}(k)$ into $u(k)$ using (eq. 12) and perform control action.
8. $k = k + 1$, go to 3.

5. EXAMPLE

Demonstrative nonlinear controlled system is defined by difference equation (15).

$$y_S(k) = \\ \frac{1.5 \cdot y_S(k-1) - 0.8 \cdot y_S(k-2) + 0.1 \cdot u(k-1) + 0.05 \cdot u(k-2)}{1 - 0.1 \cdot y_S(k-1) + 0.2 \cdot \left[y_S(k-1)\right]^2} + \\ + 0.6 \cdot \sqrt{u(k-1)} \tag{15}$$

There are defined the boundaries of input $u(k)$ to interval <0;3>. Static characteristic of the system is figured below (Fig. 5).

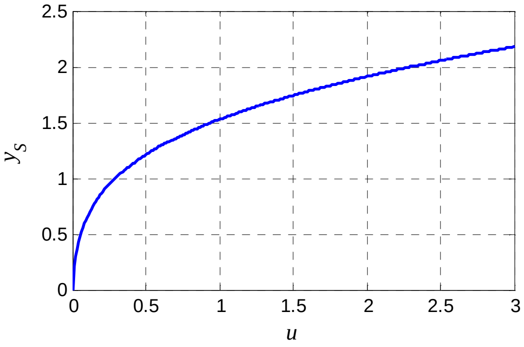


Fig. 5. Static characteristic of the system

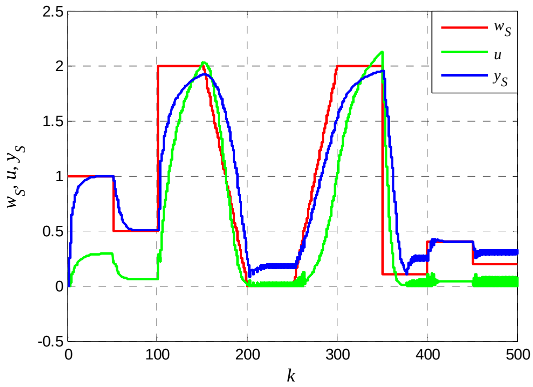


Fig. 6. Control response with PID controller

Firstly, system is controlled with PID controller tuned by trial and error – more sophisticated tuning methods fail to bring better performances because of significant nonlinearity of the plant. Control response (Fig. 6) shows serious lack of quality. For lower values of controlled variable $y_S(k)$, control performance oscillates unacceptably, while for higher values of $y_S(k)$, control performance is too damped.

Then, piecewise-linear neural model is used for control. Neural model is designed according to information described in section 4. Detailed description of the process is not referred here, because it is standard well-known procedure. Certain control technique, which can use system model in form of (eq. 13), has to be determined. In this demonstration, Pole Assignment control technique (PA) of Algebraic Control Theory is used.

In simple words, this control technique determines controller parameters so that whole closed control loop behaves as some defined standard. In one its version, PA uses control loop shown in Fig. 7. Controlled system should be described by polynomials $A(z^{-1})$, $B(z^{-1})$, where polynomial parameters are equal to difference equation parameters used for linear model of the controlled system. Both feedforward and feedback part of controller are defined by polynomials $P(z^{-1})$, $Q(z^{-1})$, $R(z^{-1})$, which can be determined by solving of several diophantine equations. Standard for control loop behaviour has to be chosen. Whole procedure of PA is described in detail in book edited by K. J. Hunt (1993).

Standard for this demonstration is defined as discrete first order system with unit gain and denominator (1 -0.6065z^{-1}).

Control performance is shown in Fig. 8. Compared to Fig. 6, there comes clear improvement.

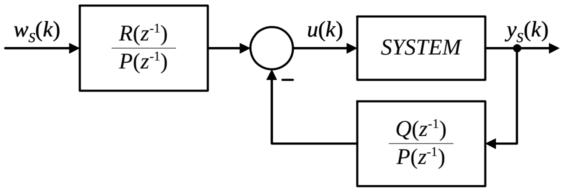


Fig. 7. Pole Assignment Control Technique

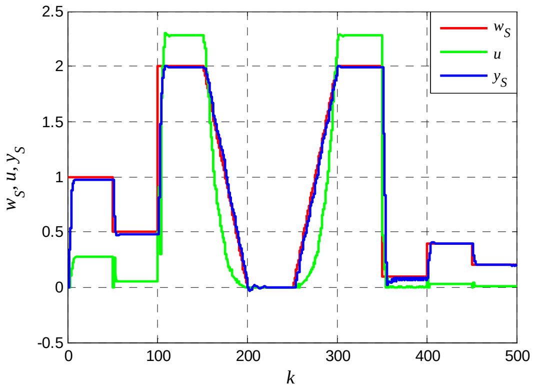


Fig. 8. Control Response with PA Controller and Piecewise-Linear Neural Model

6. CONCLUSIONS

The paper is focused on usage of neural network with linear saturated activation functions in process control. Neural model with such a neural network within is suitable for controller design using any of huge set of classical or modern control techniques. As example, there is presented control of nonlinear discrete plant using Pole Assignment technique. Comparison to control performance provided by PID controller proves great improvement.

ACKNOWLEDGMENTS

The work has been supported by the funds No. MSM 6046137306 and No. MSM 0021627505 of Ministry of Education of the Czech Republic, No. MEB 0810003 of Ministry of Education, Science, Research and Sport of the Slovak Republic and of Ministry of Education of the Czech Republic and No. SGFEI06/2011. This support is very gratefully acknowledged.

REFERENCES

Dwarapudi, S., Gupta, P. K. and Rao, S. M. (2007). Prediction of iron ore pellet strength using artificial neural network model, *ISIJ International,* Vol. 47, No 1. pp. 67-72, ISSN 0915-1559.

Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*. Prentice Hall. New Jersey. ISBN 0023527617

Hecht-Nielsen, R. (1987). Kolmogorov's mapping neural network existence theorem. In: *Proc 1987 IEEE International Conference on Neural Networks.* Vol. 3, pp. 11-13. IEEE Press.

Hunt, K. J., Ed. (1993). *Polynomial methods in optimal control and filtering*. Peter Peregrinus Ltd. Stevenage. ISBN 0-86341-295-5.

Lichota, J. and Grabovski, M. (2010). Application of artificial neural network to boiler and turbine control, *Rynek Energii*, Vol. 16, No 1. ISSN 1425-5960.

Montague, G. and Morris, J. (1994). Neural network contributions in biotechnology, *Trends in biotechnology*, Vol. 12, No 8. pp. 312-324, ISSN 0167-7799.

Taufer, I., Drábek, O., Seidl, P. (2008). Umělé neuronové sítě – základy teorie a aplikace (10), *CHEMagazín*, vol. XVII, issue 1, pp. 35-37. ISSN 1210-7409.

Teixeira, A., Alves, C. and Alves, P. M. (2005). Hybrid metabolic flux analysis/artificial neural network modelling of bioprocesses, In: *Proceedings of the 5th International Conference on Hybrid Intelligent Systems*, IEEE Computer Society, Los Alamitos. ISBN 0-7695-2457-5.

PI/PID Controller Design for FOPDT Plants Based on the Modulus Optimum Criterion

J. Cvejn*

** University of Pardubice, Faculty of Electrical Engineering and Informatics, Studentská 95, 532 10 Pardubice, Czech Republic (e-mail: jan.cvejn@upce.cz)*

Abstract: We present the PI/PID controller settings for the first order systems with dead time, based on the modulus optimum criterion. The settings provide fast closed-loop response to changes of the reference input. Unlike most other tuning methods, the parameters are obtained without approximation of the delay term, so they remain valid for long dead time. Besides the performance indices, quality of the settings is also evaluated by the stability margin. Although optimal values of the parameters are valid for the reference tracking problem, a compensation of the disturbance lag that preserves the stability margin is proposed for the disturbance rejection problem.

1. INTRODUCTION

Many industrial processes are modelled by the stable first-order plus dead time (FOPDT) transfer function:

$$S(s)=\frac{K}{Ts+1}e^{-\tau s} \tag{1}$$

where K is the system gain, $T>0$ is the time constant and τ is the dead time parameter. The model (1) allows simple experimental identification from the step response, which can be in most cases easily measured. Simple methods based on coincidence in one or more points and more complex methods suitable for noisy data are described in (Åström and Hägglund 1995) and (Kiong et al. 1999).

For tuning PID controllers based on the model (1) many approaches exist, see e.g. (Åström and Hägglund 1995) for description of the most important methods. A comprehensive survey of known formulas is available in (O´Dwyer 2003). Early methods were derived from empirical requirements on the step response, such as one-quarter decay ratio (Cohen and Coon 1953), (Ziegler and Nichols 1942), step-response overshoot (Chien at al. 1952) or from integral criterions in time domain with approximation of the dead-time dynamics (Lopez et al. 1967), (Wang et al. 1995). These methods, however, usually work well only for a rather limited range of the ratio τ/T.

Among methods for the model of type (1) without approximation of the delay term the design with given gain and phase margins (Ho et al. 1995) and LQR design (Kiong et al. 1999) should be mentioned. Alternative ways for systems with long time delay include internal model control (Rivera et al. 1986), Smith predictor and λ-tuning (Åström and Hägglund 1995). These approaches, however, require implementation of delay in the control system.

The method for setting up PI controller parameters based on cancellation of the factor $(Ts+1)$ was proposed in (Haalman 1965). In this method the dead-time dynamics is manipulated without approximation. Good reference tracking performance is achieved, but on the other hand, poor results may be observed for rejection of load disturbances (Åström and Hägglund 1995). In a similar way it is possible to compensate dynamics of the second order plus dead time system by a PID controller.

In this paper we utilize Haalman's idea of pole compensation for designing optimal PI and PID controller parameters for the model (1). The pole compensation fixes the value of one parameter of the controller. We adjust the remaining controller parameters to meet analytic design criteria. We show that in this case especially the modulus optimum criterion leads to a simple choice of the parameters and to a control loop with very good practical properties. In this case, derivative term of the controller increases both the performance and the stability margin. The results presented here appeared in full context in the journal paper (Cvejn 2009), where also the settings based on the minimum ISE criterion were analyzed.

The modulus optimum criterion introduced in (Oldenbourg and Sartorius 1956) requires that the amplitude of the closed-loop frequency response is close to one for low frequencies. If the closed-loop frequency response is decreasing, this condition is analogous to the requirement that the frequencies in the reference input are passed in the broadest possible range. Such a behaviour is desirable for the reference tracking cases, because then the closed-loop system is able to respond quickly to changes of the reference input.

Let us write the closed-loop frequency response in the form

$$T(\omega)=\frac{L(i\omega)}{1+L(i\omega)}=\frac{1}{1+1/L(i\omega)} \tag{2}$$

where $L(s)$ is the open-loop transfer function in Laplace transform. If $L(s)$ contains a pole in the origin, which is necessary to achieve asymptotically zero regulation error, for

$\omega \to 0$ holds $L(i\omega) \to \infty$ and thus $\lim_{\omega \to 0} |T(\omega)|^2 = 1$. Therefore, it is possible to write $|T(\omega)|^2$ as

$$|T(\omega)|^2 = 1 + H(\omega) \tag{3}$$

where $H(\omega) \in C_\infty$. Maximal flatness of the closed-loop frequency-response modulus is then equivalent to the requirement that

$$n_0\left(H(\omega)\right) \to \max \tag{4}$$

where $n_0\left(H(\omega)\right)$ denotes the index of the first nonzero coefficient in the Taylor expansion of $H(\omega)$.

Besides performance objectives, the design has to respect stability requirements. As the stability margin we consider the distance of the open-loop Nyquist plot from the critical point $[-1,\,0]$, i.e. the value

$$\gamma = \inf_{\omega \in (0,\infty)} \left\{1 + L(i\omega)\right\}, \quad \gamma \in [0,\,1]. \tag{5}$$

The reciprocal value of γ is known as the sensitivity function. In general case it is recommended that the sensitivity is in the range from 1.3 to 2 (Åström and Hägglund 1995).

2. THE CONTROLLER DESIGN

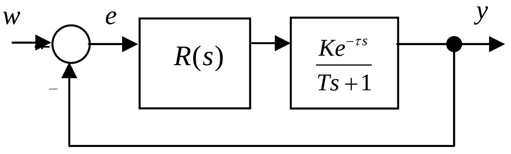


Fig. 1. Control scheme for reference tracking

At first, consider the PI controller case. Consider the reference tracking control problem in Fig. 1. If we compensate the factor $(Ts+1)$ by the PI controller

$$R(s) = K_C\left(1 + \frac{1}{T_I s}\right) \tag{6}$$

where

$$K_C = \frac{\kappa T}{K\tau}, \quad T_I = T \tag{7}$$

the open-loop transfer function is

$$L(s) = \frac{\kappa}{\tau s} e^{-\tau s} \tag{8}$$

where κ is a tuning parameter. The corresponding frequency response can be written as

$$L(\xi) = \frac{\kappa}{i\xi} e^{-i\xi} = \frac{\kappa}{\xi} e^{-i(\xi + \pi/2)} = -\kappa\left(\frac{\sin\xi}{\xi} + i\frac{\cos\xi}{\xi}\right) \tag{9}$$

where $\xi = \tau\omega$ is normalized frequency.

The corresponding Nyquist plot is dependent only on a single parameter κ, which can be adjusted so that sufficient stability margin is guaranteed and performance objectives are fulfilled.

In the case of serial PID controller we can put analogously

$$\begin{aligned} R(s) &= K_C\left(1 + \frac{1}{T_I s}\right)\left(1 + T_D s\right) = \\ &= \kappa\frac{T}{\tau K}\left(1 + \frac{1}{Ts}\right)\left(1 + T_D s\right) \end{aligned} \tag{10}$$

and we easily obtain the corresponding open-loop transfer function

$$L(s) = \kappa\left(\frac{T_D}{\tau} + \frac{1}{\tau s}\right)e^{-\tau s} \tag{11}$$

and the frequency response in the form

$$L(\xi) = \kappa\left(\delta - i\frac{1}{\xi}\right)e^{-i\xi} \tag{12}$$

where $\xi = \tau\omega$ is normalized frequency and $\delta = T_D / \tau$.

Proposition 1. The modulus-optimum settings for the PID controller (10) are:

$$\delta^* = 1/3 \text{ and } \kappa^* = 3/4. \tag{13}$$

and for the PI controller (6):

$$\kappa^* = 1/2. \tag{14}$$

Proof: The closed-loop frequency response (2) square modulus is

$$|T(\xi)|^2 = \frac{1}{1 + \frac{1}{|L|^2}\left(1 + 2\operatorname{Re} L\right)} = \frac{1}{1 + Q(\xi)}. \tag{15}$$

It can be easily verified (Cvejn 2009) that if

$$1 + H(\xi) = \frac{1}{1 + Q(\xi)} \tag{16}$$

where $Q(\xi) \in C_\infty$, it holds

$$n_0\left(H(\xi)\right) = n_0\left(Q(\xi)\right). \tag{17}$$

Since

$$|L(\xi)|^2 = \kappa^2\left(\delta^2 + \frac{1}{\xi^2}\right) \tag{18}$$

it is

$$n_0\left(Q(\xi)\right) = n_0\left(1 + 2\operatorname{Re} L(\xi)\right) + 2 = n_0\left(G(\xi)\right) + 1 \tag{19}$$

where

$$G(\xi) = \xi\left(1 + 2\operatorname{Re} L(\xi)\right). \tag{20}$$

To achieve maximal $n_0\left(Q(\xi)\right)$ and thus maximal $n_0\left(H(\xi)\right)$ we require that the derivatives of $G(\xi)$ are zero for $\xi \to 0$ up to maximal order. After substitution, it is easily found that

$$\begin{aligned}&\lim_{\xi\to 0}\frac{dG}{d\xi} = 1+2\kappa\left(\delta-1\right)\\ &\lim_{\xi\to 0}\frac{d^3G}{d\xi^3} = 2\kappa\left(1-3\delta\right)\\ &\lim_{\xi\to 0}\frac{d^5G}{d\xi^5} = 2\kappa\left(5\delta-1\right)\end{aligned} \qquad (21)$$

$$\text{and} \quad \lim_{\xi\to 0}\frac{d^{(k)}G}{d\xi^{(k)}} = 0,\ k = 0,2,4,... \qquad (22)$$

Putting (21) equal to zero yields $\delta^* = 1/3$ and $\kappa^* = 3/4$. For PI controller, where $\delta = 0$, it follows that the optimal setting is $\kappa^* = 1/2$. □

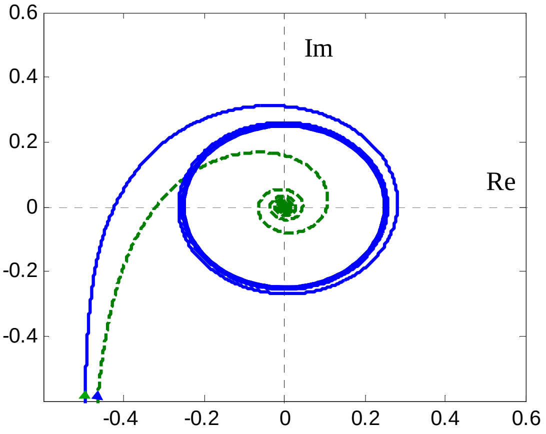


Fig. 2. Open-loop Nyquist plots of proposed settings (solid line – PID controller, dashed line – PI controller)

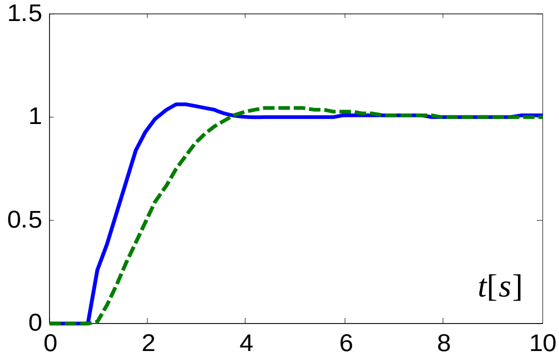


Fig. 3. Step responses (solid line – PID controller, dashed line – PI controller)

Figures 2 and 3 show the Nyquist plots and corresponding step responses for the proposed settings (the ideal open-loop transfer function (11) with $\tau = 1s$ is considered). Obtained settings obviously have very good quality for most practical purposes – the time response is fast and nearly not oscillating. The overshoot is of about 6 % in the case of PID controller. Figure 4 shows the corresponding dependence of $|T(\xi)|$ on ξ.

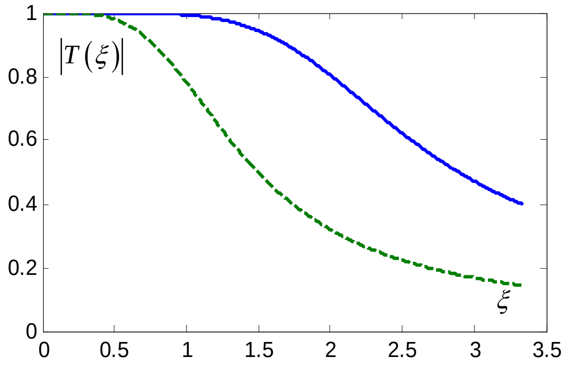


Fig. 4. Dependence $\zeta \to |T(\xi)|$ (solid line – PID controller, dashed line – PI controller)

The resulting parameters of PI, serial PID and the parallel PID controller

$$R(s) = K_C\left(1+\frac{1}{T_I s}+T_D s\right) \qquad (23)$$

are summarized in Tab. 1.

Controller	K_C	T_I	T_D
PI	$\frac{1}{2}\frac{T}{K\tau}$	T	-
PID (serial)	$\frac{3}{4}\frac{T}{K\tau}$	T	$\frac{1}{3}\tau$
PID (parallel)	$\frac{1}{4K}\left(1+\frac{3T}{\tau}\right)$	$T+\frac{\tau}{3}$	$\frac{\tau}{3+\tau/T}$

Tab. 1. PI/PID controller settings for reference tracking

For the ultimate normalized frequency, where $\arg L(\xi_c) = -\pi$, we easily obtain the equation

$$\xi_c = \pi - \operatorname{arctg}\left(\frac{1}{\delta\xi_c}\right) \qquad (24)$$

which can be solved iteratively. Denote ψ the angle between negative real axis and the Nyquist plot of $L(\xi)$ at the ultimate frequency $\xi = \xi_c$. Geometrical shape of the curve (Fig. 2) enables to construct a lower bound of the stability margin γ using the angle ψ:

$$\gamma \ge \underline{\gamma} = \left(1-\frac{1}{\alpha}\right)\sin\psi \qquad (25)$$

where α is the amplitude margin. $\underline{\gamma} \approx 0.54$ was obtained for PID controller and $\underline{\gamma} = 0.57$ for PI controller.

3. THE DISTURBANCE REJECTION PROBLEM

In most practical cases the reference input is held constant, but the system is excited by external disturbances. We consider that the disturbance influences output through the FOPDT transfer function (see Fig. 5).

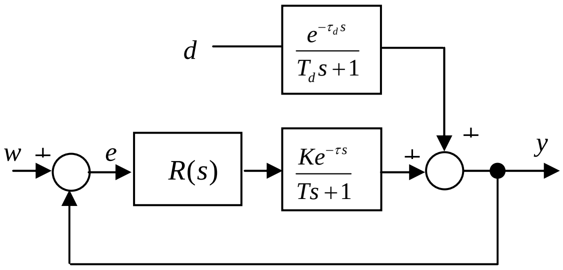


Fig. 5. Control scheme for disturbance rejection

It is well known that good tracking performance does not imply efficient disturbance rejection (Åström and Hägglund 1995). The closed-loop transfer function between *d* and *y* is

$$S_d(s) = \frac{e^{-\tau_d s}}{T_d s+1} \frac{1}{1+L(s)} \tag{26}$$

and thus the factor $1/(T_d s+1)$ will be present in the response regardless of the controller settings, unless it is compensated by a closed-loop zero.

Since the rise time in the optimal configuration including dead time is not shorter than about 2τ in all the configurations, if $T_d \le 2\tau$, total dynamics is not affected much adversely by the term $1/(T_d s+1)$ in the input. On the other hand, if $T_d \gg 2\tau$, the factor $1/(T_d s+1)$ can slow down the response significantly.

The term on the right in (26) corresponds to the transfer function of the regulation error at the reference tracking problem and thus the optimal disturbance rejection problem is analogous to the problem of optimal tracking reference signal with L-transform

$$W(s) = \frac{e^{-\tau_d s}}{T_d s+1}\frac{1}{s}. \tag{27}$$

Therefore, one way how to achieve good performance is to sufficiently decrease T_d, while keeping the other parts of the closed-loop transfer function unchanged. If *d* is not measured, both these objectives probably cannot be fulfilled. Below a compensation that reduces T_d to T_d' and simultaneously approximately preserves the stability margin is proposed. It is possible to assume that in this case the performance will not be much degraded.

If we choose the controller in the form

$$R_d(s) = \frac{T_d}{T_d'} \frac{T_d' s+1}{T_d s+1} R(s) \tag{28}$$

where $R(s)$ is the controller tuned for the reference tracking and $T_d' < T_d$, the closed-loop transfer function is

$$\begin{aligned} S_d(s) &= \frac{e^{-\tau_d s}}{T_d s+1} \frac{1}{1+\frac{T_d}{T_d'}\frac{T_d' s+1}{T_d s+1}L(s)} = \\ &= \frac{T_d'}{T_d}\frac{e^{-\tau_d s}}{T_d' s+1} \frac{1}{\frac{T_d'}{T_d}\frac{T_d s+1}{T_d' s+1}+L(s)}. \end{aligned} \tag{29}$$

However, such a reduction of T_d at the same time decreases the stability margin. Denote $L_d(\xi)$ the open-loop frequency response if the controller (28) is used. If we assume that $\tau/(T_d'\xi) \ll 1$ and $T_d \ge T_d'$, holds

$$L_d(\xi) = \frac{T_d}{T_d'} \frac{iT_d'\xi/\tau+1}{iT_d\xi/\tau+1} L(\xi) \approx \left(1 - i\frac{1}{\xi r_d}\right) L(\xi) \tag{30}$$

where

$$r_d = \frac{1}{\tau}\left(\frac{1}{T_d'} - \frac{1}{T_d}\right)^{-1}. \tag{32}$$

We determine the parameter r_d from the condition

$$\frac{\underline{\gamma} - \underline{\gamma}_d}{\underline{\gamma}} = h \tag{33}$$

where $\underline{\gamma}$, $\underline{\gamma}_d$ are the lower estimates of the stability margin given by formula (25) for the original controller and the modified controller (28), respectively, and *h* is a sufficiently small chosen constant. Equation (33) is solved iteratively, see (Cvejn 2009) for complete explanation. T_d' is then obtained from

$$T_d' = \left(\frac{1}{r_d \tau} + \frac{1}{T_d}\right)^{-1}. \tag{34}$$

The value h should be chosen so that the stability margin be approximately preserved, but since small h leads to large T_d', a compromise has to be looked for. A good choice seems to lie in the range $h \in [0.05, 0.08]$. The results corresponding to $h = 1/16$ are $r_d = 5.92$ for PI controller and $r_d = 3.91$ for PID controller. Note that for $T_d/\tau \to 0$ we obtain $R_d(s) \to R(s)$ and the controller is the same as for the reference tracking problem.

Figure 6 shows the open-loop Nyquist plots after the compensation for PID controller, for $T_d \to \infty$, $T_d = 5\tau$, $T_d = \tau$ and $T_d = 0$ (i.e. without compensation).

A disadvantage of the proposed compensation is that additional term $(T_d' s+1)/(T_d s+1)$ is needed. However, if $T_d = T$ (this is also the case when the disturbance influences the system input), the factor $1/(Ts+1)$ of the system need not be directly compensated. Instead, the PID controller takes the form

$$R_d(s) = \frac{T}{T_d'}\frac{T_d' s+1}{Ts+1}\frac{\kappa T}{\tau K}\left(1+\frac{1}{Ts}\right)(T_D s+1) = \\ = \frac{\kappa T}{\tau K}\left(1+\frac{1}{T_d' s}\right)(T_D s+1) \qquad (35)$$

which is similar to (10). Note that the integral term of the controller, which is needed to achieve zero regulation error, here plays the additional role of compensating the disturbance lag.

The corresponding settings of PI and PID controllers for input disturbance rejection, $h=1/16$, are summarized in Tab. 2.

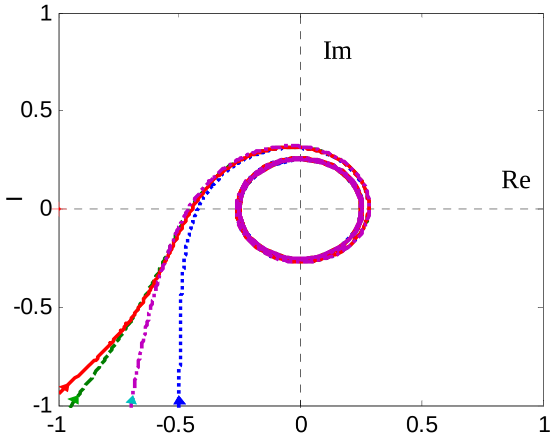


Fig. 6. Open-loop Nyquist plots after compensation, PID controller for $T_d \to \infty$ (solid line), $T_d = 5\tau$ (dashed), $T_d = \tau$ (dash-dotted) and $T_d = 0$ (dotted)

Ctrl.	K_C	T_I	T_D
PI	$\frac{1}{2}\frac{T}{K\tau}$	$\left(\frac{1}{5.9\tau}+\frac{1}{T}\right)^{-1}$	-
PID serial	$\frac{3}{4}\frac{T}{K\tau}$	$\left(\frac{1}{3.9\tau}+\frac{1}{T}\right)^{-1}$	$\frac{1}{3}\tau$
PID parallel	$\frac{1}{4K}\left(1+3.26\frac{T}{\tau}\right)$	$\left(\frac{1}{3.9\tau}+\frac{1}{T}\right)^{-1}+\frac{1}{3}\tau$	$\left(\frac{3.26}{\tau}+\frac{1}{T}\right)^{-1}$

Tab. 2. PI/PID controller settings for input disturbance rejection

4. EXPERIMENTAL COMPARISONS

At first, we consider the reference tracking problem, where the reference signal is not known in advance. Although many of PID tuning formulas for the model (1) are available, most of them are applicable only for a limited range of the ratio $\theta = \tau/T$. Usually, it is required that $\theta \ge 0.1$ and $\theta \le 1$ or $\theta \le 2$. The minimum ISE, IAE and ITAE tuning rules in (Wang et al. 1995), where the recommended range of θ is $\theta \in [0.05,\ 6]$, are among the exceptions. Note that the modulus-optimum (MO) settings we propose admit any positive value of θ.

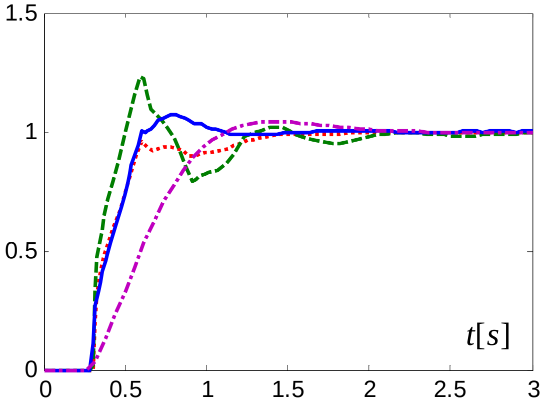


Fig. 7. $F_1(s)$ reference tracking, step response. Settings: MO-PID (solid line), MO-PI (dash-dotted), Chien (dashed), Wang (dotted)

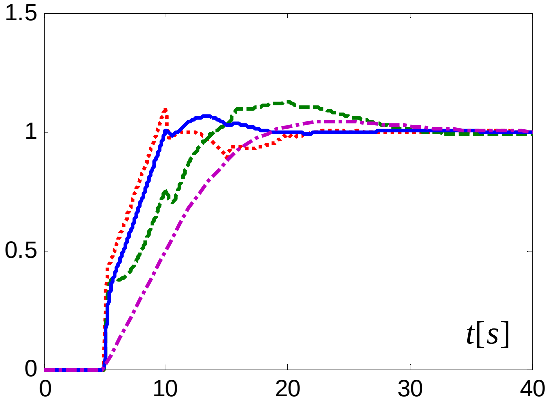


Fig. 8. $F_2(s)$ reference tracking, step response. Settings: MO-PID (solid line), MO-PI (dash-dotted), Chien (dashed), Wang (dotted)

Figures 7 and 8 show the reference signal step responses for the plants

$$F_1(s) = \frac{1}{s+1}e^{-0.3s}, \quad F_2(s) = \frac{1}{s+1}e^{-5s} \qquad (36)$$

and parallel (ideal) PID controller tuned by using Wang IAE formulas (Wang et al. 1995), well known formulas by Chien et al. (Chien et al. 1952) for 20% step-response overshoot and

MO settings of PI and PID controller. Chien settings for large θ result in too slow response, which could be expected, since recommended range of θ is $\theta \in (0.11, 1)$. We also tested well known Ziegler-Nichols formula (Ziegler and Nichols 1942), which for $F_1(s)$ give a rather oscillating response with about 75% overshoot and for $F_2(s)$ slow and overdamped response.

For the disturbance rejection problem we consider that the disturbance influences the system input, i.e. $T_d = T$. Figures 9 an 10 show the load disturbance step responses for the plants $F_1(s)$ and $F_2(s)$ and parallel PID controller tuned according to the minimum IAE formulas in (Lopez et al. 1967), disturbance rejection formulas with 20% overshoot in (Chien et al. 1952) and MO-tuned PI and PID controllers with the input disturbance compensation. Obviously, Lopez and Chien formulas, recommended for $\theta \in [0.1, 1]$, are not suitable for large θ. Ziegler-Nichols settings give responses very similar to Chien settings, in both the cases. In all the cases the proposed settings give very satisfactory results.

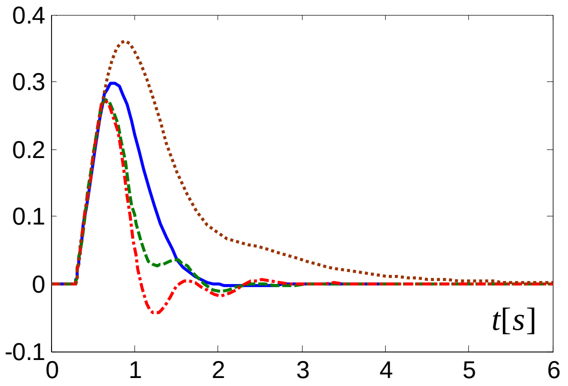


Fig. 9. $F_1(s)$ load disturbance, step response. Settings: MO (solid line), Chien (dashed), Lopez (dash-dotted), Haalman (PI) (dotted)

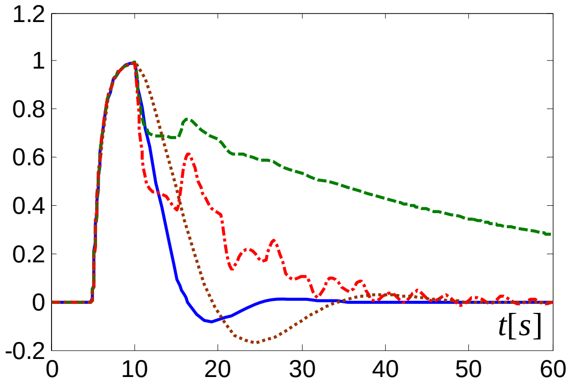


Fig. 10. $F_2(s)$ load disturbance, step response. Settings: MO (solid line), Chien (dashed), Lopez (dash-dotted), Haalman (PI) (dotted)

ACKNOWLEDGEMENTS

The work has been supported under the Czech research project MSM 0021627505.

REFERENCES

Åström, K.J., Hägglund, T. (1995). *PID Controllers: Theory, Design, and Tuning. 2nd edition*. Instrument Society of America (ISA).

Chien, K.L., Hrones, J.A., Reswick, J.B. (1952). On the automatic control of generalised passive systems. *Transactions of the ASME*, 74, 175-185.

Cohen, G.H., Coon, G.A. (1953). Theoretical Consideration of Retarded Control. *Transactions of the ASME*, 75, 827-834.

Cvejn, J. (2009). Sub-optimal PID controller settings for FOPDT systems with long dead time. *Journal of process control*, 19, 1486-1495.

Ho, W.K., Hang, C.C., Zhou, J.H. (1995). Performance and gain and phase margins of well-known PI tuning formulas. *IEEE Transactions on Control Systems Technology*, 3(2), 245-248.

Kiong, T.K., Quing-Guo, W., Chiech, H.Ch., Hägglund, T.J. (1999). *Advances in PID Control*. Springer Verlag.

Lopez, A.M., Murrill, P.W., Smith, C.L. (1967). Controller Tuning Relationships Based On Integral Performance Criteria. *Instrumentation Technology*, 14 (11).

Oldenbourg, R.C., Sartorius, H. (1956). A Uniform Approach to the Optimum Adjustments of Control Loops. In: *Frequency Response*, The Macmillan Co.

Rivera, D.E., Morari, M., Skogestad, S. (1986). Internal Model Control 4. PID Controller Design. *Industrial & Engineering Chemistry Process Design and Developement*, 1, 252-265.

Wang, F.S., Juang, W.S., Chan, C.T. (1995). Optimal tuning of PID controllers for single and cascade control loops. *Chemical Engineering Communications*, 132, 15-34.

Ziegler, J.G., Nichols, N.B. (1942). Optimum Settings for Automatic Controllers. *Transactions of the ASME*, 64, 759-768.

Haalman, A. (1965). Adjusting controllers for deadtime processes. *Control Engineering*, July-65 , 71-73.

O´Dwyer, A. (2003). *Handbook of PI and PID Tuning Rules*. Imperial College Press.

Signal shapers for BWB aircraft control

V. Kucera* M. Hromcik**

** Czech Technical University*
in Prague, FEE, Department of control engineering
(Tel:+420737438497.; e-mail: kucervl2@fel.cvut.cz)
*** Czech Technical University*
in Prague, FEE, Centre for Applied Cybernetics
(e-mail: m.hromcik@c-a-k.cz)

Abstract: This article shows results related to application of impulse signal shapers (ZV, ZVD, EI), applied on reference signal as a feed-forward vibration compensator for pilot commands incorporated with feedback control system (FCS) design for blended-wing-body aircraft (ACFA 2020). The results are nicely indicating ability of connection feed-forward approach for reducing vibration cooperation with active damping. Other goals are study signal shapers design for hard nonlinearities in elevator and ailerons like saturation and rate limiters, induce by finite rates and deflection of servomechanism of control surface. Standard design can't be directly used for higher deflection of elevator and ailerons, because nonlinearities deform shaped command. Two efficient modifications used as alternatives to standard shapers configuration, suggested in the article, permit application of the feed-forward compensation in respect of this setup.

This is preliminary papers for EUCASS conference in Petrohrad 2011

1. INTRODUCTION

Impulse shapers (PosiCast, ZV, ZVD, EI, ...) have been in the centre of attention for last two decades, [1, 2, 3], as an efficient feed-forward approach for vibration control of flexible systems, like cranes, manipulators or flexible mechanical structures [4, 5, 6]. Realisation is based on convolution of a sequence of impulses, an input shaper sets, with desired references. The shaped command cancels vibration which is responsible for excitation of the flexible modes of the aircraft. If the impulses, which is defined the shaper behaviour are chosen correctly, then the system will respond for desired reference without undesirable vibration.
Input shaper as a feed-forward controller can be regarded as a smart filter of the reference signal, an add-on to a functional reference-tracking feedback control system. If the controlled system is flexible, such our aircraft typically features are bending of the hull and wing behavior during pitching maneuver due to excitation of underlying flexible modes by set point changes (step commands), is shaper appropriate alternatives to classic low-pass filters. Therefore, posicast control can be regarded as a complementary measure in a two-degree-offreedom control scheme, when the feedback loop is closed first to guarantee robust stability, disturbance rejection and positioning, and then the input command pre-filter shapes the reference signal such that the transient response is less oscillatory.

2. SIGNAL SHAPERS FOR FLEXIBLE AIRCRAFT

Apparently, signal shapers are clear candidates for inclusion into an efficient FCS for flexible aircraft on figure 1, like the ACFA 2020 blended-wing-body design (www.acfa2020.eu).

Figure 1 The ACFA 2020 blended-wing-body aircraft

For some reason though, signal shapers are not commonly known in the flight controls area and some more traditional solutions, like „structural filters“ are routinely used – basically low-pass Chebyshev or other-type filters included in the FCS so as not to excite flexible modes. In comparison, properly tuned signal shapers, targeted at the most prominent flexible modes of the aircraft, lead to superior responsiveness and more efficient vibrations suppression.

The role and placement of a properly designed PosiCast shaper in a traditional feedback SAS (stability augmentation system) or CAS (control augmentation system) is depicted in the following scheme on figure 2(inputs: El-elevator, Ai-aileron, R-rudder). SAS/CAS is not supposed to act as flexible modes damper in the following figure. For a SAS/CAS augmented by (or integrated with) feedback active damping system, the scheme changes as figure 3 (for a particular case of roll autopilot).

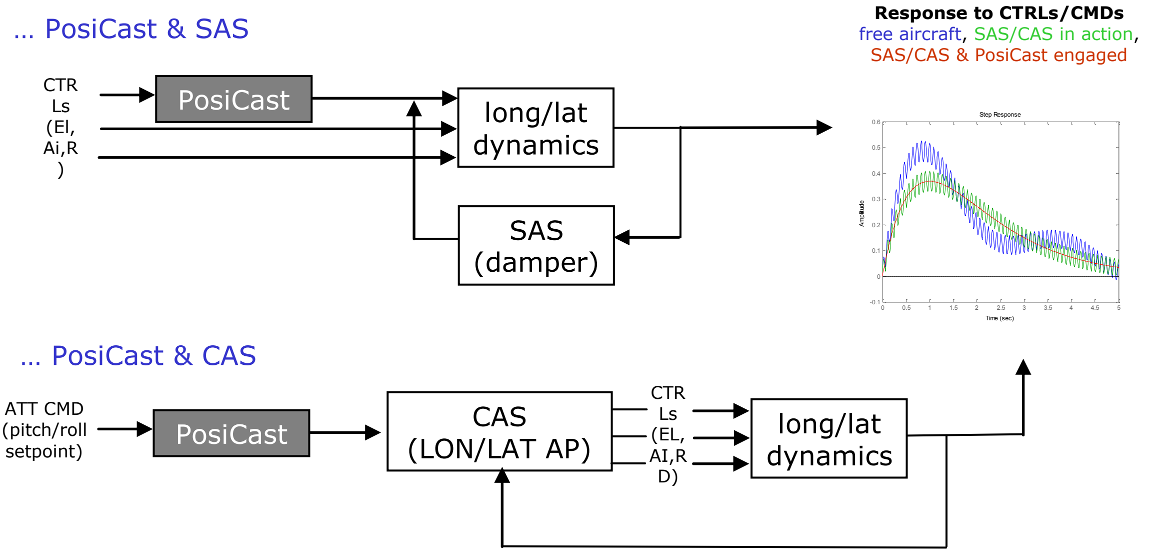


Figure 2 *Placement and function of PosiCast input command shaper in a feedback SAS/CAS systems*

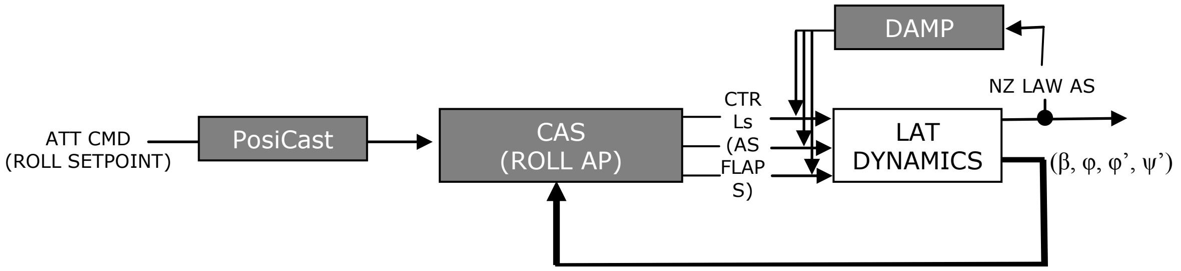


Figure 3 Posicast ***and active-damping-augmented lateral CAS***

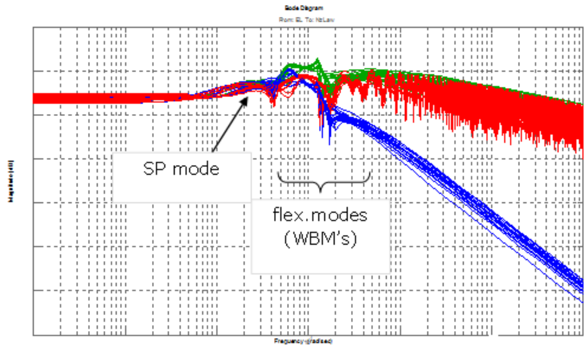


Figure 4 PosiCast and longitudinal NACRE model. Stick to NZ law wings (bode)

3. RESULTS FOR ACFA 2020 BWB AIRCRAFT

First, for the purpose of this extended abstract, a very simple SAS (SISO, stabilizing unstable longitudinal dynamics) is considered and classical PosiCast signal shaper is included in the stick-input channel. The aircraft dynamics is left as unchanged as possible obviously.
Transfer function from stick input to the wings modal sensor (accelerometer-based, Nz Law) is in the figure 4, 5, showing significant damping of wings first two flexible symmetric modes (red), compared to a free aircraft (green) and also to the Chebyshev-type structural low-pass filter (blue). Note that a two-modes (four sub-steps) version of PosiCast was designed to cover both modes simultaneously for all 18 mass cases (6 for fuel, 3 for passengers).

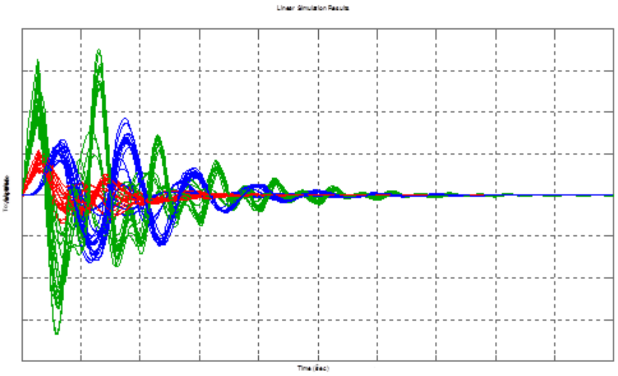


Figure 5 PosiCast and longitudinal control. Stick to modal sensor (step).

The PosiCast shaper certainly affects responsiveness of the aircraft (see the close coupling of flexible and SP modes). In any case nevertheless, it does not affect it more negatively than the structural filter. See the following figure:

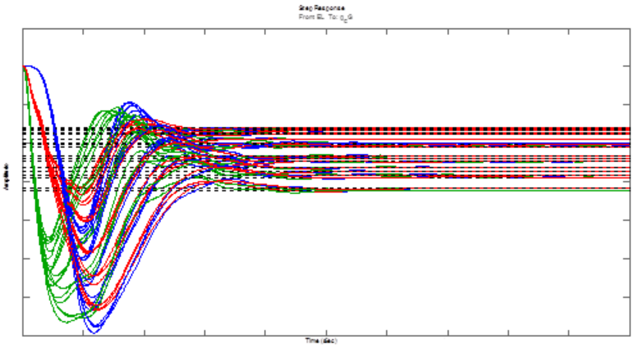


Figure 6 PosiCast and longitudinal NACRE model. Stick to q (step).

In this case, PosiCast is acting directly on the control surface signals where the rate limiters need to be taken into account. All effects described in section 4 are evidenced (posicast out of the game for step stick command and elevator deflection above 5 degrees) and the measures proposed in the next sections (like ramp split-up for higher amplitudes) lead to exactly the same results. Refer to section 4 for detailed description.

In the full paper, further results shall be reported for both longitudinal and lateral ACFA 2020 BWB controls, also in combination with active damping feedback system, figure 7.

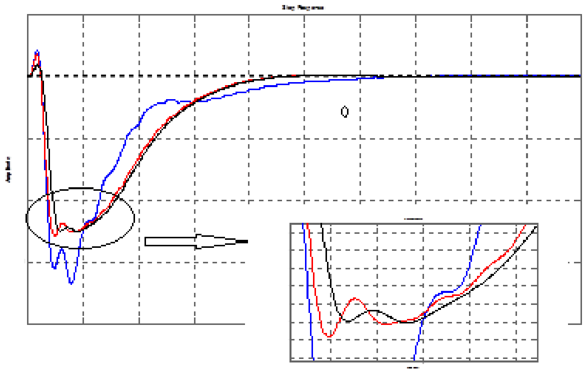


Figure 7 BWB LAT AP (blue,) augmented with active FB damper (red), and with PosiCast on top of that (black). ROLL SP to NZ LAW WINGS antisymmetric (step). Selected passangers-fuel combination (2-5)

Compare the modal sensor reading (LAT CAS, step for roll-angle setpoint), for FCS only (red), FCS+feedback active damper (blue), and with PosiCAST shaper on top of that all. Contribution of the signal shaper in this setup, for reduction of vibrations caused by the automatically engaged maneuver, is obvious. In addition, responses of the aircraft in all cases are almost identical:

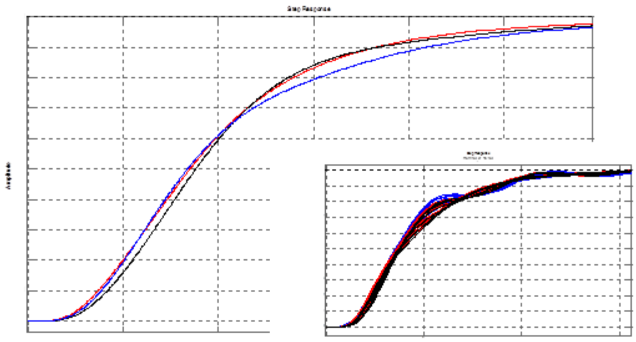


Figure 8 BWB LAT AP (blue,) augmented with active FB damper (red), and with PosiCast on top of that (black). ROLL SP to ROLL ANGLE (step). Selected passangers-fuel combination (2-5, all cases (sub-figure))

4. SIGNAL SHAPERS AND RATE LIMITERS

As shown, delay-based input shaper, like zero vibration and extra insensitive (ZV, EI), can be effectively used as a feed-forward reference filter applied to pilot command in order to reduce wing bending and vertical bending of hull during a maneuver. This strategy is nevertheless strongly limited by the rate limiter nonlinearity (standing for finite servos rates), having substantial, amplitude-dependent filtering effect on the input signal. This observation can be interpreted both in the frequency and time domain terms. Speaking in frequency-domain words, rate limiter acts as a low-pass filter, with cut-off frequency strongly dependent on the amplitude of the input signal on figure 9.

The higher the amplitude is, the stronger filtering effect arises. By inspection of the dependency on figure 9, for elevator commands greater than five degrees (cut-off approx. 15 rad/s for five degrees amplitude), the influence of the delay-based filters on the command signal is strongly weakened by the rate limiter at the higher frequency of the HBM Considering these facts, filters cannot in principle be successful for all cases

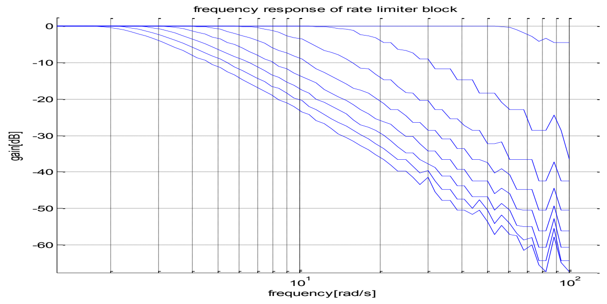


Figure 9 Frequency responses of a 30deg/s rate limiter for varying signal amplitude.

By inspection of the dependency on figure 9, for elevator commands greater than five degrees (cut-off approx. 15 rad/s for five degrees amplitude), the influence of the delay-based filters on the command signal is strongly weakened by the rate limiter, figure 11, at the higher frequency of the HBM Considering these facts, filters cannot in principle be successful for all cases.

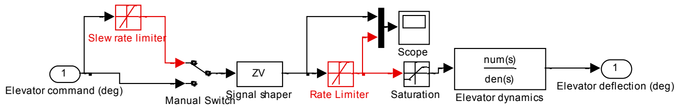


Figure 10 Configuration

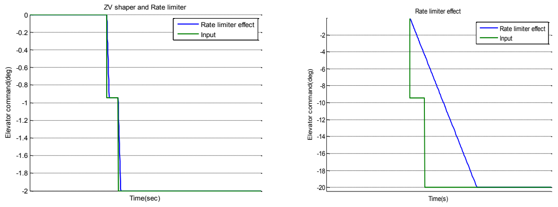


Figure 11 Effect of rate limiter

For this reason, two approaches are suggested. In both cases, the shaped signal is modified in such a way that it becomes tractable through the subsequent rate-limiter block without distortion (unlike the pure signal shaper output itself.

The first approach is based on splitting the ramp signal, coming out from a rate-limiter block as a response to step, artificially, for a time-delay slightly smaller than the ZV shaper-suggested value. This leads to a fair reduction of the HBM peak, figure 12. In this particular case, the shaped reference in the figure 12 for the 20 deg. elevator command, which has transfer function (1),

$$G_{mod} = \frac{1}{spoint}\left(\frac{rate}{s}e^{\frac{-spoint.rate}{2}s}\right) - \left(\frac{rate}{s}e^{-\left(\frac{spoint}{rate}+delay\right)s}\right) + \left(\frac{rate}{s}e^{-\left(\frac{spoint.rate}{2}+delay\right)s}\right) \qquad (1)$$

where spoint is set point, rate is setting of rate limiter and delay is value from posicast approach.

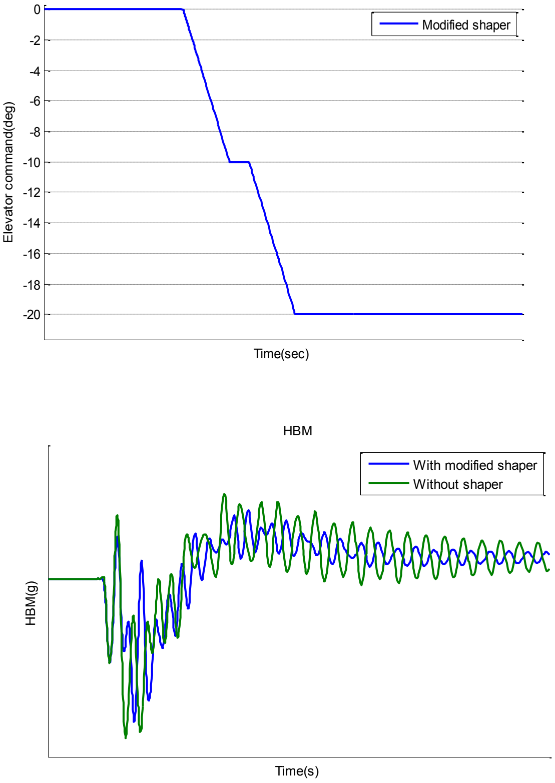


Figure 12 Modified reference shaper and effect on hull bending mode

This command is fully accepted by rate limiters without any distortion. The filter G_{mod} is, unfortunately, parameterized by the amplitude of the step, so it is not a constant, or time-invariant system.

Alternatively, the following procedure can be applied. The main idea is to attach an additional rate limiter, with the same setting as the one representing servos, in front delay-based shaper, figure 10. The modified filtered command is obviously accepted by (passed-through) the finite-rate servos without any distortion, the red line on figure 13, and it does not contain frequencies corresponding to flexible modes of aircraft (as the signal shaper is in the command line). The results of the hull bending sensors on figure 14 show power of this method, where the green line isn't treat by new approach and blue(ZV) and red(EI) line is for different used shapers with naturally adaptive rate limiter before.

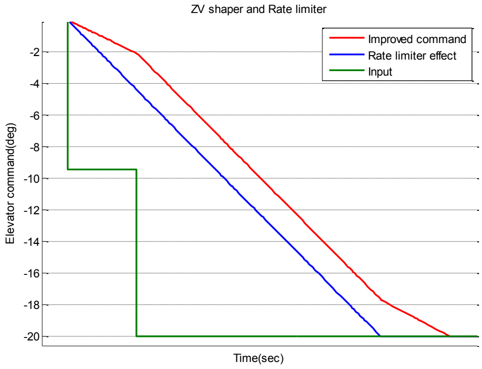


Figure 13 Effect of Rate limiter on shaped reference

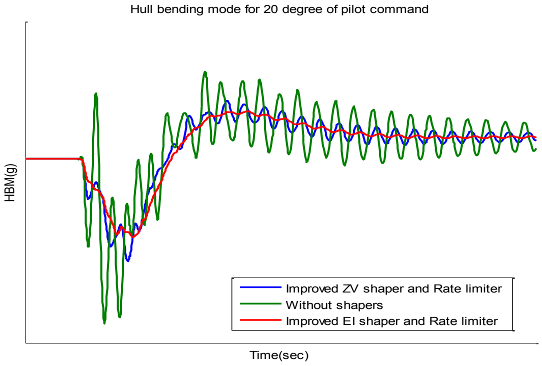


Figure 14 Bending of hull for 20 degree of reference command

CONCLUSIONS

Results will be further developed and applied for the case study of large flexible blended-wing-body aircraft, similarly to section 3 (for the classical signal shapers designs). Data come from the ongoing European project ACFA 2020. ACFA 2020 (Active Control for Flexible Aircraft, www.acfa2020.eu) is a collaborative research project funded by the European Commission under the seventh research framework programme (FP7). The project deals with innovative active control concepts for ultra-efficient 2020 aircraft configurations like the blended wing body (BWB) aircraft.

REFERENCES

[1] Smith, O.J.M. "Posicast control of damped oscillatory systems Proceedings of the IRE col 45", September 1957, pp 1249-1255.

[2] Singer, N.C. a Seering, "W.P. Preshaping command input to reduce system vibration", Journal of Dynamics, System, Measure and Control, vol 112. March 1990, pp 76-82.

[3] W. E. Singhose, S. Derezinski, N. C. Singer, "Extra-insensitive input shapers for controlling flexible spacecraft", Journal of Guidance, Control, and Dynamics, vol. 19, no. 2, pp. 285-391, 1996.

[4] Bing Li, Xuping Zhang, J.K. Mills, W.L. Cleghorn, Liyang Xie, "Vibration Suppression of a 3-PRR Flexible Parallel Manipulator Using Input Shaping " , International Conference on Mechatronics and Automation, August 9-12, 2009, China

[5] M.A. Ahmed, M.S. Ramli, R.M.T. Raja Ismail, N.Hanbali, M.A. Zawawi," The investigation of input shaping with optimal state feedback for vibration control of flexible joint manipulator" ,Conference on Innovative Technologies in Intelligent Systems and Industrial Applicaton, 25th & 26th July 2009, Malaysia

[6] W. E. Singhose, S. Derezinski, N. C. Singer, "Extra-insensitive input shapers for controlling flexible spacecraft", Journal of Guidance, Control, and Dynamics, vol. 19, no. 2, pp. 285-391, 1996.

Time Sub-Optimal Control of Triple Integrator

P. Bistak *

**Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology, Bratislava, Slovakia, (e-mail: pavol.bistak@stuba.sk)*

Abstract: The time sub-optimal control is studied in this paper. The nonlinear controller that respects input saturations is derived for the simple linear system represented by the triple integrator. In comparison with the pure time optimal controller the designed sub-optimal controller changes its limit values smoothly with exponential behaviour. Similarly to the time optimal control the design is based on switching surfaces but these are shifted and modified according to the original ones in the time optimal control. This can assure the decrease of high sensitivity of time optimal control. New parameters introduced during the design correspond in linear cases to the poles of the closed loop system. They enable to tune the control changes. The time sub-optimal control is compared with model predictive control. The resulting formulas for the control value are complicated but they have an explicit form so they can be evaluated fast enough to be used in real time systems.

Keywords: time optimal control, input constraints, smooth switching, triple integrator.

1. INTRODUCTION

The optimality principle played an always an important role in the design of control circuits. In the previous century the optimal control was studied heavily in 50-ties and 60-ties (Athans and Falb, 1996). The real applications have shown that it is very sensitive to unmodelled dynamics, parametric variations, disturbances and noise. Therefore it was in the main-stream control strategies replaced by other techniques, e.g. pole assignment control. This allowed to choose the position of poles and so influence the speed of changes what enabled to decrease the sensitivity of a control circuit. This paper shows how it is possible to combine “fast” time optimal control with “slow” pole assignment control.

Generally the time optimal control can be solved by computation of switching surfaces. There are several ways how to derive them. They can result from the Pontryagin’s maximum principle. Pavlov solved switching surfaces for the systems up to the third order from the phase trajectories (Pavlov, 1966). Switching surfaces can also be expressed by the set of algebraic equations (Walther et al., 2001) that results from the time solution in the phase space. Of course, for higher order systems it can be rather complicated to find the exact solution of such a set. This paper shows how it is possible for a simple linear system represented by the triple integrator to derive and solve a set of polynomial equations in order to get the control value in the exact form.

The controller proposed in this paper is not pure time optimal control. As it has been mentioned above it tries to combine qualities of both time optimal and pole assignment control (Huba, 2006). The time optimal control belongs to the nonlinear class of controllers whereas the pole assignment control is typically linear type of control. Switching between these two classes of control is assured by the saturation function that is applied individually to each mode of control. In this paper a set of additional parameters is introduced that corresponds to the set of poles of the closed loop in the linear case. These parameters specify exponential changes from one limit control value to the opposite one. By introducing such parameters in the control design the original switching surfaces valid for the time optimal control are modified. They are not smooth and consist of several regions. Then it is difficult to identify the corresponding region for an initial state. After deciding for the right region the control law is calculated according to the position of the representative point expressing the actual state with respect to the corresponding region of the switching surface.

The paper is organized in five chapters. After introduction and problem statement chapters there is the main chapter where the design of the sub-optimal controller is described in details. This chapter discusses the nonlinear dynamics decomposition and regions of the switching surface. There is a corresponding control law derived for each region. The fourth chapter shows time responses of the designed controller and compares it to the time optimal controller and model predictive controller. The paper is finished with short conclusions.

2. PROBLEM STATMENT

Let us consider the linear system given in the state space

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{b}u \\ y &= \mathbf{c}^t\mathbf{x}\end{aligned} \qquad \mathbf{A} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix}0\\0\\1\end{bmatrix} \qquad \mathbf{c}^t = \begin{bmatrix}1 & 0 & 0\end{bmatrix} \tag{1}$$

that represents the triple integrator. The control input signal is constrained

$$u = \langle U_1 \quad U_2 \rangle \tag{2}$$

The task is to design the time sub-optimal controller what means to drive the system from an initial state $\mathbf{x} = [x \quad y \quad z]^t$ to the desired state $\mathbf{x}_w$ in a minimum time $t_{\min}$ under the additional condition that limits the changes of the control action between two opposite values. When it is required that these changes should have an exponential behaviour the additional condition can be expressed by a scalar function $\varsigma_i : R^n \to R$ representing the distance of the current state $\mathbf{x}$ from the switching surface (curve, point) and it holds

$$\frac{d\varsigma_i}{dt} = \alpha_i \varsigma_i \ , \ \alpha_i \in R^- \ , \ i = 1,2,3 \tag{3}$$

For the sake of simplicity we should admit that using a coordinate transformation it is always possible to set the desired state equal to the origin $\mathbf{x}_w = \mathbf{0}$.

3. TIME SUB-OPTIMAL CONTROLLER DESIGN

It is well known that minimum time optimal control with saturated input leads to the control action with at most n intervals switching between limit values where n represents the order of the system. Usually the control algorithm results in deriving switching surfaces as functions of states which signs determine the switching times. It can be very hard task to express these functions exactly and there is no general solution for higher order systems ($n > 3$). Bang-bang control in practice is not desirable because of chattering and noise effects but there are techniques have to cope with them (Pao and Franklin, 1993, Bistak et al. 2005).

The presented sub-optimal controller design belongs to one of them. This time the control action will not be calculated as the sign of the switching surface but will result from (3). If we apply the condition (3) also for the switching curve and switching point this will influence the construction of the switching surface itself. We will explain it with the help of a state vector nonlinear decomposition.

3.1 Nonlinear Decomposition

Let us consider ordered coefficients

$$\alpha_3 < \alpha_2 < \alpha_1 < 0 \tag{4}$$

Then the eigenvectors

$$\mathbf{v}_i = [\alpha_i \mathbf{I} - \mathbf{A}]^{-1} \mathbf{b} = \begin{bmatrix} \dfrac{1}{\alpha_i^3} & \dfrac{1}{\alpha_i^2} & \dfrac{1}{\alpha_i} \end{bmatrix}^t \tag{5}$$

form a base of the state space. In the linear case any point of the state space can be expressed as

$$\mathbf{x} = q_1 \mathbf{v}_1 + q_2 \mathbf{v}_2 + q_3 \mathbf{v}_3 \ , \ q_1, q_2, q_3 \in R \tag{6}$$

Because the control signal is limited only the points where $u = \sum_{i=1}^{3} q_i$ fulfils (2) are covered by (6). In order to express the whole space we have to introduce the nonlinear decomposition of the state

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3 \tag{7}$$

where each mode $\mathbf{x}_i$

$$\mathbf{x}_i = e^{\mathbf{A}t_i} q_i \mathbf{v}_i + \int_0^{t_i} e^{\mathbf{A}(t-\tau)} \mathbf{b} q_i(\tau) d\tau \tag{8}$$

consists of a linear part given by the parameter q_i

$$q_i \in \left\langle U_1 - \sum_{k=1}^{i-1} q_k \quad U_2 - \sum_{k=1}^{i-1} q_k \right\rangle \quad when \quad t_i = 0 \tag{9}$$

and a nonlinear part specified by the parameter t_i

$$0 < t_i < t_{i-1} \ \ when \ \ q_i = U_{3-j} - \sum_{k=1}^{i-1} q_k \ \ , t_0 = \infty \ , \ j = 1,2 \tag{10}$$

After substituting (1) and (5) into (8) one gets

$$\mathbf{x}_i = \begin{bmatrix} \dfrac{1}{\alpha_i^3} - \dfrac{t_i}{\alpha_i^2} + \dfrac{t_i^2}{2\alpha_i} - \dfrac{t_i^3}{6} \\ \dfrac{1}{\alpha_i^2} - \dfrac{t_i}{\alpha_i} + \dfrac{t_i^2}{2} \\ \dfrac{1}{\alpha_i} - t_i \end{bmatrix} q_i \tag{11}$$

If we take the subsystem $\mathbf{x}_1$ it represents a one-dimensional variety that corresponds to the switching curve. Points from the linear part of $\mathbf{x}_1$ where (9) is fulfilled satisfy (3), i.e. they are decreasing the distance ς_1 from the origin. In this case the system is moving along the line. The other points of the subsystem $\mathbf{x}_1$ given by (10) could not fulfil (3) because of the limited control value (2). They are approaching the linear part of $\mathbf{x}_1$ with the limit control value so they are moving along the trajectory in the form of a curve. In this case (3) is superimposed by (2).

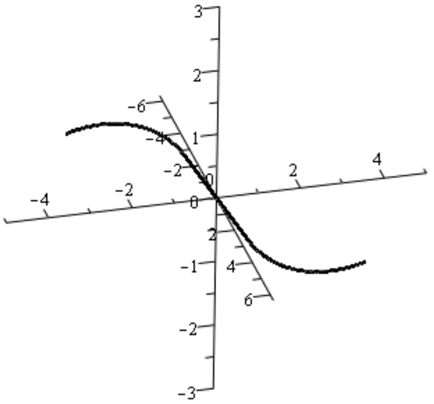


Fig. 1. Subsystem $\mathbf{x}_1$ representing the switching curve

Similarly we can create a two-dimensional variety that will express the switching surface. We simply add to the subsystem $\mathbf{x}_1$ the subsystem $\mathbf{x}_2$. The points of the subsystem $\mathbf{x}_1$ become the target points for the second subsystem $\mathbf{x}_2$.

$$\mathbf{x}_{12} = \mathbf{x}_1 + \mathbf{x}_2 \tag{12}$$

This time we define the distance ς_2 in the direction of the second eigenvector $\mathbf{v}_2$. The points of the $\mathbf{x}_2$ try to reach the $\mathbf{x}_1$ points according to (3) if it does not break (2). Otherwise they are moving with the limit control value q_2 given by (10).

Again there exist a linear and a nonlinear parts of the subsystem $\mathbf{x}_2$. In combination with the previous subsystem $\mathbf{x}_1$ we get four possibilities, i.e. four regions of the switching surface with respect to the limit and nonzero values of q_1, q_2, t_1, t_2 (Fig. 2). If we take into account the parameter $j = 1,2$ the number of the regions doubles. Later on we will describe these regions in details and derive for each of them the corresponding control value.

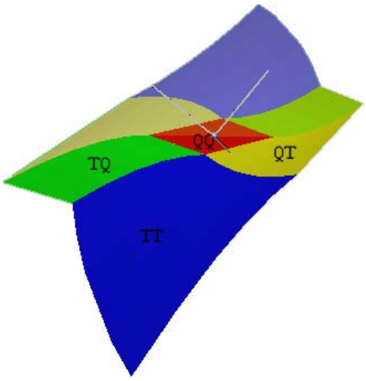


Fig. 2. Regions of the switching surface (denoted for *j*=1)

To cover the whole space we should realize also the third subsystem $\mathbf{x}_3$ but in the presented control algorithm design it is not necessary. To give reasonable results that can be applied in real time applications we simplified the third subsystem to following one

$$\mathbf{x}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} q_3 \quad , \quad q_3 \in R \tag{13}$$

It represents the unit vector in the direction of the x-axis multiplied by the quotient q_3. Thus the quotient q_3 expresses the distance ς_3 between the current state and the switching surface that is measured in the direction of the x-axis. This simplification enables easier localization of the initial state with respect to the regions of the switching surface because it represents the projection of the switching surface to the (y,z)-plane where the borders between regions are parabolic curves or lines.

In (Ťapák et al., 2006) one can find the solution of the control algorithm when the third subsystem was given by the third eigenvector $\mathbf{v}_3$ multiplied by the quotient q_3

$$\mathbf{x}_3 = q_3 \mathbf{v}_3, q_3 \in R \tag{14}$$

but this was not in the form suitable for real time systems.

After completely decomposing the system to the three subsystems (7) it is necessary to derive the formula of the corresponding region of the switching surface. This comes from the set of equations (12) when the parameters t_i or q_i are evaluated from the last two equations and replaced in the first one. Then one gets the formula for the corresponding switching surface in the form

$$x = f(y,z) \;\; ; \;\; f : R^2 \to R \tag{15}$$

Now the resulting control value can be computed from (3) when we realize that the distance $\varsigma_3 = q_3$ can be expressed as the difference between the x-coordinate of the initial point and the x-coordinate of the switching surface given by (15) in the form $f(y,z)$

$$\varsigma_3 = x - f(y,z) \tag{16}$$

After substituting (16) into (3) and taking into account (1) it results in

$$\begin{aligned} \frac{d\varsigma_3}{dt} = \frac{dx}{dt} - \frac{df(y,z)}{dt} = y - \frac{df(y,z)}{dy} z - \frac{df(y,z)}{dz} u = \\ = \alpha_3 (x - f(y,z)) \end{aligned} \tag{17}$$

Finally the control value u can be isolated

$$u = \frac{-\alpha_3 (x - f(y,z)) + y - \dfrac{df(y,z)}{dy} z}{\dfrac{df(y,z)}{dz}} \tag{18}$$

The resulting control value u must be limited by (2).

As one can see from (18) the only one term not evaluated yet is $f(y,z)$ representing the switching trajectory. Because it differs according to the regions of the switching trajectory we will evaluate it individually.

3.2 Control for Region QQ

The region QQ denotes the subset of (12) where both subsystems $\mathbf{x}_1$ and $\mathbf{x}_2$ are in the linear cases, i.e. (9) is fulfilled for q_1 and q_2. The parameters q_1 and q_2 can be evaluated from the last two equations of the set (12). After using (11) and substituting $t_1 = t_2 = 0$ into (12) one gets

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dfrac{q_1}{\alpha_1^3} + \dfrac{q_2}{\alpha_2^3} \\ \dfrac{q_1}{\alpha_1^2} + \dfrac{q_2}{\alpha_2^2} \\ \dfrac{q_1}{\alpha_1} + \dfrac{q_2}{\alpha_2} \end{bmatrix} \tag{19}$$

And for parameters q_1 and q_2 it yields

$$q_1 = -\frac{\alpha_1^2 (z - y\alpha_2)}{\alpha_2 - \alpha_1} \tag{20}$$

$$q_2 = \frac{\alpha_2^2 (-\alpha_1 y + z)}{\alpha_2 - \alpha_1} \tag{21}$$

By the substitution of (20) and (21) into the first equation of (19) we derive the analytical expression for the region QQ

$$x = f(y,z) = -\frac{z - y\alpha_2}{\alpha_1(\alpha_2 - \alpha_1)} + \frac{-\alpha_1 y + z}{\alpha_2(\alpha_2 - \alpha_1)} \tag{22}$$

According to (18) the control value u results in the form

$$\begin{aligned} u = {} & -y\alpha_1\alpha_2 + \alpha_2 z + \alpha_1 z + \alpha_3 x\alpha_1\alpha_2 \\ & - \alpha_3 y\alpha_2 - \alpha_3\alpha_1 y + \alpha_3 z \end{aligned} \tag{23}$$

This is the well-known linear pole assignment controller for the triple integrator.

3.3 Control for Region TQ

By TQ we denote the region of the switching surface when the subsystem $\mathbf{x}_1$ is in the nonlinear cases, i.e. for its states (10) is valid and the subsystem $\mathbf{x}_2$ is in the linear cases, i.e. (9) comes true. The procedure how to derive the control value is very similar to that one performed in the previous region QQ. First we express (12) when $q_1 = U_j$ and $t_2 = 0$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} U_j \left(\frac{1}{\alpha_1^3} - \frac{t_1}{\alpha_1^2} + \frac{1}{2}\frac{t_1^2}{\alpha_1} \right) - \frac{1}{6} U_j t_1^3 + \frac{q_2}{\alpha_2^3} \\ U_j \left(\frac{1}{\alpha_1^2} - \frac{t_1}{\alpha_1} \right) + \frac{1}{2} U_j t_1^2 + \frac{q_2}{\alpha_2^2} \\ \frac{U_j}{\alpha_1} - U_j t_1 + \frac{q_2}{\alpha_2} \end{bmatrix} \tag{24}$$

For this and following calculations we have used the Maple computer algebra system and because of the complexity of several expressions we have used the Maple outputs.

Now it is necessary to solve the last two equations of the set (24). The difference consists in that the second equation of the set (24) is now the quadratic equation. From its two solutions we have chosen such one that assures the positive value of t_2. Then after introducing the notation for the discriminant DTQ

$$DQT = -2\,\alpha_1^2\,U_j\,z\,\alpha_2 + \alpha_1^2\,U_j^2 - U_j^2\,\alpha_2^2 + 2\,y\,\alpha_1^2\,\alpha_2^2\,U_j \tag{25}$$

the parameters q_2 and t_1 it can be expressed

$$q_2 = \frac{z\,\alpha_1\,\alpha_2 - \alpha_1\,U_j + \mathit{sign}(U_j)\,\sqrt{DTQ}}{(U_{3-j} - U_j)\,\alpha_1} \tag{26}$$

$$t_1 = -\frac{-U_j\,\alpha_2 + \alpha_1\,U_j - \mathit{sign}(U_j)\,\sqrt{DTQ}}{U_j\,\alpha_1\,\alpha_2} \tag{27}$$

Again we substitute (26) and (27) into the first equation of (24) and get the expression for the region TQ

$$\begin{aligned} x = {} & \frac{1}{6}\,\frac{1}{U_j^2\,\alpha_1^3\,\alpha_2^3}\Big(2\,U_j^3\,\alpha_2^3 + 3\,U_j^3\,\alpha_2^2\,\alpha_1 - 3\,U_j^2 \\ & \alpha_2^2\,\mathit{sign}(U_j)\,\sqrt{DTQ} - 5\,\alpha_1^3\,U_j^3 + 3\,\alpha_1^2\,U_j^2\,\mathit{sign}(U_j)\,\sqrt{DTQ} \\ & + 3\,\alpha_1\,U_j\,\mathit{sign}(U_j)^2 DTQ - \mathit{sign}(U_j)\,DTQ^{3/2} + 6\,\alpha_1^3\,U_j^2\,z\,\alpha_2\Big) \end{aligned} \tag{28}$$

From (18) the control value u is

$$\begin{aligned} u = {} & \frac{1}{3}\,\frac{1}{\alpha_2^2\,\alpha_1^4\,(-y\,\alpha_2 + z)}\Big(-3\,\alpha_2^4\,\alpha_1^4\,z\,y + 2\,\alpha_3\,U_j^2\,\alpha_2^2\,\alpha_1^2 - \alpha_3\,U_j \\ & \alpha_2^3\,\alpha_1^2\,z + \alpha_3\,U_j\,\alpha_2^4\,y\,\alpha_1^2 + \alpha_3\,\alpha_1^4\,U_j\,z\,\alpha_2 - \alpha_3\,\alpha_1^4\,U_j\,y\,\alpha_2^2 + 2\,\alpha_3 \\ & \alpha_1^4\,z^2\,\alpha_2^2 - \alpha_3\,U_j^2\,\alpha_2^4 - \alpha_3\,\alpha_1^4\,U_j^2 - 3\,y\,\mathit{sign}(U_j)\,\sqrt{DTQ}\,\alpha_1^3\,\alpha_2^3 \\ & - \alpha_3\,\mathit{sign}(U_j)\,\sqrt{DTQ}\,U_j\,\alpha_2^3 + \alpha_3\,\mathit{sign}(U_j)\,\sqrt{DTQ}\,\alpha_1^3\,U_j \\ & - 4\,\alpha_3\,\alpha_1^4\,z\,\alpha_2^3\,y + 2\,\alpha_3\,y^2\,\alpha_1^4\,\alpha_2^4 + 3\,\alpha_2^2\,\alpha_1^3\,z\,\mathit{sign}(U_j)\,\sqrt{DTQ} \\ & - 3\,\alpha_3\,\mathit{sign}(U_j)\,\sqrt{DTQ}\,\alpha_1^3\,y\,\alpha_2^2 + 3\,\alpha_3\,\mathit{sign}(U_j)\,\sqrt{DTQ}\,x\,\alpha_1^3 \\ & \alpha_2^3 + 3\,\alpha_2^3\,\alpha_1^4\,z^2\Big) \end{aligned} \tag{29}$$

3.4 Control for Region TT

The region TT denotes the subset of (12) where both subsystems $\mathbf{x}_1$ and $\mathbf{x}_2$ are in the nonlinear cases, i.e. (10) is fulfilled for t_1 and t_2. Again the parameters t_1 and t_2 can be computed from the last two equations of the (12). This time we substitute $q_1 = U_j$ and $q_2 = U_{3-j} - U_j$ into (12)

$$\begin{aligned} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \Bigg[\bigg[& U_j \left(\frac{1}{\alpha_1^3} - \frac{t_1}{\alpha_1^2} + \frac{1}{2}\frac{t_1^2}{\alpha_1} \right) - \frac{1}{6} U_j t_1^3 + (U_{3-j} \\ & - U_j) \left(\frac{1}{\alpha_2^3} - \frac{t_2}{\alpha_2^2} + \frac{1}{2}\frac{t_2^2}{\alpha_2} \right) - \left(\frac{1}{6} U_{3-j} - \frac{1}{6} U_j \right) t_2^3 \bigg], \\ & \bigg[U_j \left(\frac{1}{\alpha_1^2} - \frac{t_1}{\alpha_1} \right) + \frac{1}{2} U_j t_1^2 + (U_{3-j} - U_j) \left(\frac{1}{\alpha_2^2} - \frac{t_2}{\alpha_2} \right) \\ & + \left(\frac{1}{2} U_{3-j} - \frac{1}{2} U_j \right) t_2^2 \bigg], \\ & \bigg[\frac{U_j}{\alpha_1} - U_j t_1 + \frac{U_{3-j} - U_j}{\alpha_2} - (U_{3-j} - U_j)\,t_2 \bigg]\Bigg] \end{aligned} \tag{30}$$

and when solving the last two equations again the criterion for the choice of the right solution is that the times t_1 and t_2 must be positive.

$$\begin{aligned} DTT = {} & -\alpha_2^2\,\alpha_1^2\,U_{3-j}\,z^2\,U_j + \alpha_2^2\,U_j^2\,z^2\,\alpha_1^2 - \alpha_1^2\,U_{3-j}^3\,U_j + 2\,U_j^2\,\alpha_1^2 \\ & U_{3-j}^2 + 2\,U_{3-j}^2\,y\,\alpha_1^2\,\alpha_2^2\,U_j - U_{3-j}^2\,U_j^2\,\alpha_2^2 - 2\,U_{3-j}\,U_j^2\,y\,\alpha_1^2 \\ & \alpha_2^2 + U_{3-j}\,U_j^3\,\alpha_2^2 - U_{3-j}\,U_j^3\,\alpha_1^2 \end{aligned} \tag{31}$$

$$t_1 = \frac{-U_j\,U_{3-j}\,\alpha_2 + \alpha_1\,U_j\,z\,\alpha_2 + \sqrt{DTT}}{U_j\,\alpha_1\,U_{3-j}\,\alpha_2} \tag{32}$$

$$\begin{aligned} t_2 & \\ & = \frac{1}{\alpha_2\,\alpha_1\,U_{3-j}\,(U_{3-j} - U_j)}\Big(\alpha_1\,U_{3-j}\,z\,\alpha_2 - \alpha_1\,U_j\,z\,\alpha_2 \\ & \quad + U_j\,\alpha_1\,U_{3-j} - \alpha_1\,U_{3-j}^2 - \sqrt{DTT}\Big) \end{aligned} \tag{33}$$

After using (32) and (33) in the first equation of the set (30) the points of the region TT can be expressed

$$
\begin{aligned}
x = {} & \frac{1}{6}\,\frac{1}{U_{3-j}^2\,\alpha_2^3\,\alpha_1^3\,U_j^2\,\left(-U_{3-j}+U_j\right)^2}\Big(-2\,\alpha_2^3\,U_j^3\,z^3\,\alpha_1^3\,U_{3-j} + {} \\
& \alpha_2^3\,U_j^4\,z^3\,\alpha_1^3 + 3\,\alpha_2^3\,U_j^3\,U_{3-j}^3\,z\,\alpha_1 - 6\,\alpha_2^3\,U_j^4\,U_{3-j}^2\,z\,\alpha_1 + 3\,\alpha_2^3 \\
& U_j^5\,U_{3-j}\,z\,\alpha_1 - 3\,U_j^5\,\alpha_2\,z\,\alpha_1^3\,U_{3-j} + U_j^2\,\alpha_2^3\,\alpha_1^3\,U_{3-j}^2\,z^3 + 3 \\
& U_j^2\,\alpha_2\,\alpha_1^3\,U_{3-j}^4\,z - 9\,U_j^3\,\alpha_2\,\alpha_1^3\,U_{3-j}^3\,z + 9\,U_j^4\,\alpha_2\,\alpha_1^3\,U_{3-j}^2\,z \\
& + 3\,\alpha_2\,U_j\,z\,\alpha_1\,DTT\,U_{3-j} - 3\,\alpha_2\,U_j^2\,z\,\alpha_1\,DTT + DTT^{3/2}\,U_{3-j} \\
& - 2\,DTT^{3/2}\,U_j + 2\,\alpha_2^3\,U_j^3\,U_{3-j}^4 - 4\,\alpha_2^3\,U_j^4\,U_{3-j}^3 + 2\,\alpha_2^3\,U_j^5 \\
& U_{3-j}^2 - 6\,U_j^3\,\alpha_1^3\,U_{3-j}^4 + 6\,U_j^4\,\alpha_1^3\,U_{3-j}^3 + 2\,U_j^2\,\alpha_1^3\,U_{3-j}^5 - 2 \\
& U_j^5\,\alpha_1^3\,U_{3-j}^2 + 3\,\alpha_2^2\,U_j^4\,U_{3-j}\sqrt{DTT} - 3\,U_j^2\,\alpha_1^2\,U_{3-j}^3\sqrt{DTT} \\
& + 6\,U_j^3\,\alpha_1^2\,U_{3-j}^2\sqrt{DTT} - 3\,U_j^4\,\alpha_1^2\,U_{3-j}\sqrt{DTT} + 3\,\alpha_2^2\,U_j^2 \\
& U_{3-j}^3\sqrt{DTT} - 6\,\alpha_2^2\,U_j^3\,U_{3-j}^2\sqrt{DTT}\Big)
\end{aligned}
\tag{34}
$$

In this case the resulting control value resulting from (18) is the most complicated one

$$
\begin{aligned}
u = {} & \frac{1}{3}\Big(-6\,U_{3-j}^2\,\alpha_3\sqrt{DTT}\,x\,\alpha_2^3\,\alpha_1^3 - 4\,U_{3-j}^2\,\alpha_3\,\alpha_2^4\,z^2\,\alpha_1^4\,y + 12 \\
& U_{3-j}^2\,U_j\,\alpha_1^4\,\alpha_2^4\,z\,y + 6\,U_{3-j}^3\,\alpha_3\,U_j\,\alpha_1^4\,y\,\alpha_2^2 + 2\,U_{3-j}^3\,\alpha_3\,U_j\,\alpha_2^4\,y \\
& \alpha_1^2 - U_{3-j}^2\,\alpha_3\,\alpha_2^4\,U_j\,z^2\,\alpha_1^2 + 2\,U_{3-j}^2\,\alpha_3\,U_j^2\,\alpha_2^4\,y\,\alpha_1^2 - 8 \\
& U_{3-j}^2\,\alpha_3\,U_j\,y^2\,\alpha_1^4\,\alpha_2^4 - 3\,U_{3-j}^2\,\alpha_3\,U_j\,\alpha_2^2\,z^2\,\alpha_1^4 - 2\,U_{3-j}^2\,\alpha_3\,U_j^2 \\
& \alpha_1^4\,y\,\alpha_2^2 + U_{3-j}\,\alpha_3\,U_j^2\,\alpha_2^2\,z^2\,\alpha_1^4 - U_{3-j}\,\alpha_3\,U_j^2\,\alpha_2^4\,z^2\,\alpha_1^2 + \alpha_3 \\
& \alpha_1^4\,U_{3-j}^5 - 2\,\alpha_3\sqrt{DTT}\,\alpha_2^3\,z^3\,\alpha_1^3 - 2\,\alpha_3\,U_j\,\alpha_2^4\,z^4\,\alpha_1^4 + 6\,U_{3-j}^2 \\
& \alpha_1^3\,\alpha_2^3\,y\sqrt{DTT} - 6\,U_{3-j}\,\alpha_1^3\,\alpha_2^3\,z^2\sqrt{DTT} - 6\,U_{3-j}^3\,\alpha_1^4\,\alpha_2^4\,z\,y \\
& - 4\,U_{3-j}^4\,\alpha_3\,\alpha_1^4\,y\,\alpha_2^2 + 2\,U_{3-j}^3\,\alpha_3\,\alpha_2^2\,z^2\,\alpha_1^4 + 4\,U_{3-j}^3\,\alpha_3\,y^2\,\alpha_1^4 \\
& \alpha_2^4 + U_{3-j}\,\alpha_3\,\alpha_2^4\,z^4\,\alpha_1^4 - 6\,U_{3-j}^3\,U_j\,\alpha_1^4\,\alpha_2^2\,z + 3\,U_{3-j}^2\,U_j^2\,\alpha_1^4 \\
& \alpha_2^2\,z - 3\,U_{3-j}^2\,U_j^2\,\alpha_1^2\,\alpha_2^4\,z - 6\,U_{3-j}\,U_j\,\alpha_1^4\,\alpha_2^4\,z^3 - 2 \\
& U_{3-j}^2\,\alpha_3\sqrt{DTT}\,U_j\,\alpha_1^3 + 2\,U_{3-j}^2\,\alpha_3\sqrt{DTT}\,U_j\,\alpha_2^3 - {} \\
& U_{3-j}^4\,\alpha_3\,U_j\,\alpha_1^2\,\alpha_2^2 + 3\,U_{3-j}^3\,\alpha_3\,U_j^2\,\alpha_1^2\,\alpha_2^2 - 2\,U_{3-j}^2\,\alpha_3\,U_j^3\,\alpha_1^2\,\alpha_2^2 \\
& + 2\,U_{3-j}^3\,\alpha_3\sqrt{DTT}\,\alpha_1^3 - U_{3-j}^4\,\alpha_3\,U_j\,\alpha_1^4 - U_{3-j}^3\,\alpha_3\,U_j^2\,\alpha_1^4 + {} \\
& U_{3-j}^2\,\alpha_3\,U_j^3\,\alpha_1^4 - 2\,U_{3-j}^3\,\alpha_3\,U_j^2\,\alpha_2^4 + U_{3-j}^2\,\alpha_3\,U_j^3\,\alpha_2^4 + 3 \\
& U_{3-j}^4\,\alpha_1^4\,\alpha_2^2\,z + 3\,U_{3-j}^2\,\alpha_1^4\,\alpha_2^4\,z^3 + 6\,U_{3-j}\,\alpha_3\sqrt{DTT}\,\alpha_2^3\,z\,\alpha_1^3\,y \\
& + 8\,U_{3-j}\,\alpha_3\,U_j\,\alpha_2^4\,z^2\,\alpha_1^4\,y\Big)\Big/\Big(\Big(z\,\alpha_1^2\,U_{3-j}^3 - 2\,U_{3-j}^2\,\alpha_1^2\,z\,y\,\alpha_2^2 \\
& - 2\,U_{3-j}^2\,\alpha_1^2\,U_j\,z + 4\,U_{3-j}\,\alpha_1^2\,\alpha_2^2\,U_j\,z\,y + U_{3-j}\,\alpha_1^2\,\alpha_2^2\,z^3 \\
& + U_{3-j}\,\alpha_1^2\,U_j^2\,z + 2\,U_{3-j}\,y\sqrt{DTT}\,\alpha_2\,\alpha_1 - U_{3-j}\,U_j^2\,\alpha_2^2\,z \\
& - 2\,U_j\,z^3\,\alpha_2^2\,\alpha_1^2 - 2\,z^2\sqrt{DTT}\,\alpha_2\,\alpha_1\Big)\,\alpha_1^2\,\alpha_2^2\Big)
\end{aligned}
\tag{35}
$$

3.5 Control for Region QT

The last region of the switching surface denoted QT is very similar to the second one denoted TQ. As the name says the combination of parameters q_i and t_i values is opposite to the region TQ. Here the first subsystem $\mathbf{x}_1$ is in the linear case, i.e. its states comply with (9) and the second subsystem $\mathbf{x}_2$ fulfils (10) that means t_2 is nonzero. Therefore we substitute $t_1 = t_2$ and $q_2 = U_{3-j} - q_1$ in (12)

$$
\begin{aligned}
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {} & \Bigg[\Bigg[q_1\left(\frac{1}{\alpha_1^3} - \frac{t_2}{\alpha_1^2} + \frac{1}{2}\,\frac{t_2^2}{\alpha_1}\right) - \frac{1}{6}\,U_j\,t_2^3 + \big(U_{3-j} \\
& - q_1\big)\left(\frac{1}{\alpha_2^3} - \frac{t_2}{\alpha_2^2} + \frac{1}{2}\,\frac{t_2^2}{\alpha_2}\right) - \left(\frac{1}{6}\,U_{3-j} - \frac{1}{6}\,U_j\right)t_2^3\Bigg], \\
& \Bigg[q_1\left(\frac{1}{\alpha_1^2} - \frac{t_2}{\alpha_1}\right) + \frac{1}{2}\,U_j\,t_2^2 + \big(U_{3-j} - q_1\big)\left(\frac{1}{\alpha_2^2} - \frac{t_2}{\alpha_2}\right) \\
& + \left(\frac{1}{2}\,U_{3-j} - \frac{1}{2}\,U_j\right)t_2^2\Bigg], \\
& \Bigg[\frac{q_1}{\alpha_1} - U_j\,t_2 + \frac{U_{3-j} - q_1}{\alpha_2} - \big(U_{3-j} - U_j\big)\,t_2\Bigg]\Bigg]
\end{aligned}
\tag{36}
$$

First we solve parameters q_1 and t_2 from the last two equations of the set (36). From the solution of the quadratic equation we choose that one that gives the positive solution of t_2. After introducing discriminant DQT

$$
DQT = \alpha_2^2\,U_{3-j}^2 + z^2\,\alpha_1^2\,\alpha_2^2 + \alpha_1^2\,U_{3-j}^2 - 2\,y\,\alpha_1^2\,\alpha_2^2\,U_{3-j}
\tag{37}
$$

we get

$$
q_1 = \frac{-\alpha_2\,U_{3-j} + sign\left(U_j\right)\sqrt{DQT}}{\alpha_1 - \alpha_2}
\tag{38}
$$

$$
t_2 = -\frac{-\alpha_1\,U_{3-j} + z\,\alpha_1\,\alpha_2 - \alpha_2\,U_{3-j} + sign\left(U_j\right)\sqrt{DQT}}{\alpha_1\,\alpha_2\,U_{3-j}}
\tag{39}
$$

To express the points of the region QT we substitute (38) and (39) into the first equation of the (36)

$$
\begin{aligned}
x = {} & \frac{1}{6}\,\frac{1}{U_{3-j}^2\,\alpha_1^3\,\alpha_2^3}\Big(3\,\alpha_1^3\,U_{3-j}^2\,z\,\alpha_2 + z^3\,\alpha_1^3\,\alpha_2^3 + 2\,\alpha_1^3\,U_{3-j}^3 \\
& + 3\,z\,\alpha_1\,\alpha_2^3\,U_{3-j}^2 - 3\,sign\left(U_j\right)^2 DQT\,z\,\alpha_1\,\alpha_2 + 2\,\alpha_2^3\,U_{3-j}^3 \\
& - 2\,sign\left(U_j\right)^3 DQT^{3/2}\Big)
\end{aligned}
\tag{39}
$$

The control value results from (18)

$$
\begin{aligned}
u = {} & \frac{1}{3}\Big(-3\,y\,\alpha_1^3\,\alpha_2^3\,U_{3-j}^2 + 3\,\alpha_1^3\,\alpha_2^3\,U_{3-j}\,z^2 + 3\,\alpha_1^2 \\
& \alpha_2^2\,U_{3-j}\,z\,sign\left(U_j\right)\sqrt{DQT} + 3\,\alpha_3\,x\,\alpha_1^3\,\alpha_2^3\,U_{3-j}^2 + \alpha_3\,z^3\,\alpha_1^3\,\alpha_2^3 \\
& - 3\,\alpha_3\,\alpha_1^3\,z\,\alpha_2^3\,y\,U_{3-j} - \alpha_3\,\alpha_1^3\,U_{3-j}^3 - \alpha_3\,\alpha_2^3\,U_{3-j}^3 \\
& + \alpha_3 sign\left(U_j\right) DQT^{3/2}\Big)\Big/\Big(\alpha_1^2\,\alpha_2^2\,\big(z^2\,\alpha_1\,\alpha_2 - \alpha_1\,\alpha_2\,y\,U_{3-j} \\
& + sign\left(U_j\right)\sqrt{DQT}\,z\big)\Big)
\end{aligned}
\tag{40}
$$

3.6 Control Algorithm

The control algorithm consists in the localization of the initial state to one of the above mentioned regions and then of the control value calculation. But before we have to specify the parameter j. Then we can calculate the parameters q_i. According their values we can find the region to which the initial point belongs and finally evaluate the control value.

START

1. Evaluate q_1 according (20) and q_2 according (21)
2. IF q_1 fulfils (9) AND q_2 fulfils (9) THEN calculate u according (23) – *Region QQ*

3. IF q_1 fulfils (9) AND q_2 NOT fulfils (9) THEN calculate $j = \frac{3 - sign(q_2)}{2}$ AND GOTO 8
4. Calculate $j = \frac{3 + sign(q_1)}{2}$, $q_1 = U_j$ and q_2 according (26)
5. IF q_2 fulfils (9) THEN calculate u according (29) – *Region TQ*
6. IF $q_2 U_j < 0$ THEN calculate u according (35) – *1st part of the region TT*
7. Calculate $j = \frac{3 - sign(q_1)}{2}$
8. Evaluate q_1 according (38)
9. IF q_1 fulfils (9) THEN calculate u according (40) - *Region QT*
10. Evaluate u according (35) – *2nd part of the region TT*

END

It is important to notice that at the end the control value computed according this algorithm must be limited by (2).

4. EVALUATION OF DESIGNED CONTROLLER

To show the performance of the designed controller we have carried out several simulations that differ from the starting point, parameters of the controller, and constraints. In the Fig. 3 one can see the time responses from the initial state $\mathbf{x} = [200 \quad 0 \quad 0]^t$ under nonsymmetrical control value constraints $u = \langle -1 \quad 2 \rangle$.

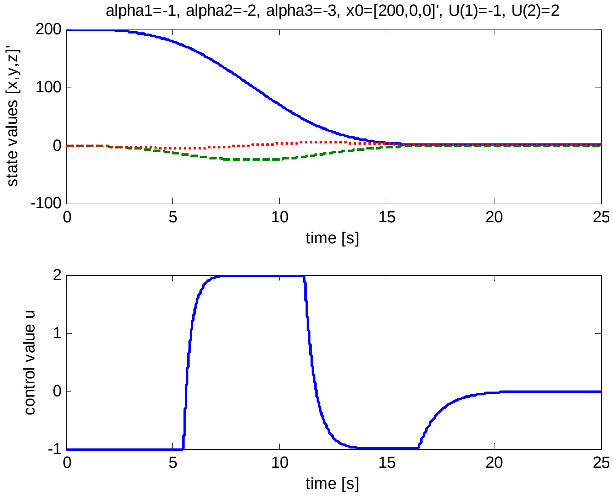


Fig. 3. Time responses of state and control variables. Sub-optimal controller with nonsymmetrical constraints.

All three pulses of the time optimal control can be mentioned but the control value switches from one limit value to the other one smoothly. The change rate is given by the choice of parameters α_i. In this case the values of α_i were $\alpha_1 = -1$, $\alpha_2 = -2$, $\alpha_3 = -3$.

The comparison of the time sub-optimal control with the optimal one is shown in the Fig. 4. This time the starting point was $\mathbf{x} = [15.7916 \quad -4.25 \quad 1]^t$. The other parameters were $\alpha_1 = -1.5$, $\alpha_2 = -3$, $\alpha_3 = -6$ and $u = \langle -1 \quad 1 \rangle$.

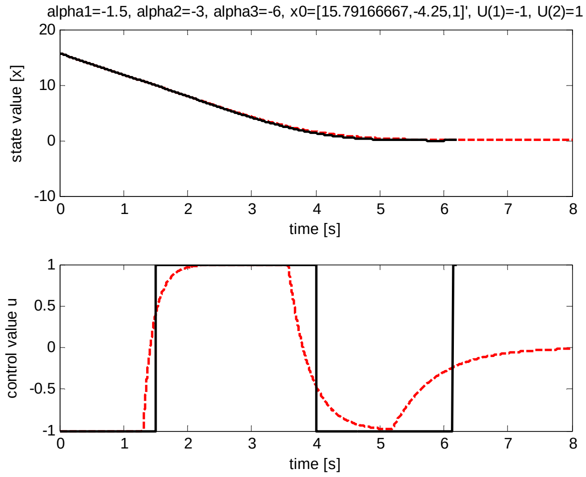


Fig. 4. Comparison of time optimal and sub-optimal control.

There are not big differences in the time responses of the state variable x. But one can see the difference in the behaviour of the control variables. The time sub-optimal control variable uses limits for a shorter period because it needs a certain time to switch to the opposite value. The time optimal control variable switches immediately that can cause problems when the dynamics of a controlled system is not precisely identified. The time sub-optimal controller switches in advance and it finishes later but it is not so sensitive to the uncertain parameters or unmodelled dynamics. By moving the negative values of parameters α_i towards a zero we could get behaviour similar to the linear pole assignment controller. To get the exact linear behaviour we have to higher the control value limits.

Fig. 5 shows the comparison of the time sub-optimal control with the model predictive control (MPC). The example for comparison is taken from (Glattfelder and Schaufelberger, 2003). The parameters of MPC controller were following: prediction horizon $N = 50$, sampling period $T_s = 0.050$, weights of linear quadratic controller $Q = \begin{bmatrix} 11.78 & 0 & 0 \\ 0 & 3.345 & 0 \\ 0 & 0 & 0.3175 \end{bmatrix}$, corresponding bandwidth $\Omega = 3.25$ for $r = 0.01$. Higher Ω-values result in overshooting responses. Parameters of the time sub-optimal controller have been chosen in order to get similar response of control action. They are $\alpha_1 = -3$, $\alpha_2 = -20$, $\alpha_3 = -100$. There are time responses from the initial state $\mathbf{x} = [-1 \quad 0 \quad 0]^t$ under control value constraints $u = \langle -1 \quad 1 \rangle$. One can see that the output response (state value x) is for both controllers almost identical. Very small differences can be shown in the time responses of the control action. This

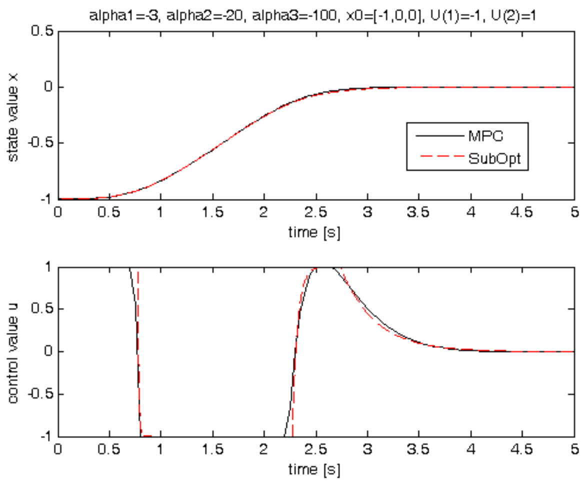


Fig. 5. Comparison of sub-optimal and model predictive control with similar behaviours.

example shows that it is possible to set the parameters of the sub-optimal controller to get equal results with the MPC controller.

Two different responses of the MPC and sub-optimal controllers are shown in the Fig. 6. This time the parameters of both controllers have been decreased. The MPC controller parameters were: prediction horizon $N = 50$, sampling period $T_s = 0.050$, weights of linear quadratic controller $Q = \begin{bmatrix} 2.4413 & 0 & 0 \\ 0 & 1.175 & 0 \\ 0 & 0 & 0.1875 \end{bmatrix}$, corresponding bandwidth $\Omega = 2.5$ for $r = 0.01$. The decrease of parameter values caused that the control action did not reach the control constraint in the last (third) interval of control. One can mention that the switch between the limit values in the first and the second interval of control is almost time optimal. The end of the response (third interval of control) corresponds to the linear controller behaviour.

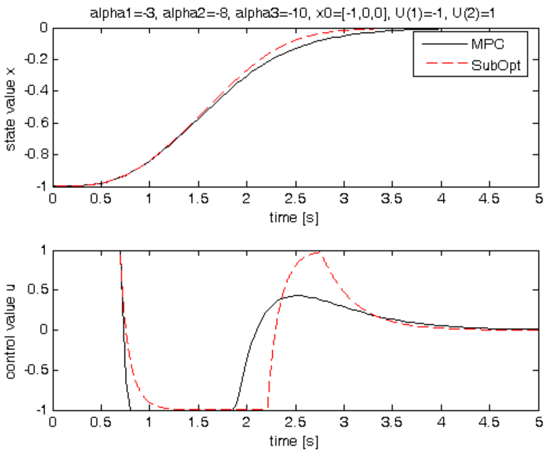


Fig. 6. Comparison of sub-optimal and model predictive control with different behaviours.

Decreasing Ω to the value 1.575 would cause the second interval of control does not reach the limit value and would have linear behaviour too. Thus decreasing the parameter Ω assures the linear character of the control action in the last intervals of control. Choosing the Ω parameter it is not possible to influence the behaviour of switches between the limit control values separately. Opposite to this the sub-optimal controller has three parameters $\alpha_1, \alpha_2, \alpha_3$ that directly influence the exponential behaviour of switches. So it offers to design the control action response more specifically. In this example the values of α_i were $\alpha_1 = -3$, $\alpha_2 = -8$, $\alpha_3 = -10$.

5. CONCLUSIONS

Presented controller design relies on switching surfaces. Because our aim was to decrease the sensitivity of controller we introduced new parameters into the design. In linear cases these parameters are identical with the poles of the closed loop. Of course, the switching surfaces are more complicated by additional parameters. In this paper we derived the solution with explicit mathematical formulas that is fast enough to be used in real time applications. The designed time sub-optimal controller has been compared with pure time optimal controller and model predictive controller.

ACKNOWLEDGMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09 and by the Grant VEGA No. 1/0656/09.

REFERENCES

Athans, M. and Falb, P. (1996). *Optimal Control: An Introduction to the Theory and its Applications*. McGraw-Hill, New York.

Bisták, P., Ťapák, P., and Huba, M.(2005). Constrained Pole Assignment Control of Double and Triple Integrator. In: CASYS 05, Seveth International Conference on Computing Anticipatory Systems, HEC-Ulg, LIEGE, Belgium.

Huba, M. (2006). "Constrained pole assignment control," In: *Current Trends in Nonlinear Systems and Control*, Boston: Birkhäuser, 163-183.

Pao, L. Y. and Franklin, G. F. (1993). Proximate Time-Optimal Control of Third-Order Servomechanisms, *IEEE Transactions on Automatic Control*, Vol. 38, No. 4.

Pavlov, A.A. (1966). *Synthesis of relay time-optimal systems*. In Russian, Publishing House "Nauka", Moskva.

Ťapák, P., Bisták, P., and Huba, M. (2006). Control for Triple Integrator with Constrained Input. In: 14th Mediterranean Conference on Control and Automation, Piscataway : IEEE, Ancona, Italy.

Walther, U., Georgiou, T. T., and Tannenbaum, A. (2001). On the Computation of Switching Surfaces in Optimal Control: A Gröbner Basis Approach, *IEEE Transactions on Automatic Control*, Vol. 46, No. 4.

Glattfelder, A.H., Schaufelberger, W. (2003). Control Systems with Input and Output Constraints, Springer-Verlag.

Robust Decentralized Controller Design for Performance

Alena Kozáková. Vojtech Veselý.
Jakub Osuský

Institute of Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava,
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
(e-mail: alena.kozakova@stuba.sk)

Abstract: The paper presents an innovation of the robust decentralized controller design for multivariable uncertain systems within the setting of the Equivalent Subsystems Method (ESM). The aim of the proposed design procedure is to guarantee robust stability and plant-wide nominal performance in terms of maximum overshoot achieved through phase margins specified for equivalent subsystems. The developed design procedure is illustrated by an example.

Keywords: Decentralized controller, Frequency domain control, Nominal performance, Robust stability

1. INTRODUCTION

When designing decentralized control (DC), performance objectives can be of two basic types: a) achieving required performance in different subsystems; or b) achieving plant-wide desired performance. The Nyquist-based frequency domain decentralized controller design technique for performance called "Equivalent Subsystems Method" (ESM) (Kozáková et al., 2009a, b) belongs to the latter group. According to it, the DC design for plants described by transfer function matrices is performed through independent designs for equivalent subsystems that are actually Nyquist plots of decoupled subsystems shaped by a selected characteristic locus of the interactions matrix. It has been proved that local controllers independently tuned for stability and specified feasible performance in terms of degree of stability in equivalent subsystems provide a decentralized controller guaranteeing the very degree of stability of the full system. In (Kozáková et al., 2010), the ESM design technique has been used to design digital decentralized controller for specified phase margin thus guaranteeing plant-wide maximum overshoot by applying discrete Bode plots of equivalent subsystems,.

Application of the ESM in the design for robust stability and nominal performance can be found in (Kozáková and Veselý, 2007; 2008; 2009), (Kozáková et al., 2009a) always in a two-stage design methodology: first, the DC for nominal performance is designed according to ESM, and afterwards, fulfillment of the robust stability conditions is examined; if robust stability is not achieved either controller parameters are to be modified, or the redesign is to be carried out with modified performance requirements.

This paper presents a robust DC design methodology based on direct integrating of robust stability conditions in the ESM. In this way, local controllers of equivalent subsystems are designed with regard to robust stability, and nominal performance in terms of maximum peak of the complementary sensitivity (or sensitivity, depending on uncertainty type) that provides information about the maximum overshoot and is transformable into lower bound for the phase margin of equivalent subsystems.

The paper is organized as follows: Preliminaries and problem formulation are in Section 2, principles of the Equivalent Subsystems Method (ESM) are revisited in Section 3. Section 4 presents the direct robust DC design procedure in the ESM setting. Theoretical results are demonstrated on an example in Section 5.

2. PRELIMINARIES AND PROBLEM FORMULATION

Consider a MIMO system described by a transfer function matrix $G(s) \in R^{m\times m}$, and a controller $R(s) \in R^{m\times m}$ in the standard feedback loop (Fig. 1); Necessary and sufficient closed-loop stability conditions are stipulated by the Generalized Nyquist Stability Theorem applied to the closed-loop characteristic polynomial (CLCP)

$$det\, F(s) = det[\, I + Q(s)\,] \qquad (1)$$

where $Q(s) = G(s)R(s) \in R^{m\times m}$.

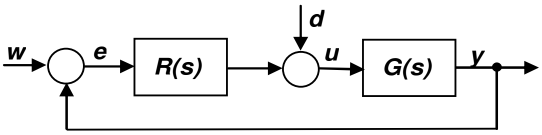


Fig. 1. Standard feedback configuration

In the sequel, D denotes the standard Nyquist D-contour in the complex plane; Nyquist plot of $g(s)$ is the image of the Nyquist contour under $g(s)$; $N[k, g(s)]$ is the number of anticlockwise encirclements of $(k, j0)$ by the Nyquist plot of $g(s)$. Characteristic functions of $Q(s)$ are the set of m algebraic functions $q_i(s),\ i = 1,...,m$ given as

$$det[\,q_i(s)I_m - Q(s)\,] = 0 \qquad i = 1,...,m \tag{2}$$

Characteristic loci (CL) are the set of loci in the complex plane traced out by the characteristic functions of $Q(s), \forall s \in D$. The CLCP (1) expressed in terms of characteristic functions of $Q(s)$ reads as follows

$$det\,F(s) = det[\,I + Q(s)\,] = \prod_{i=1}^{m}[\,1 + q_i(s)\,] \tag{3}$$

Theorem 1. (Generalized Nyquist Stability Theorem)
The closed-loop system in Fig. 1 is stable if and only if

a. $det\,F(s) \neq 0 \qquad \forall s \in D$

b. $N[\,0, det\,F(s)\,] = \sum_{i=1}^{m} N\{0,[\,1 + q_i(s)\,]\} = n_q$ (4)

where $F(s) = (I + Q(s))$ and n_q is the number of unstable poles of $Q(s)$.
Let the uncertain plant be specified as a set Π of N transfer function matrices

$$\Pi = \{\,G^k(s)\,\},\ k = 1,2,...,N \quad \text{where} \quad G^k(s) = \left\{G_{ij}^k(s)\right\}_{m\times m} \tag{5}$$

The set of unstructured perturbations D_U is defined as follows

$$D_U := \{\,E(j\omega) : \sigma_{max}[\,E(j\omega)\,] \leq \ell(\omega),\\ \ell(\omega) = \max_k \sigma_{max}[\,E(j\omega)\,]\} \tag{6}$$

where $\ell(\omega)$ is a scalar weight function on the norm-bounded perturbation $\Delta(s) \in R^{m\times m}$, $\sigma_{max}[\,\Delta(j\omega)\,] \leq 1$ over the given frequency range, $\sigma_{max}(\cdot)$ is the maximum singular value of (.); hence $E(j\omega) = \ell(\omega)\Delta(j\omega)$.
For unstructured uncertainty, the set Π can be generated by either additive (E_a), multiplicative input (E_i) or output (E_o) uncertainties, or their inverse counterparts (E_{ia} E_{ii} E_{io}) used for uncertainty associated with plant poles located in the closed right half-plane. Only the additive and inverse additive uncertainties will be addressed in detail; relations for other uncertainty forms can be derived by analogy.
Denote $G(s)$ any member of Π, $G_0(s)$ the nominal model, and $\ell_j(\omega)$ the scalar weight on a normalized perturbation. Individual uncertainty forms generate the related sets Π_j for *j=a, ia.*

Additive uncertainty:

$$\Pi_a := \{\,G(s) : G(s) = G_0(s) + E_a(s),\\ E_a(j\omega) \leq \ell_a(\omega)\Delta(j\omega)\}\\ \ell_a(\omega) = \max_k \sigma_{max}[\,G^k(j\omega) - G_0(j\omega)\,],\ k = 1,2,\dots,N \tag{7}$$

Inverse additive uncertainty

$$\Pi_{ia} := \{\,G(s) : G(s) = G_0(s)[\,I - E_{ia}(s)G_0(j\omega)\,]^{-1},\\ E_{ia}(j\omega) \leq \ell_{ia}(\omega)\Delta(j\omega)\}\\ \ell_{ia}(\omega) = \max_k \sigma_{max}\{[\,G_0(j\omega)\,]^{-1} - [\,G^k(j\omega)\,]^{-1}\,\},\\ k = 1,2,\dots,N \tag{8}$$

The standard feedback loop with uncertain plant can be recast into the $M - \Delta$ structure (Fig. 2) where $\Delta(s) \in R^{m\times m}$ is the norm-bounded complex perturbation. For the uncertainty forms (7), (8) the corresponding $M(s)$ are respectively

$$M(s) = \ell_a(s)R(s)[\,I + G_0(s)R(s)\,]^{-1} \tag{9}$$

$$M(s) = \ell_{ia}(s)[\,I + G_0(s)R(s)\,]^{-1}G_0(s) \tag{10}$$

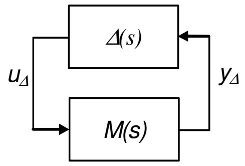


Fig. 2. $M - \Delta$ structure

According to (Skogestad and Postlethwaite, 2005) if $M(s)$ is stable (nominal stability) and the perturbation $\Delta(s)$ is stable, then the $M - \Delta$ system is stable for all $\Delta(s) : \sigma_{max}(\Delta) \leq 1$ if and only if

$$\sigma_{max}[\,M(j\omega)\,] < 1\,, \ \forall\omega \tag{11}$$

Conservatism of the robust stability condition (19) can be relaxed by "structuring" the additive uncertainty to yield the additive affine-type uncertainty $E_{af}(s)$ (Kozáková and Veselý, 2007; 2008)

$$E_{af}(s) = \sum_{i=1}^{p} G_i(s)q_i \tag{12}$$

where $G_i(s) \in R^{m\times m}$, *i=0,1, …, p* are stable matrices, *p* is the number of uncertainties defining 2^p polytope vertices that correspond to individual perturbed models; q_i are polytope parameters. The related Π_{af} is

$$\Pi_{af} := \{\,G(s) : G(s) = G_0(s) + E_{af},\\ E_{af} = \sum_{i=1}^{p} G_i(s)q_i,\\ q_i \in < q_{i\,min}, q_{i\,max} >,\ q_{i\,min} + q_{i\,max} = 0\,\} \tag{13}$$

where $G_0(s)$ is the „affine“ nominal model. In the matrix form, individual plants from the set Π_{af} can be expressed as follows

$$G(s) = G_0(s) + QG_u(s) \tag{14}$$

where $Q = [\,I_{q_1} \dots I_{q_p}\,]^T \in R^{m\times(m\times p)}$, $I_{q_i} = q_i I_{m\times m}$,

$G_u(s) = [\,G_1 \dots G_p\,]^T \in R^{(m\times p)\times m}$.

Similarly to previous uncertainty forms, the feedback loop with uncertain plant modeled using the additive affine type uncertainty in Fig. 3, can be recast into the $M_{af} - Q$ structure with $M_{af} = G_u R(I + G_0 R)^{-1} = G_u(I + RG_0)^{-1}R$.
Stability condition for the $M_{af} - Q$ system is

$$\sigma_{max}(M_{af}Q) < 1\,, \tag{15}$$

under the assumption $q_0 = |q_{i\,min}| = |q_{i\,max}|$, (14) can further be modified to yield

$$\sigma_{max}(M_{af})q_0\sqrt{p} < 1 \tag{16}$$

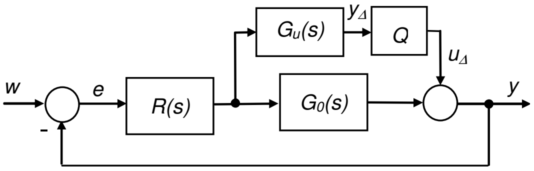


Fig. 3. Standard feedback loop with additive affine-type additive uncertainty

2.1 Problem Formulation

Consider an uncertain system with m subsystems given as a set of N transfer function matrices obtained as a result of identification in N working points of the plant operation range. Assume that the uncertain system be described by a nominal model $G_0(s)$ and any unstructured uncertainty form (7), (8) or (12) where $G_0(s)$ can be split as follows

$$G_0(s) = G_d(s) + G_m(s) \tag{17}$$

where $\forall s \in D$

$$G_d(s) = diag\{ G_i(s)\}_{m\times m},\ det\ G_d(s) \neq 0$$
$$G_m(s) = G_0(s) - G_d(s)$$

A decentralized controller (DC)

$$R(s) = diag\{ R_i(s)\}_{m\times m},\ det\, R(s) \neq 0 \tag{18}$$

is to be designed to guarantee stability over the whole operating range of the plant specified by either (7), (8) or (13) (robust stability) and a specified performance of the nominal model (nominal performance), Fig. 4.
To solve the this problem, a frequency domain robust decentralized controller design technique has been developed (Kozáková and Veselý, 2009; Kozáková et. al., 2009b); the core of it is the Equivalent Subsystems Method (ESM).

3. EQUIVALENT SUBSYSTEMS METHOD

The Equivalent Subsystems Method (ESM) is a Nyquist-based DC design method for stability and guaranteed performance of the full system. According to it, independent local controller designs are carried out for the so-called equivalent subsystems that are actually Nyquist plots of decoupled subsystems, shaped by one selected characteristic locus of the interactions matrix. If local controllers of equivalent subsystems are independently tuned for stability and specified feasible performance, the resulting decentralized controller guarantees for the full system the same performance as specified for equivalent subsystems. ESM used in the design for robustness (Kozáková et al., 2009b) allows to consider the *full* nominal model, thus reducing conservatism of resulting robust stability conditions.

The key idea behind the method is factorization of the CLCP (1) in terms of the split nominal system (17) under the decentralized controller (18)

$$det\, F(s) = det[\, R^{-1}(s) + G_d(s) + G_m(s)]\, det\, R(s) \tag{19}$$

Denote the sum of diagonal matrices in the first bracketed term as follows

$$R^{-1}(s) + G_d(s) = P(s) \tag{20}$$

where $P(s) = diag\{ p_i(s)\}_{m\times m}$.

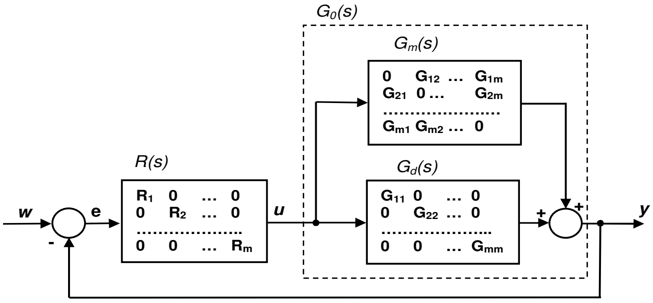


Fig. 4. Standard feedback loop under decentralized controller

If choosing the diagonal matrix $P(s) = diag\{ p_k(s)\}_{m\times m}$ with identical entries so as to „counterbalance" interactions $G_m(s)$ then, according to (2), the characteristic equation corresponding to the first r.h.s. term in (19) defines the *k-th* of the m characteristic functions of $[-G_m(s)]$ denoted $g_i(s), i = 1,2,...,m$; thus

$$\begin{aligned} det[\, P(s) + G_m(s)] &= det[\, p_k I + G_m\,] = \\ &= \prod_{i=1}^{m} [-g_k(s) + g_i(s)] = 0,\ \ k = 1,2,...m \end{aligned} \tag{21}$$

According to the Cayley-Hamilton theorem from the viewpoint of stability, the interactions matrix $G_m(s)$ can be replaced by *[-P(s)]* yielding the important relationship

$$\begin{aligned} det[\, I + G(s)R(s)] &= det\{\, I + [\, G_d(s) + G_m(s)]R(s)\} = \\ &= det[\, R^{-1}(s) + G_d(s) - P(s)]\, det\, R(s) = \\ &= det[\, I + G^{eq}(s)R(s)] \end{aligned} \tag{22}$$

where

$$G^{eq}(s) = diag\{ G_i^{eq}(s)\}_{m\times m} \tag{23}$$

is a diagonal matrix of m equivalent subsystems

$$G_i^{eq}(s) = G_i(s) + g_k(s),\qquad i = 1,2,\ldots,m\,; \tag{24}$$

As all matrices are diagonal, on subsystems level (22) breaks down into m equivalent characteristic polynomials

$$CLCP_i^{eq}(s) = 1 + R_i(s)G_i^{eq}(s)\qquad i = 1,2,...,m \tag{25}$$

Considering (21)-(25), stability conditions of Theorem 1 modify as follows:

Corollary 1.
The closed-loop in Fig. 4 comprising the system (17) and the decentralized controller (18) is stable if and only if there exists a diagonal matrix $P(s) = diag\{ p_i(s)\}_{m\times m}$ such that

1. $det[\, p_k(s)I + G_m\,] = 0,$ for fixed $k \in \{1,...,m\}$;

2. all equivalent characteristic polynomials (25) have roots with $Re\{ s \} < 0$;

3. $N[0, det\, F(s)] = n_q$

where $det\, F(s) = I + G(s)R(s)$ and n_q is number of open loop poles with $Re\{ s \} > 0$.

The design technique resulting from *Corollary 1* enables to design the decentralized controller through designing local controllers for independent equivalent subsystems using any SISO frequency-domain design method, e.g. the Neymark D-partition method (Kozáková et al. 2009b), standard Bode diagram design (Bucz et al., 2010) etc.

In the originally developed ESM version (Kozáková et. al. 2009a; 2009b) it was proved that local controllers independently tuned for stability and a specified feasible degree of stability of equivalent subsystems constitute the decentralized controller guaranteeing the same degree of stability plant-wide. In (Kozáková et al. 2010) the performance specification applied in ESM was based on the relationship between phase margins of equivalent subsystems and the maximum overshoot. This performance specification is further developed towards robust stability.

4. ROBUST DECENTRALIZED CONTROLLER DESIGN

This section deals with implementation of the ESM in the decentralized controller design for robust stability and nominal performance applicable for uncertain systems described as a set of transfer function matrices. The nominal model can be calculated either as the mean value parameter model (Skogestad and Postlethwaite, 2005), or the "affine" model, obtained within the procedure for calculating the affine-type additive uncertainty (Kozáková and Veselý, 2007; 2008). Unlike the standard robust approach to DC design in which the diagonal model as the nominal one (interactions are included in the uncertainty), the ESM method applied in the design for nominal performance allows to consider the *full* nominal model. Model uncertainty is described by any unstructured uncertainty form (7), (8) or (13).

In (Kozáková and Veselý, 2008; 2009; Kozáková et al. 2009a) a two-stage robust DC design methodology was proposed based on ESM and fulfillment of the *M-Δ* structure stability conditions. The direct DC design for robust stability and nominal performance is the main result of this paper.

4.1 Direct decentralized controller design for robust stability and nominal performance

If the robust stability conditions (11) or (16) are directly integrated in the ESM, local controllers of equivalent subsystems are designed already with regard to robust stability. A suitable performance specification is the maximum peak of the complementary sensitivity M_T related to maximum overshoot in the full system; in equivalent subsystems it can be translated into lower bounds of phase margins according to (26) (Skogestad and Postlethwaite, 2005)

$$PM \geq 2\, arcsin\left(\frac{1}{2M_T} \right) \geq \frac{1}{M_T} [\, rad\,] \qquad (26)$$

where PM is the phase margin, and M_T is the maximum peak of the complementary sensitivity *T(s)*

$$T(s) = G(s)R(s)[I + G(s)R(s)]^{-1} \qquad (27)$$

As for MIMO systems

$$M_T = \sigma_{max}(T), \qquad (28)$$

the upper bound for the nominal complementary sensitivity $T_0(s) = G_0(s)R(s)[I + G_0(s)R(s)]^{-1}$ can be derived by substituting into (1) the uncertain system model (additive uncertainty is considered in the following development) where $G_0(s)$ is the nominal model:

$$\begin{aligned} &det[I + (G_0 + \ell_a \Delta)R] = \\ &\quad = det(I + G_0 R)\, det[I + \ell_a \Delta R(I + G_0 R)^{-1}] \end{aligned} \qquad (29)$$

where the first term on the r.h.s. of is the CLCP of the nominal system that corresponds to the CLCP^{eq} according to (22); condition for stability of the second term is determined using the small gain theorem. Hence the uncertain system is stable if and only if the nominal closed loop is stable and

$$\left\| \ell_a \Delta R(I + G_o R)^{-1} \right\| < 1 \qquad (30)$$

Considering the spectral norm and the singular value properties, (30) can readily be manipulated to yield the final condition (33). Bounds for other uncertainty forms can be derived by analogy.

In case of inverse uncertainty forms, robustness bounds are obtained in terms of the maximum peak of the sensitivity $M_S = \sigma_{max}(S)$ where

$$S(s) = [I + G(s)R(s)]^{-1} \qquad (31)$$

and using the lower bounds for PM in the form (Skogestad and Postlethwaite, 2005)

$$PM \geq 2\, arcsin\left(\frac{1}{2M_S} \right) \geq \frac{1}{M_S} [\, rad\,] \qquad (32)$$

Upper bounds for $\sigma_{max}[T_0(j\omega)]$ or $\sigma_{max}[S_0(j\omega)]$ for additive-type uncertainties are summarized below.

Additive uncertainty:

$$\sigma_{max}[T_0(j\omega)] < \frac{\sigma_{min}[G_0(j\omega)]}{|\ell_a(\omega)|} = L_A(\omega) \quad \forall \omega \qquad (33)$$

Additive affine-type uncertainty:

$$\sigma_{max}[T_0(j\omega)] < \frac{1}{q_0\sqrt{p}} \frac{\sigma_{min}[G_0(j\omega)]}{\sigma_{max}[G_u(j\omega)]} = L_{AF}(\omega) \quad \forall \omega \qquad (34)$$

Inverse additive uncertainty:

$$\sigma_{max}[S_0(j\omega)] < \frac{1}{|\ell_{ia}(\omega)| \sigma_{max}[G_0(j\omega)]} = L_{IAF}(\omega), \ \forall \omega \qquad (35)$$

Any of the derived bounds (33), (34) or (35) for the nominal model can directly be implemented in the ESM due to the fact that performance achieved in equivalent subsystems is simultaneously guaranteed for the full system. The main benefit of this approach is the possibility to find the maximum overshoot of the full system in terms of $\sigma_{max}(T_0)$ or $\sigma_{max}(S_0)$ for which robust stability is guaranteed, translate it into corresponding minimum phase margins required in equivalent subsystems and finally design local controllers for individual single input – single output equivalent subsystems independently. In this case the recommended design method for the ESM setting is the Bode diagram design.

Considering performance just in terms of M_T or M_S is not sufficient, the speed of response has to be considered as well which leads to considering the bandwidth frequency of the closed-loop system as well. In general, a large bandwidth corresponds to a smaller rise time, since high frequency signals are more easily passed on to the outputs. If the bandwidth is small, the time response will generally be slow and the system will usually be more robust. The gain crossover frequency ω_0 is frequently used to define closed-loop bandwidth.

The Bode plot design procedure with regard both to the required phase margin and the required bandwidth is demonstrated in the next section on a simple example of SISO robust PI(D) controller design with guaranteed overshoot and settling time of transients.

5. EXAMPLE

Consider a SISO plant given by 3 transfer functions corresponding to its three different working points:

$$G_1(s)=\frac{0.5s+1.2}{50s^2+15s+1} \qquad G_2(s)=\frac{0.5s+1.3}{45s^2+15s+1.3}$$

$$G_3(s)=\frac{0.5s+0.7}{53s^2+15s+0.8}$$

Next calculations include the nominal model (as a mean value parameter one), the additive uncertainty $\ell_a(\omega)$ according to (7) and the upper bound for the nominal complementary sensitivity $L_A(\omega)$ according to (33).

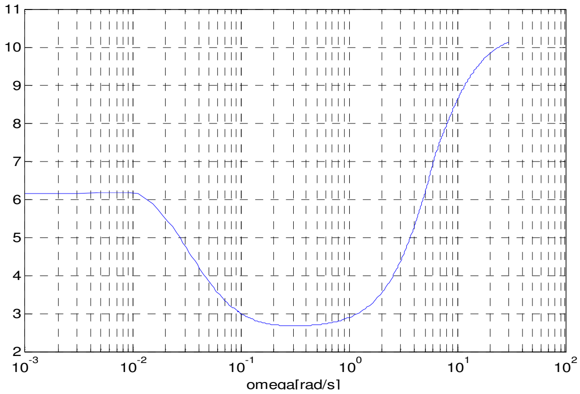


Fig. 5. Robustness bound $L_A(\omega)$ for nominal complementary sensitivity

The least value $min_\omega L_A(\omega)=M_T$ is chosen to generate the minimum required phase margin guaranteeing robust stability; in our case $M_T=2.66$ corresponds to $PM_{min}=21.6^0$.

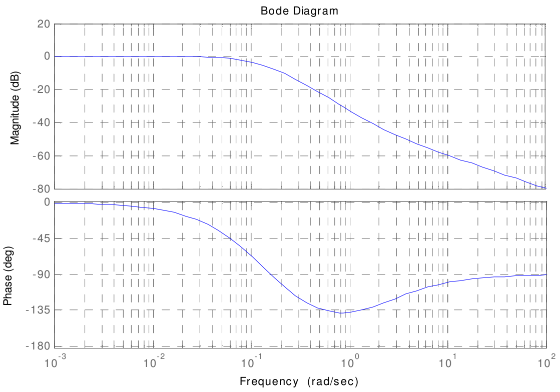


Fig. 6. Bode plot of the nominal system $G_0(s)$

The PI(D) controller design is carried out with regard to both the required phase margin and the required bandwidth, the latter being related to the settling time according to the relation

$$\frac{\pi}{t_s}<\omega_0<\frac{4\pi}{t_s} \qquad (36)$$

The design philosophy is as follows:

After specifying the required PM_{req} and settling time t_S, ω_0 is calculated from (36) and the $PM(\omega_0)$ is read off. If $PM(\omega_0)>PM_{req}$, a PI controller is designed.

If $PM(\omega_0)<PM_{req}$, a PD controller $G_{PD}(s)=1+K_Ds$ is to be designed first to provide $PM_{req}(\omega_0)$ and then a PI controller $G_{PI}(s)=K_P+\frac{K_I}{s}$ is designed. The resulting PID controller is a combination of both $G_{PID}(s)=(K_P+\frac{K_I}{s})(1+K_Ds)$.

Consider the required $t_S=60s$ which corresponds to $\omega_0=0.1309s^{-1}$. From the Bode plot of the uncompensated system in Fig. 6 and $PM_{req}(\omega_0)=49^0$ it is obvious that a PI controller will be sufficient, its resulting parameters are $G_{PI}=0.4602+\frac{0.053}{s}$. Bode plot of the compensated system in Fig. 7 proves achieving of the required parameters. Closed-loop step responses of the nominal model and models in individual working points are in Fig. 8 and Fig. 9, respectively. Stability robustness is verified in Fig. 10.

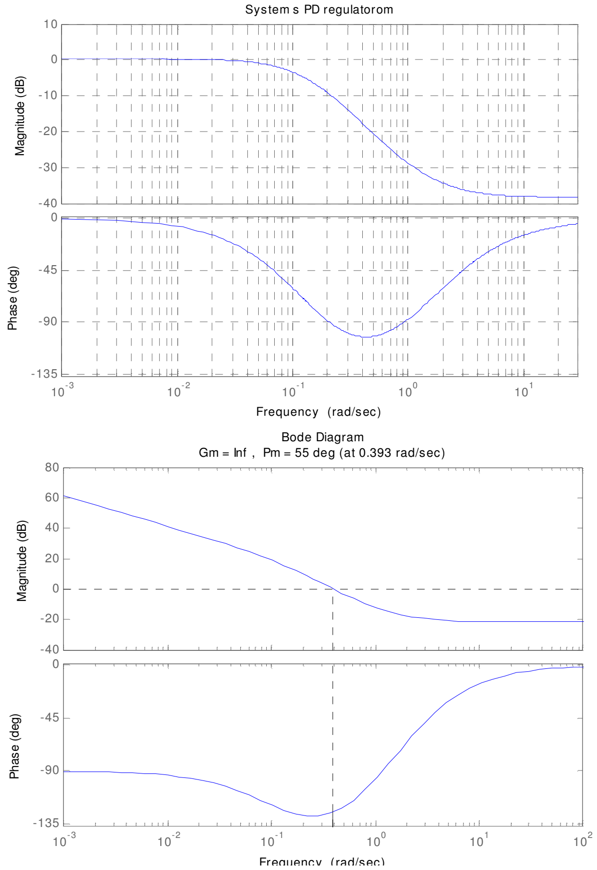


Fig. 7 Bode plot of the compensated system (PI controller)

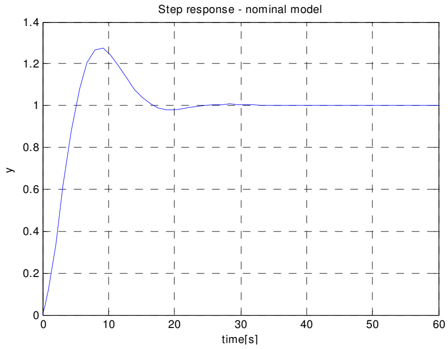


Fig. 8 Closed-loop step response with the nominal model

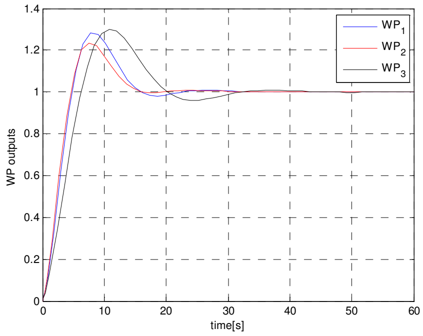


Fig. 9 Closed-loop step responses in individual working points

CONCLUSIONS

The paper deals with the decentralized PID controller design for robust stability and plant-wide nominal performance within the setting of the Equivalent Subsystems Method (ESM). The nominal performance for the full system specified in terms of maximum overshoot is achieved through phase margins specified for equivalent subsystems. The design methodology per se uses the Bode plots and is therefore applicable also for SISO systems. The design procedure is illustrated by an example.

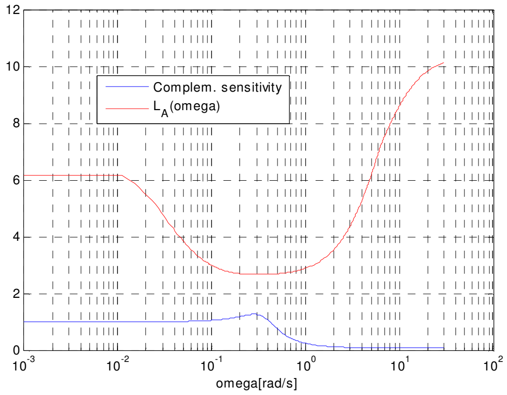


Fig. 10 Robust stability verification

ACKNOWLEDGMENT

The work has been supported by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic and of Slovak Academy of Sciences, grants No. 1/0544/09 and 1/1105/11. This support is very gratefully acknowledged.

REFERENCES

Bucz, Š, Marič, L, Harsányi, L. et al. (2010): A simple robust PID controller design method based on sine-wave identification of the uncertain plant. *Journal of Electrical Engineering*, Vol. 61, No.3, 164-170.

Kozáková, A., and Veselý, V. (2007). Robust decentralized controller design for systems with additive affine-type uncertainty. *Int. J. of Innovative Computing, Information and Control (IJICIC)*, Vol. 3 (5), 1109-1120.

Kozáková, A., and Veselý, V. (2008). Robust MIMO PID controller design using additive affine-type uncertainty. *Journal of Electrical Engineering*, Vol. 59 (5), 241-247.

Kozáková, A., and Veselý, V. (2009). Design of robust decentralized controllers using the M-Δ structure robust stability conditions. *Int. Journal of Systems Science*, Vol. 40 (5), 497-505.

Kozáková, A., Veselý, V., and Osuský, J. (2009a). A new Nyquist-based technique for tuning robust decentralized controllers, *Kybernetika*, Vol. 45 (1), 63-83.

Kozáková, A., Veselý, V., and Osuský, J. (2009b). Decentralized Controllers Design for Performance: Equivalent Subsystems Method. *European Control Conference, ECC'09*, 2295-2300, EUCA Budapest, Hungary.

Kozáková, A., Veselý, V., and Osuský, J. (2010). Decentralized Digital PID Controller Design for Performance. *12th IFAC Symposium on Large Scale*

Systems: Theory and Applications, Ecole Centrale de Lille, Villeneuve d'Ascq, France.

Skogestad, S., and Postlethwaite, I. (2005). *Multivariable fedback control: analysis and design*, John Wiley & Sons Ltd., The Atrium, Southern Gate. Chichester, West Sussex, UK.

Solution of a Robust Stabilization Problem Using YALMIP and Robust Control Toolboxes

M. Bakošová * **J. Oravec** * **M. Kačur** *

* *Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics, Radlinského 9, 812 37 Bratislava, Slovakia*
Tel: +421 2 59325364; e-mail:
{*monika.bakosova,juraj.oravec,michal.kacur*}*@stuba.sk*

Abstract: The aim of this paper is to compare two toolboxes used for solving the robust stabilization problem. Robust static output feedback controller was designed for a continuous stirred tank reactor (CSTR) in which two parallel exothermic reactions take place. The reactor is a system with parametric uncertainty and multiple steady states. The problem of robust controller design was converted to a problem of solution of linear matrix inequalities (LMIs) and computationally simple non-iterative and iterative algorithms can be used for controller tuning. The MATLAB–Simulink environment enables to compare the results of the YALMIP and the Robust Control toolboxes.

Keywords: chemical reactor, multiple steady states, robust stabilization, static output feedback

1. INTRODUCTION

Continuous-time stirred tank reactors (CSTRs) belong to the most important plants in chemical and food industries. From the control viewpoint, CSTRs are very interesting systems, because of their potential safety problems and the possibility of exotic behavior such as multiple steady states, see e.g. Molnár et al. (2002). Furthermore, operation of chemical reactors is corrupted by various uncertainties. Some of them arise from varying or not exactly known parameters, as e.g. reaction rate constants, reaction enthalpies, heat transfer coefficients, etc. Operating points of reactors change in other cases. All these uncertainties can cause poor performance or even instability of closed-loop control systems. Application of robust control approach is one way for overcoming all these problems, as it is shown e.g. in Alvarez-Ramirez and Femat (1999), Gerhard et al. (2004), Bakošová et al. (2005), Tlacuahuac et al. (2005) and others. From the viewpoint of safety operation or in the case when the unstable steady state coincides with the point that yields the maximum reaction rate at a prescribed temperature, it is necessary to control CSTRs into the prescribed open-loop unstable steady state (Bakošová and Oravec (2010), Bakošová et al. (2009), Bakošová et al. (2006), Puna et al. (2006)).

One of solved problems in robust control theory is the problem of robust static output feedback control (Dong and Yang (2007), Iwasaki et al. (1994), Syrmos et al. (1997) and references therein). This approach can be successfully used for solving the problem of robust stabilization of CSTRs. For obtaining robust stabilizing controllers, the non-iterative and iterative algorithms can be applied (Veselý (2002)).

In this paper, the problem of robust stabilization of a CSTR is solved. The conditions for robust stabilization are formulated in the form of linear matrix inequalities (LMIs). Solution of LMIs represents a convex optimization problem that has been solved in the MATLAB environment by Robust Control toolbox (Balas et al. (2006)) and YALMIP toolbox (Löfberg (2004), Kvasnica and Fikar (2010)) with solver SeDuMi (Henrion and Lasserre (2003)).

2. CONTROLLED CSTR

The controlled reactor is a continuous-time stirred tank reactor with two first order irreversible parallel exothermic reactions according to the scheme A $\xrightarrow{k_1}$B, A $\xrightarrow{k_2}$ C, where B is the main product and C is the side product. Chemical reactions are performed in a reaction vessel and reaction heat is removed from the reactor by coolant in a reactor jacket. Because of the exponential dependency of reactant concentrations on the temperature of the reaction mixture known as the Arrhenius equation (Molnár et al. (2002)), it is supposed that it is not necessary to control directly concentrations. The multivariable controller is used in order to achieve control of the reaction mixture temperature in the reaction vessel and the coolant temperature in the jacket. Control inputs are flow rates of reaction mixture and coolant. Parameters and inputs of the considered CSTR (Bakošová et al. (2006)) are shown in Table 1 and Table 2.

Model uncertainties of the over described reactor follow from the fact that there are four only approximately known physical parameters in the reactor, which values are shown in Table 3. Here, $\Delta_r H_1$, $\Delta_r H_2$ are reaction enthalpies of the chemical reactions and $k_{\infty 1}$, $k_{\infty 2}$ are

Table 1. Parameters of CSTR

Variable	Value	Unit
V	0.23	m^3
V_C	0.21	m^3
ρ	1020	$\mathrm{kg\ m}^{-3}$
ρ_C	998	$\mathrm{kg\ m}^{-3}$
c_P	4.02	$\mathrm{kJ\ kg}^{-1}\mathrm{K}^{-1}$
c_{P_C}	4.182	$\mathrm{kJ\ kg}^{-1}\mathrm{K}^{-1}$
A	1.51	m^2
α	42.8	$\mathrm{kJ\ min}^{-1}\mathrm{m}^{-2}\mathrm{K}^{-1}$
$g_1 = E_1/R$	9850	K
$g_2 = E_2/R$	22019	K

Table 2. Steady-state inputs of CSTR

Variable	Value	Unit
$c_{A,0}$	0.0824	$\mathrm{kmol\ m}^{-3}$
$c_{B,0}$	0	$\mathrm{kmol\ m}^{-3}$
T_0	310	K
$T_{C,0}$	288	K
q^s	0.015	$\mathrm{m}^3\mathrm{min}^{-1}$
q^s_C	0.004	$\mathrm{m}^3\mathrm{min}^{-1}$

pre-exponential factors in the reaction rate constants. The nominal values of uncertain parameters are considered to be mean values of given intervals. These uncertainties represent parametric uncertainties.

Table 3. Uncertain parameters of CSTR

Variable	Minimal value	Maximal value	Unit
$\Delta_r H_1$	-8.8×10^4	-8.4×10^4	$\mathrm{kJ\ kmol}^{-1}\ \mathrm{min}^{-1}$
$\Delta_r H_2$	-5.7×10^4	-5.3×10^4	$\mathrm{kJ\ kmol}^{-1}\ \mathrm{min}^{-1}$
$k_{\infty 1}$	1.5×10^{11}	1.6×10^{11}	min^{-1}
$k_{\infty 2}$	4.95×10^{26}	12.15×10^{26}	min^{-1}

It follows from the steady-state analysis that the reactor has three steady states, two of them are stable and one is unstable. The situation for the nominal model is shown in Figure 1, where the curve Q_{GEN} (red line) represents the heat generated by the reactions and the line Q_{OUT} (blue line) represents the heat withdrawn from the reactor. The steady-state operating points of the reactor are points, where the curve and the line intersect. The steady states are stable if the slope of the cooling line is higher then the slope of the heat generated curve. This condition is satisfied in the steady states at the temperatures $T = 308.4$ K and $T = 352.6$ K, and it is not satisfied in the steady state at $T = 338.4$ K. The steady-state behavior of the chemical reactor is similar for all vertex systems, which are obtained for all combinations of minimal and maximal values of uncertain parameters. The maximal concentration of the main product B (red line) is always obtained when the reactor operates in the unstable steady state as it is shown in Figure 2 (Bakošová et al. (2006)).

Linearized mathematical model of the reactor has been derived under the assumption that the control inputs are the reactant flow rate q and the coolant flow rate q_C, the controlled outputs are the reaction mixture temperature T and the coolant temperature T_C and the operating point of the reactor is its open-loop unstable steady state. Then the linearized model of the CSTR is in the form

$$\begin{aligned}\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t)\end{aligned} \tag{1}$$

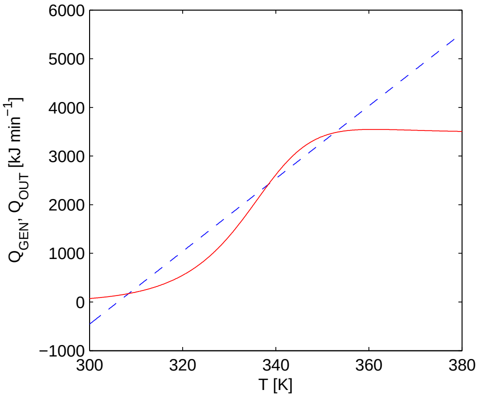


Fig. 1. Steady states of CSTR with nominal values of uncertain parameters

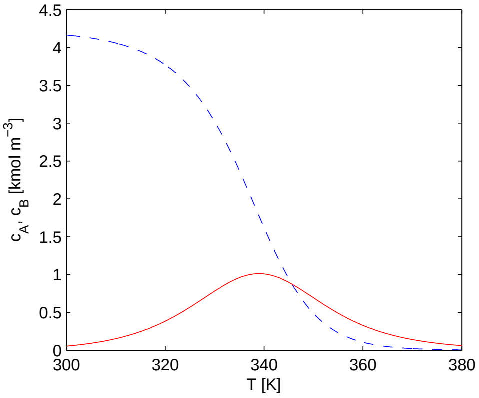


Fig. 2. Concentration of components A and B in dependence on the reaction mixture temperature – nominal model

where again $\boldsymbol{x}(t) \in R^n$ is the state, $\boldsymbol{u}(t) \in R^m$ the control, $\boldsymbol{y}(t) \in R^r$ the output. Matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are in the form

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & 0 & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & 0 & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{pmatrix} \tag{2}$$

$$\boldsymbol{B} = \begin{pmatrix} b_{11} & 0 \\ b_{21} & 0 \\ b_{31} & 0 \\ 0 & b_{42} \end{pmatrix} \tag{3}$$

$$\boldsymbol{C} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

Matrices $\boldsymbol{A}$, $\boldsymbol{B}$ have varying coefficients as according to the values of uncertain parameters steady states of the reactor vary. For coefficients of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ see Table 4.

For all combinations of boundary values of 4 uncertain parameters, we have obtained $2^4 = 16$ linearized mathematical models with matrices $\boldsymbol{A}_i$, $\boldsymbol{B}_i$. These systems represent vertices of an uncertain polytopic system. All these vertices are unstable systems as between the eigenvalues of $\boldsymbol{A}_i$, $i = 1, ..., 16$, are also positive eigenvalues. Unstable is also the linearized nominal model (Bakošová et al. (2006)).

Table 4. Matrices parameters

Parameter	Value
a_{11}	$-\left(\frac{q^s}{V}+k_1^s+k_2^s\right)$
a_{13}	$-\frac{c_A^s(k_1^s g_1+k_2^s g_2)}{(T^s)^2}$
a_{21}	k_1^s
a_{22}	$-\frac{q^s}{V}$
a_{23}	$\frac{k_1^s\Delta_r H_1+k_2^s\Delta_r H_1}{q^s c_P}$
a_{31}	$-\left(\frac{q^s}{V}+\frac{\alpha A}{V\rho c_p}+\frac{c_A^s(k_1^s g_1\Delta_r H_1+k_2^s g_2\Delta_r H_1)}{\rho c_P(T^s)^2}\right)$
a_{34}	$\frac{\alpha A}{V\rho c_P}$
a_{43}	$\frac{\alpha A}{V_C\rho_C c_{P_C}}$
a_{44}	$-\left(\frac{q_C^s}{V_C}+\frac{\alpha A}{V_C\rho_C c_C}\right)$
b_{11}	$\frac{c_{A,0}-c_A^s}{V}$
b_{21}	$\frac{c_{B,0}-c_B^s}{V}$
b_{31}	$\frac{T_0-T^s}{V}$
b_{42}	$\frac{T_{C,0}-T_C^s}{V}$

3. ROBUST STATIC OUTPUT FEEDBACK STABILIZATION OF CSTR

Design of a robust static output feedback controller is based on having a linear time-invariant state space model (1) of the controlled system. For the system (1), it is necessary to find a static output feedback $\boldsymbol{u}(t) = \boldsymbol{F}\boldsymbol{y}(t)$. Using this static output feedback we obtain an uncertain polytopic closed-loop system

$$\dot{\boldsymbol{x}}(t) = [\boldsymbol{A} + \boldsymbol{B}\boldsymbol{F}\boldsymbol{C}]\,\boldsymbol{x}(t) = \boldsymbol{A}_{CL}\boldsymbol{x}(t) \tag{5}$$

The system (1) is simultaneously static output feedback stabilizable with guaranteed cost

$$\int\limits_0^\infty \left(\boldsymbol{x}(t)^T\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}(t)^T\boldsymbol{R}\mathbf{u}(t)\right)dt \le \boldsymbol{x}_0^T\boldsymbol{P}\boldsymbol{x}_0 = J^* \qquad \boldsymbol{P} > 0 \tag{6}$$

if there exist matrices $\boldsymbol{P} > 0, \boldsymbol{Q} > 0, \boldsymbol{R} > 0$ and a matrix $\boldsymbol{F}$ such that the following inequalities hold (Veselý (2002))

$$\boldsymbol{\Omega}_i^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{\Omega}_i + \boldsymbol{Q} + \boldsymbol{C}_i^T\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{F}\boldsymbol{C}_i < 0 \qquad i = 1,\dots,N \tag{7}$$

where

$$\boldsymbol{\Omega}_i = \boldsymbol{A}_i + \boldsymbol{B}_i\boldsymbol{F}\boldsymbol{C}_i \tag{8}$$

The system (1) is simultaneously static output feedback stabilizable with guaranteed cost (6) also if there exist matrices $\boldsymbol{P} > 0, \boldsymbol{Q} > 0, \boldsymbol{R} > 0$ and a matrix $\boldsymbol{F}$ such that the inequalities hold (Veselý (2002))

$$\boldsymbol{A}_i^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P}\boldsymbol{B}_i\boldsymbol{R}^{-1}\boldsymbol{B}_i^T\boldsymbol{P} - \Theta_i + \boldsymbol{Q} \le 0 \qquad i = 1,\dots,N \tag{9}$$

where

$$\boldsymbol{\Theta}_i = \boldsymbol{C}_i^T\boldsymbol{F}^T\boldsymbol{R}\boldsymbol{F}\boldsymbol{C}_i \tag{10}$$

and also the inequalities hold

$$\lambda_i\phi_i^{-1}\lambda_i^T - \boldsymbol{R} \le 0 \qquad i = 1,\dots,N \tag{11}$$

where

$$\lambda_i = \boldsymbol{B}_i^T\boldsymbol{P} + \boldsymbol{R}\boldsymbol{F}\boldsymbol{C}_i \tag{12}$$

$$\phi_i = -(\boldsymbol{A}_i^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_i - \boldsymbol{P}\boldsymbol{B}_i\boldsymbol{R}^{-1}\boldsymbol{B}_i^T\boldsymbol{P} - \boldsymbol{\Theta}_i + \boldsymbol{Q}) \tag{13}$$

The non-iterative and iterative procedures for simultaneous static output feedback stabilization of the system (1) with guaranteed cost (6) are based on statements formulated above (Veselý (2002)).

3.1 Non - iterative algorithm

Using the Schur complement formula and defining $\mathbf{S}$ is equal to $\boldsymbol{P}^{-1}$ and considering $\boldsymbol{\Theta}_i$ is equal zero, the inequality (9) is transformed to the following LMIs

$$\begin{bmatrix} \boldsymbol{S}\boldsymbol{A}_i^T + \boldsymbol{A}_i\boldsymbol{S} - \boldsymbol{B}_i\boldsymbol{R}^{-1}\boldsymbol{B}_i^T & \boldsymbol{S}\sqrt{\boldsymbol{Q}} \\ \sqrt{\boldsymbol{Q}}\boldsymbol{S} & -\boldsymbol{I} \end{bmatrix} \le 0 \qquad \gamma\boldsymbol{I} < \boldsymbol{S}, \quad i = 1,\dots,N \tag{14}$$

where $\gamma > 0$ is any positive constant.

Using $\boldsymbol{P} = \boldsymbol{S}^{-1}$, the inequality (11) can be rewritten to the following LMIs

$$\begin{bmatrix} -\boldsymbol{R} & \boldsymbol{B}_i^T\boldsymbol{P} + \boldsymbol{R}\boldsymbol{F}\boldsymbol{C}_i \\ (\boldsymbol{B}_i^T\boldsymbol{P} + \boldsymbol{R}\boldsymbol{F}\boldsymbol{C}_i)^T & -\phi_i \end{bmatrix} \le 0 \qquad i = 1,\dots,N \tag{15}$$

The non-iterative algorithm for static output simultaneous stabilization of the system (1) with the guaranteed cost (6) is following (Veselý (2002)).

(1) Set parameter γ and required values of the weight matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ in the cost function (6).
(2) Compute $\boldsymbol{S} = \boldsymbol{S}^T > 0$ from the inequalities (14). If the solution of (14) is not feasible, the system (1) is not simultaneously stabilizable by static output feedback.
(3) Set $\boldsymbol{P} := \boldsymbol{S}^{-1}$.
(4) Compute $\boldsymbol{F}$ from the inequalities (15). If the solution of (15) is not feasible, the closed-loop system (5) is not quadratically stable with guaranteed cost. Then change $\boldsymbol{Q}$, $\boldsymbol{R}$ or γ in order to find feasible solutions.
(5) If the solutions of (14), (15) are feasible, then the system (1) is simultaneously stabilizable and the system (5) is quadratically stable with guaranteed cost control algorithm $\boldsymbol{u}^*(t) = \boldsymbol{F}\boldsymbol{y}(t)$ and $J^* = \boldsymbol{x}_0^T\boldsymbol{P}\boldsymbol{x}_0$ is the guaranteed cost.

3.2 Iterative algorithm

Using the Schur complement formula and defining $\boldsymbol{S}$ is equal to $\boldsymbol{P}^{-1}$, the inequality (7) is transformed to the following LMIs

$$\begin{bmatrix} \boldsymbol{S}_k\boldsymbol{A}_i^T + \boldsymbol{A}_i\boldsymbol{S}_k - \boldsymbol{B}_i\boldsymbol{R}^{-1}\boldsymbol{B}_i^T & \boldsymbol{S}_k\sqrt{\boldsymbol{\Psi}_i} \\ \sqrt{\boldsymbol{\Psi}_i}\boldsymbol{S}_k & -\mathbf{I} \end{bmatrix} \le 0 \qquad \gamma\boldsymbol{I} < \boldsymbol{S}_k, \quad i = 1,\dots,N \tag{16}$$

where

$$\boldsymbol{\Psi}_i = \boldsymbol{C}_i^T \boldsymbol{F}_{k-1}^T \boldsymbol{R} \boldsymbol{F}_{k-1} \boldsymbol{C}_i + \boldsymbol{Q} \tag{17}$$

and $\gamma > 0$ is any positive constant.

Using $\boldsymbol{P} = \boldsymbol{S}^{-1}$, the inequality (11) can be rewritten to the following LMIs

$$\begin{bmatrix} -\boldsymbol{R} & \boldsymbol{B}_i^T \boldsymbol{P} + \boldsymbol{R}\boldsymbol{F}_i \boldsymbol{C}_i & \boldsymbol{0} \\ \boldsymbol{P}_k \boldsymbol{B}_i + \boldsymbol{C}_i^T \boldsymbol{F}_k^T \boldsymbol{R} & \varphi_i & \boldsymbol{C}_i \boldsymbol{F}_k^T \\ \boldsymbol{0} & \boldsymbol{F}_k \boldsymbol{C}_i & \boldsymbol{R} \end{bmatrix} \leq 0$$

$$i = 1, \ldots, N \tag{18}$$

where

$$\varphi_i = \boldsymbol{A}_i^T \boldsymbol{P}_k + \boldsymbol{P}_k \boldsymbol{A}_i^T - \boldsymbol{P}_k \boldsymbol{B}_i^T \boldsymbol{R}^{-1} \boldsymbol{B}_i^T \boldsymbol{P}_k + \boldsymbol{Q} \tag{19}$$

The iterative algorithm for static output simultaneous stabilization of the system (1) with the guaranteed cost (6) is following (Veselý (2002)).

(1) Set parameter γ and required values of the weight matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ due to the cost function (6).
(2) Set $k := 0$ and initial value of matrix $\boldsymbol{F}_0$.
(3) Set $k := k + 1$.
(4) Compute $\boldsymbol{S}_k = \boldsymbol{S}_k^T > 0$ from the inequalities (16). If the solution of (16) is not feasible, the system (18) is not simultaneously stabilizable by static output feedback.
(5) Set $\boldsymbol{P}_k := \boldsymbol{S}_k^{-1}$.
(6) Compute $\boldsymbol{F}_k$ from the inequalities (18). If the solution of (18) is not feasible, the closed-loop system (5) is not quadratically stable with guaranteed cost. Then change $\boldsymbol{Q}$, $\boldsymbol{R}$ or γ in order to find feasible solutions.
(7) If $||\boldsymbol{F}_k - \boldsymbol{F}_{k-1}|| \leq$ *tolerance* then stop else go to the *Step* 3.
(8) If the solutions of (16), (18) are feasible, then the system (1) is simultaneously stabilizable and the system (5) is quadratically stable with guaranteed cost control algorithm $\boldsymbol{u}^*(t) = \boldsymbol{F}\boldsymbol{y}(t)$ and $J^* = \boldsymbol{x}_0^T \boldsymbol{P} \boldsymbol{x}_0$ is the guaranteed cost.

4. CASES OF ROBUST CONTROLLER DESIGN

Solving presented algorithms represents the feasibility problem of convex optimization. The MATLAB environment enables to solve this problem by Robust Control toolbox and YALMIP toolbox with solver SeDuMi. The Robust Control toolbox uses function `setlmis` to initialize the LMI generating. Function `lmivar` enables to define the properties of optimization variable. Function `getlmis` generates LMI in the form required for processing by the function `feasp`. This function enables to solve the LMI feasibility optimization problem. The YALMIP uses function `sdpvar` to set the properties of optimization variable. The constraints are set simple by using the parentheses in the form `[ expression ]` . Function `solvesdp` enables to solve the optimization problem. To obtain the calculated value, the function `double` can be used.

For robust controller design the above presented non-iterative algorithm has been applied. The values of matrices $\boldsymbol{Q}$, $\boldsymbol{R}$ in the cost function (6) and parameter γ used for controller tuning are shown in Table 5.

Table 5. Parameters for controller design

Cost function	Q	R	γ
1	$\begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$	0.001
2	$\begin{pmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{pmatrix}$	$\begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$	0.001

Due to the various weight matrices in the cost function (6) and using different MATLAB toolboxes, the four different cases of controller tuning have been obtained (Table 6). The following initial conditions of system (1) have been considered (Bakošová and Oravec (2010))

$$\boldsymbol{x}_0 = (\Delta c_A, \Delta c_B, \Delta T, \Delta T_C)^T \tag{20}$$

$$\begin{aligned} \boldsymbol{x}_0 &= (2.1210, 0.8644, 335.4726, 325.7271)^T - \\ &- (1.8614, 1.0113, 338.4080, 328.0599)^T = \\ &= (0.2596, -0.1469, -2.9354, -2.3328)^T \end{aligned} \tag{21}$$

Table 6. Cases of the robust controller design

Case	Method	Cost function	Used toolbox
1	non-iterative	1	*Robust Control*
2	non-iterative	1	YALMIP
3	non-iterative	2	*Robust Control*
4	non-iterative	2	YALMIP

5. RESULTS AND DISCUSSION

Four different controllers (F_1–F_4) for considered system (1) have been designed for various cases of the robust stabilization controller tuning (Table 6). The designed controllers are shown in Table 7. In this table are shown the maximal evaluated values of cost function J in comparison to guaranteed values of the cost function J^* for each designed controller. These values have been evaluated for considered initial values (21). The maximal eigenvalue of all uncertain systems (EV_{max}) have been calculated. The time measured in seconds needed for each LMI solving procedure is shown in column t_{CPU}. These data have been evaluated by computer with 3.20 GHz CPU and memory 4 GB RAM.

Table 7. Properties of the designed robust static output feedback controllers

Case	Controller F_c	J^*	J	EV_{max}	t_{CPU}
1	$\begin{bmatrix} 0.0456 & 0.0294 \\ 0.0569 & 0.0496 \end{bmatrix}$	0.2765	0.1286	−0.016	$\begin{bmatrix} 0.08 \\ 0.05 \end{bmatrix}$
2	$\begin{bmatrix} 0.0026 & -0.0026 \\ 0.0179 & 0.0160 \end{bmatrix}$	0.3221	0.2151	−0.036	$\begin{bmatrix} 0.58 \\ 0.25 \end{bmatrix}$
3	$\begin{bmatrix} 0.0333 & 0.0170 \\ 0.0261 & 0.0189 \end{bmatrix}$	0.7340	0.2638	−0.020	$\begin{bmatrix} 0.08 \\ 0.05 \end{bmatrix}$
4	$\begin{bmatrix} 0.0028 & -0.0029 \\ 0.0020 & 0.0005 \end{bmatrix}$	0.5546	0.4158	−0.033	$\begin{bmatrix} 0.51 \\ 0.47 \end{bmatrix}$

All the designed controllers assure the lower value of cost function J than the guaranteed value J^* as the theory predicts. All designed controllers guarantee the negative value of maximal eigenvalue for all uncertain closed–loop systems. The lower values of cost function J have been obtained for the controllers tuned by Robust Control toolbox.

In general, YALMIP toolbox needs more CPU time for solving the optimization problem than Robust Control toolbox. On the other hand, the YALMIP toolbox offers more comfortable environment. For system (1) and all the controllers F_1–F_4, closed–loop behaviours have been generated with corresponding control inputs using MATLAB–Simulink environment. In the Figures 3 – 12 are shown performances of the closed–loop system controlled using controllers designed by non-iterative algorithm.

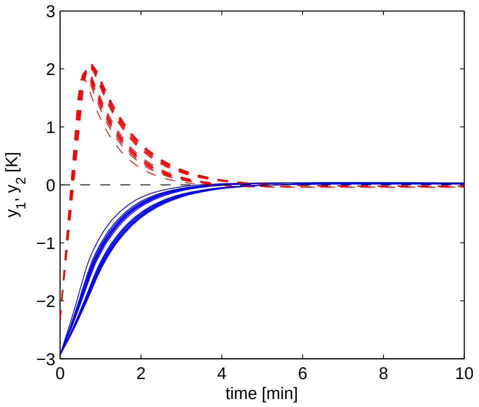


Fig. 3. Closed-loop behaviour of CSTR using the robust controller F_1

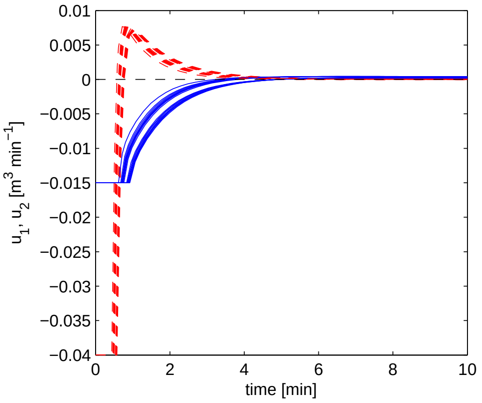


Fig. 4. Control inputs generated by the controller F_1

The controllers F_1, F_2 have been designed for the *cost function 1* and the controllers F_3, F_4 have been tuned for the *cost function 2* (Table 6). As can be seen in the Figure 3 and Figure 5 the control performances of the CSTR assured by the controllers F_1, F_2 obtain similar *overshoot* (Bakošová et al. (2003)). The Figure 3 show that using the controller F_1 tuned by Robust Control toolbox leads to the lower value of *settling time* (Bakošová et al. (2003)) in comparison to the control performance assured by the controller F_2 tuned by YALMIP (Figure 5). The control performance assured by the controller F_4 tuned by YALMIP toolbox obtain lower value of *overshoot* (Figure 9) than is assured by using the controller F_3 (Figure 7). On the other hand, the *setling time* obtained using the controller F_4 is much longer (Figures 11, 12) in comparison to the performance assured by the controller F_3 (Figure 7, 8) tuned by Robust Control toolbox.

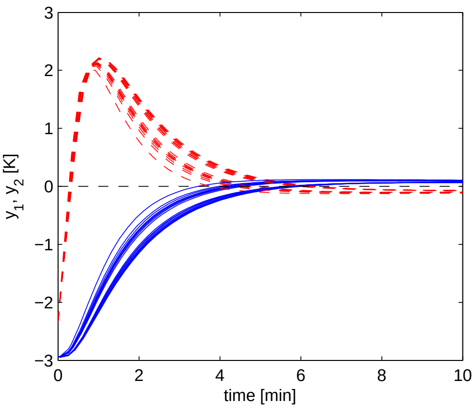


Fig. 5. Closed-loop behaviour of CSTR using the robust controller F_2

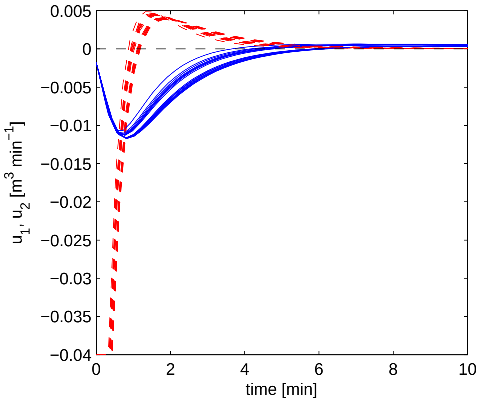


Fig. 6. Control inputs generated by the controller F_2

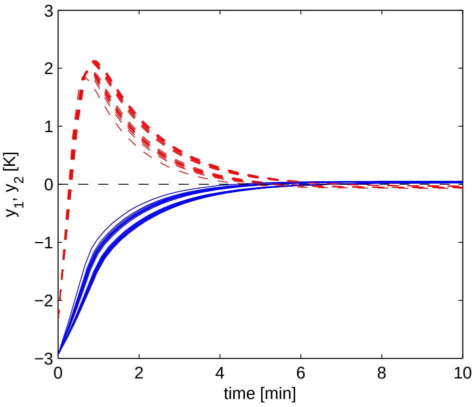


Fig. 7. Closed-loop behaviour of CSTR using the robust controller F_3

6. CONCLUSION

Robust stabilization of the exothermic CSTR with four uncertain parameters using static output feedback controllers was studied. The robust stabilizing multivariable controllers have been designed using the presented simple non-iterative and iterative algorithms, which are based on solving of two sets of LMIs. The problem of their solutions represents the feasibility problem of convex optimization.

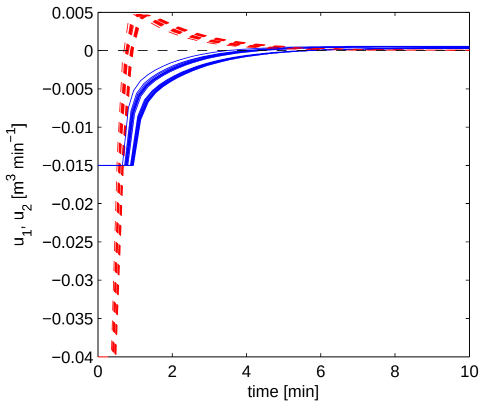


Fig. 8. Control inputs generated by the controller F_3

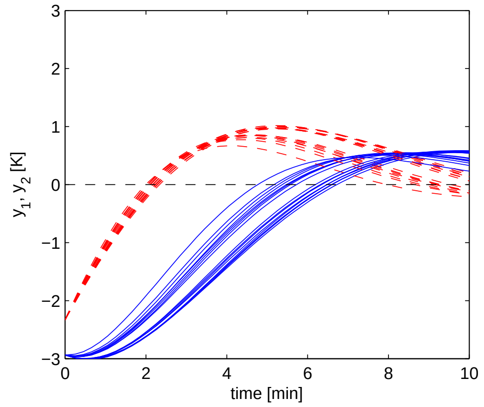


Fig. 9. Closed-loop behaviour of CSTR using the robust controller F_4

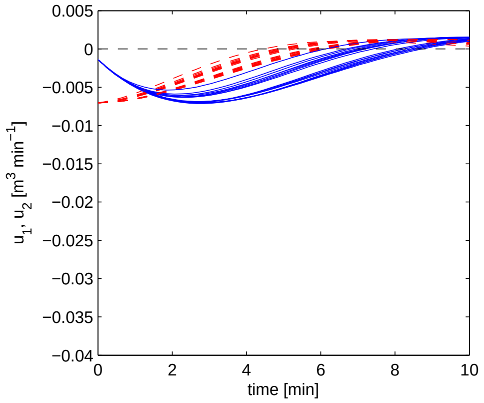


Fig. 10. Control inputs generated by the controller F_4

The Robust Control toolbox and YALMIP toolbox have been used for solving the LMIs and the results have been compared using MATLAB–Simulink environment. Description of optimization problem using YALMIP is more user-friendly. On the other hand, the CPU time decreases using Robust Control toolbox. Despite of simple using of YALMIP, using Robust Control toolbox seems to be more suitable for solving the problem of robust static output feedback stabilization.

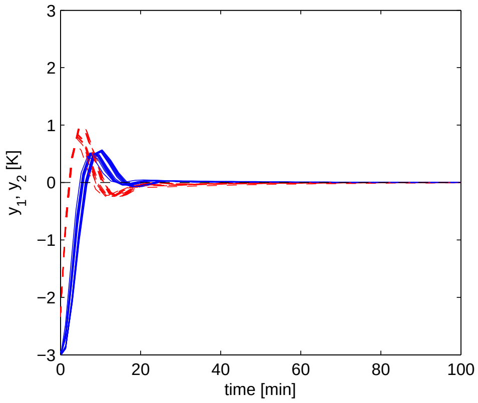


Fig. 11. Closed-loop behaviour of CSTR using the robust controller F_4 using longer evaluation time

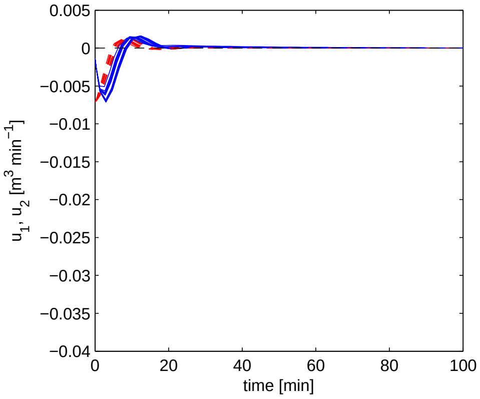


Fig. 12. Control inputs generated by the controller F_4 using longer evaluation time

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency VEGA of the Slovak Republic under the grants 1/0537/10, 1/0095/11.

REFERENCES

Alvarez-Ramirez, J. and Femat, R. (1999). Robust PI stabilization of a class of chemical reactors. *Sytems and Control Letters*, 38, 219–225.

Bakošová, M. and Oravec, J. (2010). Solutions of LMIs in the problem of robust stabilization of chemical reactors. *VOCAL 2010, Program and Abstracts*, 47.

Bakošová, M., Puna, D., Dostál, D., and Závacká, J. (2009). Robust stabilization of a chemical reactor. *Chemical Papers*, 5, 527–536.

Bakošová, M., Puna, D., and Mészáros, A. (2005). Robust controller design for a chemical reactor. In L. Puigjaner and A. Espuna (eds.), *European Symposium on Computer Aided Process Engineering - 15*, 1303–1308. Elsevier, Amsterdam.

Bakošová, M., Puna, D., and Mészáros, A. (2006). Control of a continuous-time stirred tank reactor via robust static output feedback. *V Proceedings of the 14th IEEE Mediterranean Conference on Control and Automation.*

Bakošová, M., Ľ. Čirka, and Fikar, M. (2003). *Automatic control fundaments (in Slovak).* Vydavateľstvo STU, 1st ed. edition.

Balas, G., Chiang, R., Packard, A., and Safonov, M. (2006). *Robust Control Toolbox. For use with MATLAB.* The MathWorks. Available from http://www.mathworks.com/products/robust/.

Dong, J. and Yang, G. (2007). Robust static output feedback control for linear discrete-time systems with time-varying uncertainties. *Systems & Control Letters*, 57, 123–131.

Gerhard, J., Monningmann, M., and Marquardt, W. (2004). Robust stable nonlinear control and design of a CSTR in a large operating range. In *Proceedings of the 7th International Symposium on Dynamics and Control of Process Systems*, 92.pdf. Massachusetts, USA.

Henrion, D. and Lasserre, J. (2003). Gloptipoly: Global optimizaton over polynomials with matlab and SeDuMi. *ACM Transactions on Mathematical Software, Vol. 29, No. 2*, 165–194.

Iwasaki, T., Skelton, R.E., and Geromel, J.C. (1994). Linear quadratic suboptimal control with static output feedback. *Systems Control Lett.*, 23, 421–430.

Kvasnica, M. and Fikar, M. (2010). Design and Implementation of Model Predictive Control using Multi-Parametric Toolbox and YALMIP. In *Proceedings of the 2010 IEEE International Symposium on Computer-Aided Control System Design*, 999–1004. Yokohama, Japan.

Löfberg, J. (2004). Yalmip: A toolbox for modeling and optimization in matlab. *In Proc. of the CACSD Conference, Taipei, Taiwan. Available from http://control.ee.ethz.ch/joloef/yalmip.php*, 284–289.

Molnár, A., Markoš, J., and Ľ. Jelemenský (2002). Accuracy of mathematical model with regard to safety analysis of chemical reactors. *Chemical Papers*, 56, 357–361.

Puna, D., Bakošová, M., Mészáros, A., and Závacká, J. (2006). Control of a chemical reactor with uncertainties in an unstable steady state. *V Proceedings of the 33rd International Conference of SSCHE,Tatransk Matliare.*

Syrmos, V.L., Abdallah, C.T., Dorato, P., and Grigoriadis, K. (1997). Static output feedback. a survey. *Automatica*, 33, 203–210.

Tlacuahuac, A.F., Alvarez, J., Guerra, E.S., and Oaxaca, G. (2005). Optimal transition and robust control design for exothermic continuous reactors. *AIChE Journal*, 51, 895–908.

Veselý, V. (2002). Robust output feedback controller design for linear parametric uncertain systems. *Journal of Electrical Engineering*, 53, 117–125.

Lateral control for flexible BWB high-capacity passenger aircraft

Tomáš Haniš * Martin Hromčík **

* *T. Haniš, Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic (e-mail: hanist1@fel.cvut.cz).*
** *M. Hromčík, Center for Applied Cybernetics, Czech Technical University in Prague, Faculty of Electrical Engineering, Karlovo namesti 13-G, Prague, Czech Republic (e-mail: xhromcik@control.felk.cvut.cz).*

Abstract: Two different approaches for design of lateral control augmentation system for large blended-wing-body aircraft (BWB) with flexible structure are presented and asses in this paper. The most challenging issue is handling of rigid-body dynamics and flexible modes coupling. First, a more classical approach is employed giving rise to separate flight dynamics controller (H2 optimal, with sufficient roll-off) and an active damper for most prominent lateral flexible modes on top of that (mixed-sensitivity H_∞ design). This approach proves successful and has obvious advantages related to the design process complexity, or implementation and testing issues. On the other hand, there is always a risk of potentially significant performance loss compared to a fully integrated design. For this reason, fully integrated design is also presented in the form of a fixed-order MIMO H_∞ optimal FCS controller, obtained by means of direct non-convex non-smooth optimization package HIFOO. Performance of both approaches is assessed.

Keywords: Lateral control; Fixed order optimization; BWB aircraft

1. INTRODUCTION

Large aircraft structures and novel concepts, such as Blended Wing Body (BWB) aircraft configurations, can lead to higher fuel-efficiency and reduced emissions. However, this also leads to low frequency structure vibration modes, and coupling of those to the flight mechanic modes may occur. Also, BWB concepts are expected to show coupling between longitudinal and lateral dynamics. This and significant parameter dependency of the aircraft dynamics pose significant design challenges for developing robust and well-performing flight control laws. Traditional methods for flight control design typically use nested SISO control loops and strongly structured control architectures (6). These methods are based on detailed aircraft system analysis and exploit paths with weak coupling to obtain good results for conventional flight control design. However, multivariate methods, such as optimal control and particularly robust control design methods are state of the art for more complex flight control tasks under coupled and/or uncertain system dynamics. Two large groups of control design methodologies are optimal control design methods (e.g., LQG control and the Kalman estimator (4), (3)), as well as robust control design methods (see (8) and (5) for fundamentals, or (2) for an aerospace-specific overview). This work reports first findings from ongoing research connected to the control design for a large BWB passenger aircraft.

Two different approaches to lateral MIMO feedback Control Augmentation System (CAS) for NACRE BWB aircraft are presented in the following. They are namely a robust MIMO H_2/H_∞ mixed sensitivity controller and a low-order robust MIMO H_∞ optimal controller designed by direct fixed-order control design techniques. All controllers are designed to assure for desired closed-loop rigid-body response (namely rise time and no-overshoot behavior to the reference change of the bank angle set point, attenuation of beta disturbance, and required damping ratio of the DR mode) and to damp first two antisymmetric wings flexible modes. Performance and robustness of all controllers is demonstrated by means of MATLAB/Simulink simulations, and their advantages and drawbacks are discussed to arrive at conclusions. More details about BWB aircraft control issues can by found in (9), (10), (11), (12) and (13).

2. BLENDED WING BODY AIRCRAFT

ACFA 2020 is a collaborative research project funded by the European Commission under the seventh research framework programme (FP7). The project deals with innovative active control concepts for ultra efficient 2020 aircraft configurations like the blended wing body (BWB) aircraft (see Fig. 1 and 2). The Advisory Council for Aeronautics Research in Europe (ACARE) formulated the "ACARE vision 2020", which aims for 50% reduced fuel consumption and related CO_2 emissions per passenger-kilometre and reduction of external noise. To meet these goals is very important to minimize the environmental impact of air traffic but also of vital interest for the aircraft

⋆ This is preliminary version of paper submitted for IFAC 2011.

industry to enable future growth. Blended Wing Body type aircraft configurations are seen as the most promising future concept to fulfill the ACARE vision 2020 goals because aircraft efficiency can be dramatically increased through minimization of the wetted area and reducing of structural load and vibration by active damping in a integrated control law design (addopted from (1)).

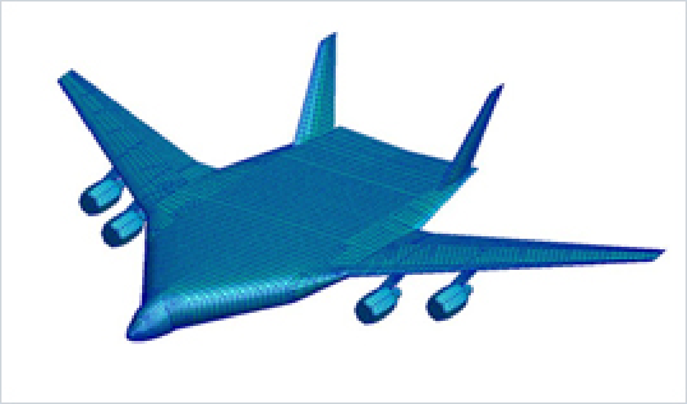


Fig. 1. BWB FEM structure.

Fig. 2. BWB visualization.

3. BLENDED WING BODY AIRCRAFT MATHEMATICAL MODEL

Mathematical model of BWB aircraft used for control law design consist of aircraft model itself, model of actuators and sensors. Actuators models are considered as 2^{nd} order linear models augmented by saturations and rate limiters. Sensors are modeled as 2^{nd} order Butterworth filters with time delays approximated by 2^{nd} order Padde approximation. Mathematical model of aircraft consist of rigid body description (modeled as a 12^{th} order linear system separated to longitudinal and lateral dynamics), flexible modes (for design purposes just four modes are considered, with rise to 8^{th} order linear model) and lag states. Overall model used for control law design is of order 52.

4. H2/H$_\infty$ MIXED SENSITIVITY CONTROLLER

A two-stage control law is devised - separate control augmentation system (CAS) taking care of the flight-dynamics (robust H$_2$ optimal roll autopilot, with roll-off at higher frequencies), and an active damper for selected flexible modes (H$_\infty$ optimal mixed-sensitivity controller tuned to first two antisymmetric wing bending modes). Such an arrangement has obvious advantages - regarding tuning (both parts are designed/tuned independently), future flight testing (the active damper can be tested after the roll autopilot is implemented and approved, and it can be turned on/off at any time while keeping the aircraft well controlled), safety (loss of the damper's functionality, e.g. due to sensors failure, does not take the airplane out of control). The drawback is potential reduction of performance compared to a fully integrated design where both flight dynamics and vibrational issues are handled by a single large multiple input multiple output (MIMO) controller.

4.1 design method

The lateral CAS (roll autopilot) is designed by H$_2$ norm minimization of the generalized plant, encompassing the lateral rigid body dynamics itself (4 states/outputs), 2 integrators (to assure for perfect steady-state tracking of roll angle set point command and for perfect steady-state attenuation of beta disturbance), and two low-pass filters (for required roll-off at higher frequencies - so that the flexible modes are left untouched, not excited by the controller). As all the rigid body (RB) states are measured, the observer needs not be implemented in fact and the resulting order of this CAS can be kept quite small (six states). Resulting controller features robust stability/performance for all considered mass cases (3 passengers and 5 fuel cases).

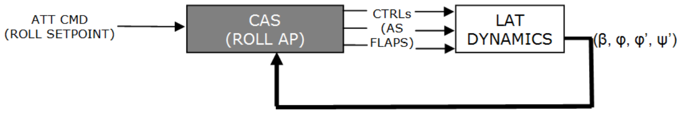


Fig. 3. Control augmentation system for H$_2$ controller design. Where control surfaces are considered as anti-symmetrically driven wings ailerons.

On top of that, a robust MIMO controller is built by minimization of the H$_\infty$ norm of the frequency weighted mixed-sensitivity function. Wings modal antisymmetric sensor and antisymmetric flaps make up the input/output groups. Loosely speaking, the closed loop sensitivity function is kept small at selected frequency regions (in our case covering the wing antisymmetric modes) to assure for good performance (disturbance attenuation) while the complementary sensitivity function is kept small everywhere else (to assure for robustness - the design model becomes invalid outside the selected frequency region). A simple design model of 8th order was constructed (modeling accurately the two modes and close region in the I/O channels). Two resonant weighting filters of 2^{nd} order are tuned to the frequencies and dampings of the antisymmetric wing bending modes of a selected representative case for this purpose. Resulting H$_\infty$ controller has 20 states.

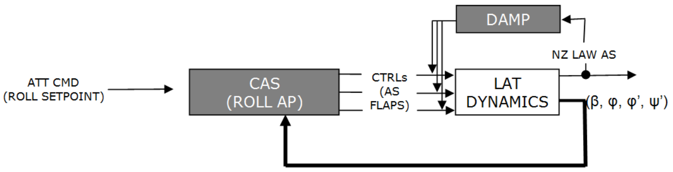


Fig. 4. Control augmentation system for H_2/H_∞ controller design. Where control surfaces are considered as anti-symmetrically driven wings ailerons.

Resulting damper (and also the overall CAS/damper combo) features robust stability for all mass cases, significant improvement regarding damping of structural vibrations for major part of mass cases (more than 5dB attenuation), and no-effect on vibrations damping for the remaining cases. These findings, and the overall performance of the designed controller and its respective parts, are visualized in the Fig. 3 and Fig. 4.

4.2 H2/H_∞ control results

Brief assessment of the controller performance is given in the text above (regarding robustness and performance). A set of selected characteristics is now given to document those findings.

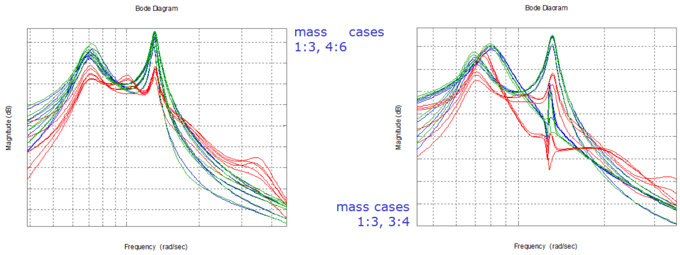


Fig. 5. Wing bending mode. Open loop (green), H_2 control (blue) and H_2/H_∞ control (red). All axis values are omitted from confidential reasons.

Note that very good performance is achieved for those cases that do not vary much in the frequency of the targeted modes (Fig. 5 left). However, even for the other cases (Fig. 5 right), some performance improvement is achieved, and robust closed loop stability is assured.

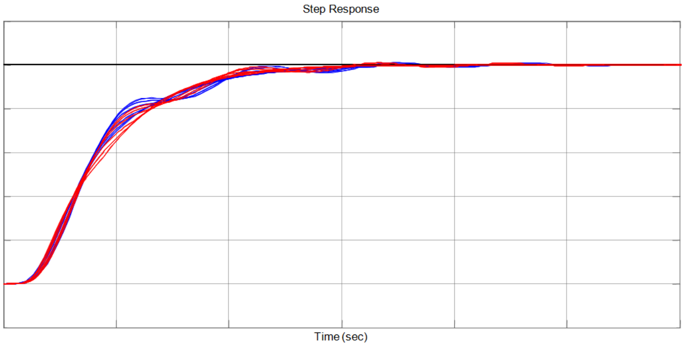


Fig. 6. Roll reference tracking. H2 control (blue) and H_2/H_∞ control (red).

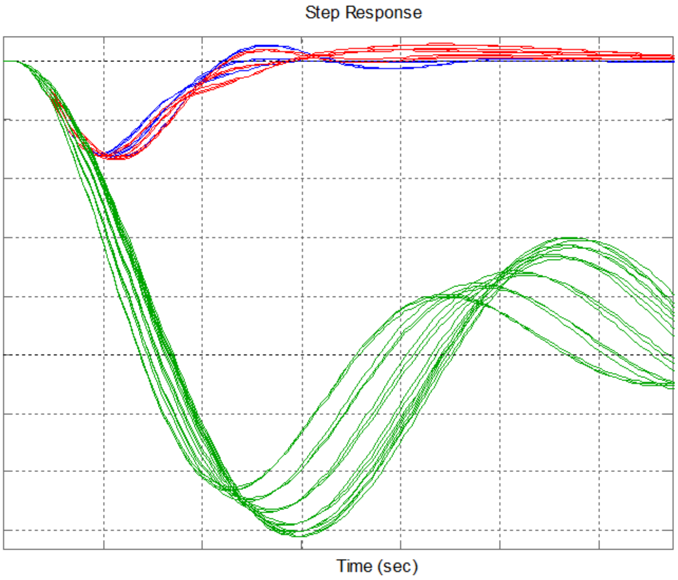


Fig. 7. Beta disturbance rejection. Open loop (green), H_2 control (blue) and H_2/H_∞ control (red).

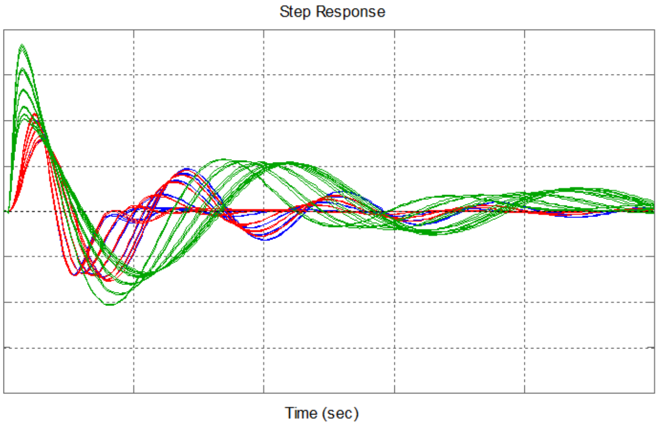


Fig. 8. Yaw rate damper. Open loop (green), H_2 control (blue) and H_2/H_∞ control (red).

Required response to a set point command is achieved. Note marginal improvement of the response when the damping system is connected (though it was not intended to influence the flight dynamics in fact). As stated above, the flight-dynamics part contains integrated yaw damper and beta compensator. Gain and phase margins for the complete designed controller have been evaluated. Robust closed loop stability for all mass cases is achieved. For simultaneous, independent, worst-case variations in the individual channels the gain margin ranges 1.9-3.7dB, phase margin 12-23 degrees, depending on the mass case (MATLAB/Robust Control Toolbox command *loopmargin*).

5. FIXED ORDER H_∞ OPTIMAL MIMO ROBUST CONTROLLER

An integrated H_∞ optimal approach was used to design Lateral Control Augmentation System (CAS) for NACRE airliner. Similarly as in previous section two different control goals were aimed, but this time in one integrated version. One part of control law is to provide autopilot functionality. The autopilot consists of Stability Augmentation System (Dutch roll damper) and CAS (roll and beta angle reference signal tracking). Other part of control law takes care of vibration and load attenuation.

5.1 Design method

In order to directly obtain a robust feedback controller of pre-specified order, the H_∞ Fixed-Order Optimization (HIFOO) toolbox is used, outlined in detail in (7). The HIFOO control design method searches for locally optimal solutions of a non-smooth optimization problem that is built to incorporate minimization objectives and constraints for multiple plants. First, the controller order is fixed at the outset, allowing for low-order controller design. Second, no Lyapunov or lifting variables are introduced to deal with the conflicting specifications. The resulting optimization problem is formulated on the controller coefficients only, resulting in a typically small-dimensional non-smooth nonconvex optimization problem that does not require the solution of large convex sub-problems, relieving the computational burden typical for Lyapunov LMI techniques. Because finding the global minimum of this optimization problem may be hard, an algorithm that searches only for local minimization is used. While no guarantee can be given on the result quality of this algorithm, in practice it is often possible to determine a satisfying controller efficiently.

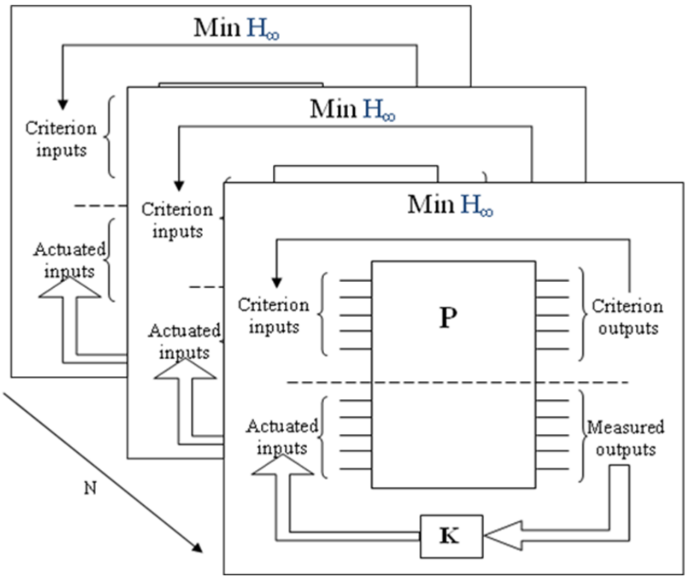


Fig. 9. H_∞ fixed order optimization setup.

The lateral integrated CAS was designed as a 2DoF architecture using fixed order optimization approach to keep control law order low. The resulting extremely low order (in this case 3^{rd} order control law was used) controller was built using HiFOO toolbox. Overall lateral CAS consist of Rigid Body autopilot (roll and beta tracker with Dutch roll damper) and structural modes control. The lateral CAS set up can be seen from Fig. 10. Two reference signals are used as inputs into feedforward part of controller (roll and beta set points). The beta reference signal is usually set to zero and then CAS provides coordinated turn functionality.

Control surfaces used by CAS are all ailerons (antisymmetricaly actuated FL1 - FL3), rudders (RU) and elevators (symmetrically actuated EL). Measured signals are lateral RB variables at CG (beta angle, roll angle, roll rate and yaw rate), for structural modes control we have selected lateral wing acceleration modal sensor in antisymmetrical setup. Resulting control law (autopilot and structural modes controller) provides robust stability as well as robust performance for all 18 cruse conditions cases (6 fuel and 3 passenger cases).

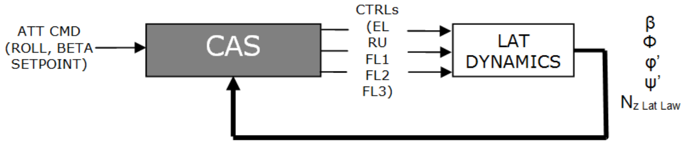


Fig. 10. Control augmentation system for HiFOO.

5.2 HiFoo control results

Improvement of damping of 1^{st} and 2^{nd} wing bending modes can be seen form Fig. 11. Simultaneously DC gain is preserve for all cases. Robust performance property can be seen form Bank angle reference signal tracking response plotted in Fig. 12 (left). Response for series of two steps is involved here and one can see that handling qualities are satisfied with suitable amount of overshot.

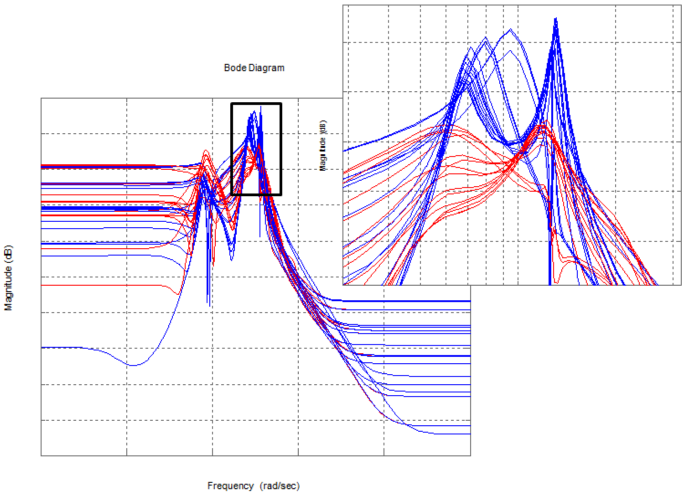


Fig. 11. Wing bending mode. Open loop (blue), closed loop (red).

Property of beta disturbance attenuation is investigated in Fig. 13 (left). One can seen complete vanishing of side wing influence in few second and without inducing of oscillation for major part of cases. Dutch roll mode damping is investigate in Fig. 13 (right).

Gain and phase margins for the complete designed controller have been evaluated. Robust closed loop stability for all mass cases is achieved. For simultaneous, independent, worst-case variations in the individual channels the gain margin ranges 0.8-2.6dB, phase margin 5-16 degrees, depending on the mass case (MATLAB/Robust Control Toolbox command *loopmargin*).

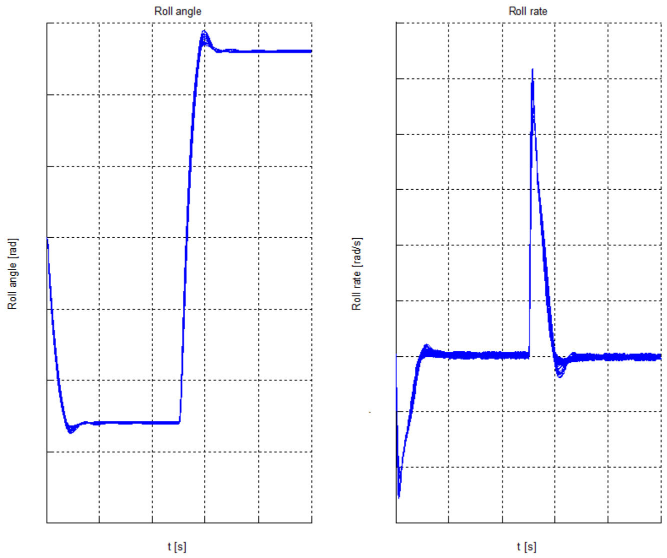


Fig. 12. Bank angle and Roll rate reference signal tracking.

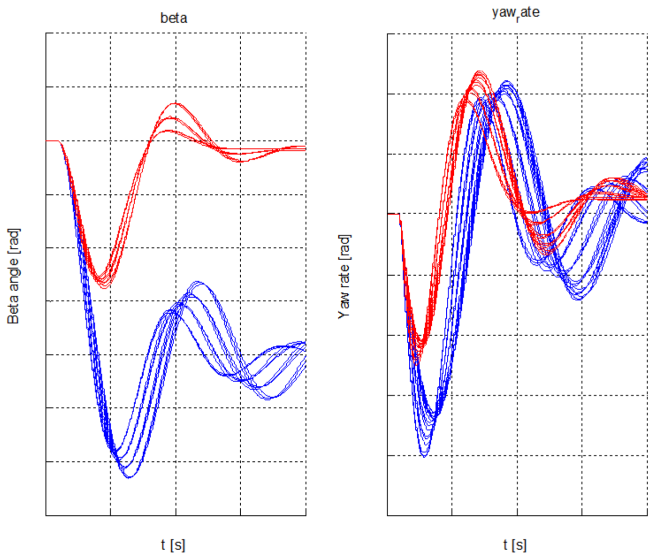


Fig. 13. Beta angle disturbance attenuation (left) and Yaw rate damping (right). Open loop (blue), closed loop (red)

6. CONCLUSIONS

Two efficient approaches to lateral control for the prospective BWB concept of large passenger aircraft are elaborated and assessed in this paper. First, a hierarchical approach is considered with separately designed control augmentation system (lateral autopilot with integrated beta-compensator and yaw damper) and the active damping system for structural vibrations on top of that. Main advantages of this approach are due to safety (the non-critical part - active damper - does not de-stabilize the plant if disengaged, e.g. due to a failure), easier process of tuning and certification (step-by-step), and the results look very good in fact. On the other hand, this approach is conservative by its nature and does not exploit fully the potential of active control as a true MIMO overall controller could do. Therefore, the second approach also presented in the paper is a fully integrated H_∞ optimal control law of low order designed by fixed order optimization. Performance of both control strategies is assessed, and the integrated design indeed features better closed loop characteristics in terms of robustness (more mass cases covered), rise times, or Dutch-roll damping.

ACKNOWLEDGEMENTS

The work of M. Hromčík was supported by the Ministry of Education of the Czech Republic (MSMT) under contract No. KONTAKT AMVIS ME10010. The work of T. Haniš was supported by the EC project ACFA 2020 - Active Control for Flexible 2020 Aircraft under No. 213321.

REFERENCES

[1] http://www.acfa2020.eu/

[2] D. Bates and I. Postlethwaite; Robust Multivariable Control of Aerospace Systems; DUP Science, Ios Pr Inc, 2002.

[3] A. Gelb, editor; Applied Optimal Estimation; The Analytic Sciences Corporation, MIT Press, 1974.

[4] F. Lewis; Optimal Estimation; John Wiley & Sons, 1986.

[5] S. Skogestad and I. Postlethwaite; Multivariable feedback control; John Wiley & Sons, 1996.

[6] B.L. Stevens and F. L. Lewis; Aircraft Control and Simulation; John Wiley & Sons, 2003.

[7] S. Gumussoy, D. Henrion, M. Millstone, M. Overton;Multiobjective Robust Control with HIFOO 2.0; IFAC Symposium on Robust Control Design, Haifa, Israel, June 16-18, 2009

[8] K. Zhou, J. C. Doyle, and K. Glover.; Robust and optimal control; Prentice Hall, 1996.

[9] A. Schirrer, C.Westermayer,M. Hemedi, andM. Kozek; A comprehensive robust control design and optimization methodology for complex flexible-structure systems. In Proc. of the 18th Mediterranean Conf. on Control and Automation, Marrakech, Morocco, 2010.

[10] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek; LQ-based design of the inner loop lateral control for a large flexible BWB-type aircraft. In 2010 IEEE Multi-Conf. on Systems and Control, Yokohama, Japan, 2010.

[11] A. Schirrer, C. Westermayer, M. Hemedi, and M. Kozek; Robust H1 control design parameter optimization via genetic algorithm for lateral control of a bwb type aircraft. In IFAC Workshop on Intell. Control Systems, Sinaia, Romania, 2010.

[12] C. Westermayer, A. Schirrer, M. Hemedi, and M. Kozek; Linear parameter-varying control of a large blended wing body flexible aircraft. In 18th IFAC Symposium on Automatic Control in Aerospace, Nara, Japan, 2010.

[13] C. Westermayer, A. Schirrer, M. Hemedi, M. Kozek, and A. Wildschek; Robust H_∞; flight and load control of a flexible aircraft using a 2DOF multi-objective design. In Proceedings of 2009 CACS International Automatic Control Conference, 2009.

Control of a Tubular Heat Exchanger

M. Bakošová * **M. Kačur** * **J. Oravec** *

* *Slovak University of Technology in Bratislava,*
Faculty of Chemical and Food Technology,
Institute of Information Engineering, Automation, and Mathematics,
Radlinského 9, 812 37 Bratislava, Slovakia
Tel: +421 2 59325364;
e-mail:{monika.bakosova, michal.kacur, juraj.oravec}@stuba.sk

Abstract: Using of a modified Smith predictor for compensation of measurable disturbances affecting a time-delay system is studied in this paper. The controlled system is a tubular heat exchanger, in which the kerosene is heated by hot water. The heat exchanger is a nonlinear system with time delay. The Smith predictor and the modified Smith predictor are used for control of the heat exchanger without and with disturbances. Obtained simulation results confirm that the modified Smith predictor with feed-forward compensation of measurable disturbances can improve the closed-loop control responses of the time delay systems with disturbances.

Keywords: time delay, disturbance, Smith predictor, modified Smith predictor, tubular heat exchanger

1. INTRODUCTION

Time delay is a typical phenomenon in real processes that is usually caused by information, mass or energy transport. It can be also caused by mass or energy accumulation in dynamic systems connected in series. Typical time-delay processes in chemical industry are tubular heat exchangers. There are several approaches to control heat exchangers as time-delay systems and the Smith predictor and its modifications belong to the approaches offering good results. The predictor based controllers are known as time delay compensators and they have been applied in many engineering fields, mainly in the process industry (Huzmezan et al. (2002), Normey-Rico et al. (1997)), but also in robotics (Normey-Rico and Camacho (1999)) and internet connection (Mascolo (2006)).

The Smith predictor and its modifications (Šulc and Vítečková (2004)) can be successfully used for control of processes with significant time delay, when the model of the controlled system and the model of the time delay are very well known. The modifications are used to improve the closed-loop control responses of time-delay integrating systems, time-delay unstable systems, time-delay systems with disturbances, etc. (Dostál et al. (2008)).

The paper presents using a modified Smith predictor for control of a co-current tubular heat exchanger, in which the kerosene is heated by hot water. Kerosene flows in the inner tube and water flows in the outer tube. The operation of the heat exchanger is affected by disturbances that are represented by changes of the kerosene inlet temperature. The objective is to heat the outlet temperature of the kerosene to the demanded value by the mass flow of heating water. The heat exchanger represents a non-linear system with time delay.

2. SMITH PREDICTOR

One of the most popular time delay compensating method is the Smith predictor. The structure of the Smith predictor is shown in Figure 1. This structure can be divided into two parts. The first part is the primary controller $G_r(s)$, which is usually the PID controller and the second part is the predictor structure. The predictor is composed of the plant model without time delay $G_m(s)$ and of the model of the time delay $e^{-D_m s}$. The complete process model is $P_m(s) = G_m(s)e^{-D_m s}$. The model $G_m(s)$ is used to compute an open loop prediction. The controller $G_r(s)$ can be tuned for the plant model without time delay, when there are no model errors or disturbances and the error between process output and model output is zero. For successful modelling, following three characteristics of the Smith predictor have to be analysed: $P(s) = P_m(s)$, $G(s) = G_m(s)$, $D = D_m$. The Smith predictor structure has for the nominal case (no modelling errors) these fundamental properties (Normey-Rico and Camacho (2008)):

- time delay compensation and prediction
- performance limitation of the Smith predictor

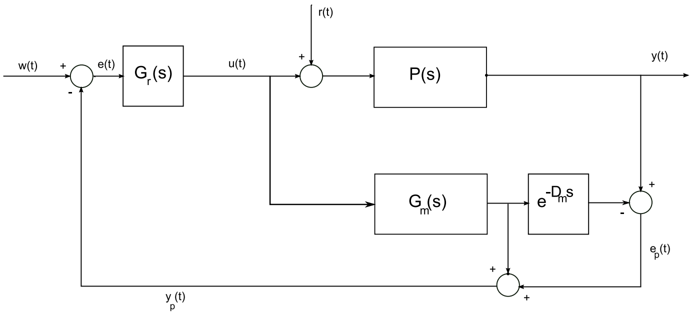


Fig. 1. Smith predictor

2.1 Property 1: Time delay compensation and prediction

It is easy to see in Figure 1, the error signal $e_p(t)$ is zero, if $r(t) = 0$ and $G(s)e^{-Ds} = G_m(s)e^{-D_m s}$. The characteristic equation is

$$1 + G_r(s)G_m(s) = 0 \tag{1}$$

Compare the equation (1) to the time-delay dependent one obtained in the PID case

$$1 + G_r(s)G_m(s)e^{-D_m s} = 0 \tag{2}$$

where the extra phase introduced by the time delay reduces the phase margin. The feedback signal $y_p(t)$ anticipates the system output for changes in the set point, although this is not the case for disturbances

$$y_p(t) = y(t + D_m) + P_m(s)\left[r(t) - r(t + D_m)\right] \tag{3}$$

For slow changes of the disturbance, it is a good prediction of $y(t + D_m)$. But if the disturbance changes rapidly then it cannot be eliminated from the feedback signal $y_p(t)$ (Normey-Rico and Camacho (2008)).

2.2 Property 2: Performance limitation for the Smith predictor

The structure of the Smith predictor divides the plant into two parts. The first is invertible $G_m(s)$ and the second is non-invertible $e^{-D_m s}$. Using this idea and considering that ideal controller with infinity gain could be applied, it follows (Fig. 2)

$$G_r'(s) = \frac{G_r(s)}{1 + G_r(s)G_m(s)} = (G_m(s))^{-1} \tag{4}$$

The ideal transfer function between the reference and the output is a simple delay. In real conditions the ideal controller cannot be applied. Even in the ideal case, if a disturbance is applied at $t = 0$, it is necessary to wait until $t = 2D_m$ to note the effect of the controller on the output (Normey-Rico and Camacho (2008)).

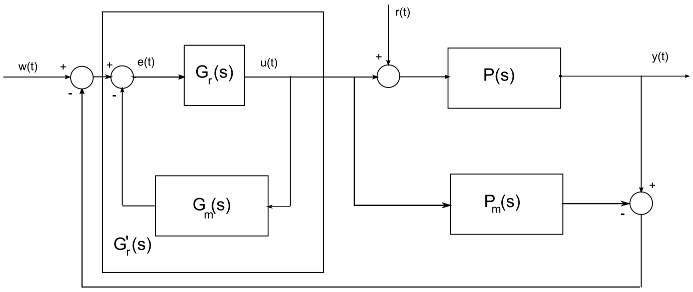


Fig. 2. Equivalent control structure of the Smith predictor

3. MODIFICATION OF THE SMITH PREDICTOR FOR DISTURBANCE COMPENSATION

A modification of the Smith predictor with feed-forward control loop can be used for improving the closed-loop control response when the controlled system is affected by measurable disturbances. When the disturbance is not measurable, this approach can be applied, but the disturbance has to be estimated (Normey-Rico and Camacho (2008)). Figure 3 shows a block diagram of the Smith predictor for disturbance compensation, where $P_{mr}(s)$ represents the model for $P_r(s)$. $P_r(s)$ represents the model of the disturbance dynamics (Normey-Rico and Camacho (2008)).

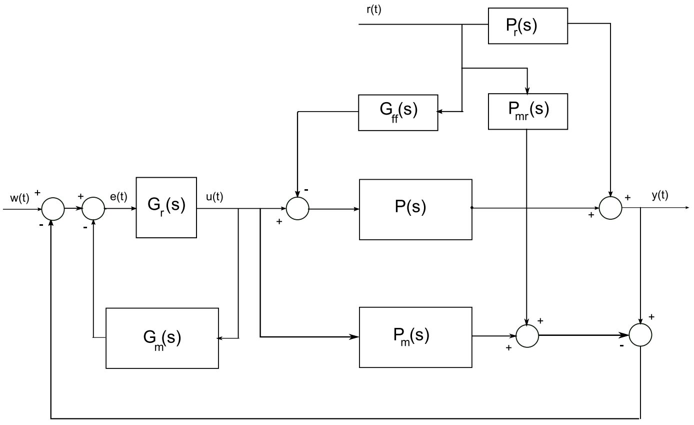


Fig. 3. Smith predictor modification for disturbance compensation

In the ideal case, when $P_{mr}(s) = P_r(s)$ and $P_m(s) = P(s)$, the transfer function $\frac{Y(s)}{R(s)}$ is in the form

$$\frac{Y(s)}{R(s)} = \left[P_r(s) - G_{ff}(s)P(s)\right] \tag{5}$$

The disturbance effect can be eliminated from the output of the process independently on the type of disturbance if exists such $G_{ff}(s)$ that

$$G_{ff} = \frac{P_r(s)}{P(s)} \tag{6}$$

Consider the plant and the load disturbance transfer functions $P(s)$ and $P_r(s)$ defined as $P(s) = G(s)e^{-Ds}$, $P_r(s) = G_{rr}(s)e^{-D_r s}$. Two situations can occur (Normey-Rico and Camacho (2008)):

- $D < D_r$

In this case, the controller is in the form

$$G_{ff} = \frac{G_r(s)}{G(s)}e^{-(D_r - D)s} \tag{7}$$

If $\frac{G_r(s)}{G(s)}$ can be computed the disturbance is eliminated from the output. Otherwise, a pseudo inverse of $G(s)$ can be computed $G_{ff}(s)P(s) = P(s)X(s)$. The final $\frac{Y(s)}{R(s)}$ is

$$\frac{Y(s)}{R(s)} = e^{-D_r s}G_r(s)[1 - X(s)], \tag{8}$$

where $1 - X(s)$ has zero static gain and the fastest achievable response (Normey-Rico and Camacho (2008)).

- $D > D_r$

In this case, it is not possible to compute the inverse of $e^{(D_r - D)s}$. The feed-forward controller is given by

$$G_{ff} = \frac{G_r(s)}{G(s)} \tag{9}$$

Table 1. Heat exchanger parameters and inputs

Variable / Unit	Value	Variable / Unit	Value
l / m	2	ρ_1 / kg m^{-3}	810
D_3 / m	0.05	ρ_2 / kg m^{-3}	8930
D_2 / m	0.028	ρ_3 / kg m^{-3}	1000
D_1 / m	0.025	C_{P1} / kJ kg^{-1}K^{-1}	2100
α_1^s / W m^{-2}K^{-1}	750	C_{P2} / kJ kg^{-1}K^{-1}	385
α_2^s/ W m^{-2}K^{-1}	1480	C_{P3} / kJ kg^{-1}K^{-1}	4186
$\dot{m}_1^s$ / kg s^{-1}	0.0556	ϑ_{1in}^s / °C	20
$\dot{m}_{3in}^s$ / kg s^{-1}	0.0417	ϑ_{3in}^s / °C	85

and the final transfer function is

$$\frac{Y(s)}{R(s)} = e^{-D_r(s)} G_r(s) \left[1 - X(s) e^{-(D-D_r)s}\right] \tag{10}$$

Note that even in this case the solution is better than the one obtained when the feed forward is not used. The advantage of this solution is less important when $D_r \to 0$. The previous structure cannot be used when disturbance is not measurable. Using an estimation of disturbance $r(t)$, the idea can be used to improve the controller. One advantage of this approach is that the controller can be easily tuned to reject other types of disturbances and not only step ones (Normey-Rico and Camacho (2008)).

4. MODEL OF THE TUBULAR HEAT EXCHANGER

The controlled process is a co-current tubular heat exchanger, in which kerosene is heated by hot water. Kerosene flows in the inner copper tube and water flows in the outer copper tube. The operation of the heat exchanger is affected by the disturbance that is represented by changes of the kerosene inlet temperature. The objective is to heat the outlet temperature of the kerosene to the demanded value by the mass flow of heating water. The heat exchanger represents a non-linear system with variable time delay, where the controlled output is the outlet kerosene temperature and the control input is the mass flow-rate of heating water.

Technological parameters and steady-state inputs of the heat exchanger are listed in the Table 1, where l is the length of the heat exchanger, D is the tube diameter, ρ is the density, α is the heat transfer coefficient, C_P is the specific heat capacity and $\dot{m}$ is the mass flow rate. The subscripts have following meaning: 1–kerosene or from the copper tube to kerosene or the inner diameter of the inner tube, 2–copper or from water to the copper tube or the outer diameter of the inner tube, 3–water or the inner diameter of the outer tube and in – the inlet. The superscript s represents the steady-state.

The mathematical model of the tubular heat exchanger is represented by three nonlinear partial differential equations in the form

$$T_1 \frac{\partial \vartheta_1(z,t)}{\partial t} + T_1 w_1 \frac{\partial \vartheta_1(z,t)}{\partial z} = -\vartheta_1(z,t) + \vartheta_2(z,t) \tag{11}$$

$$T_2 \frac{\partial \vartheta_2(z,t)}{\partial t} = Z_1 \vartheta_1(z,t) - \vartheta_2(z,t) + Z_2 \vartheta_3(z,t) \tag{12}$$

$$T_3 \frac{\partial \vartheta_3(z,t)}{\partial t} + T_3 w_3(z,t) \frac{\partial \vartheta_3(z,t)}{\partial z} = -\vartheta_3(z,t) + \vartheta_2(z,t) \tag{13}$$

where

$$T_1 = \frac{D_1 \rho_1 C_{P1}}{4\alpha_1}, \quad T_2 = \frac{(D_2^2 - D_1^2)\rho_2 C_{P2}}{4(D_1\alpha_1 + D_2\alpha_2)},$$

$$T_3 = \frac{(D_3^2 - D_2^2)\rho_3 C_{P3}}{4 D_2 \alpha_2}$$

$$w_1 = \frac{q_1}{\pi D_1^2}, \quad w_3(z,t) = \frac{q_3(z,t)}{\pi(D_3^2 - D_2^2)}$$

$$q_3(z,t) = \frac{\dot{m}_3(z,t)}{\rho_3}$$

$$Z_1 = \frac{D_1\alpha_1}{D_1\alpha_1 + D_2\alpha_2}, \quad Z_1 = \frac{D_2\alpha_2}{D_1\alpha_1 + D_2\alpha_2}$$

For simulations purposes, the heat exchanger was split in ten sections, each of them represented by three ordinary nonlinear differential equations with delayed inputs. The model was generated using MATLAB–Simulink environment.

For control purposes, the properties of the heat exchanger have been examined by simulation experiments. The model of the heat exchanger was identified using the Strejc method (Mikleš and Fikar (2007)) in the form of the transfer function

$$P_m(s) = \frac{K}{(Ts+1)^n} e^{-Ds} \tag{14}$$

where n is the order of the system, K is the gain, T is the time constant and D is the time delay.

For the identification, following step changes of the inlet mass flow-rate of heating water were generated at the time $t = 0$ s: ±10%, ±20%, ±30%. Step responses of the outlet kerosene temperature on the generated inlet step changes are shown in Figure 4. According to these step changes, the heat exchanger is a time-delay nonlinear system with asymmetric dynamics.

The heat exchanger was identified in the form of the 3rd order plus time delay system (Table 2). For various step responses, we obtained intervals for values of the gain K, the time constant T and the time delay D.

Table 2. Identification of the process dynamics

	K_{min}	K_{max}	T_{min}	T_{max}	D_{min}	D_{max}
$n = 3$	0.055	0.071	11.382	19.241	14.422	22.844

It is supposed further that the dynamics of the heat exchanger is affected by disturbances. The disturbances are caused by changes of the kerosene inlet temperature. The model of the disturbance dynamics was identified using the Strejc method. The generated step changes of the inlet kerosene temperature were ±2°C. The step responses of outlet kerosene temperature are depicted in the Figure 5. The values of the identified parameters are summarized in Table 3, where K_r represents the gain, T_r is the time constant and D_r is the time delay of the model of the disturbance dynamics.

Table 3. Identified parameters of the disturbance dynamics

	K_r	T_r	D_r
$n_r = 2$	0.6007	4.7566	20.8980

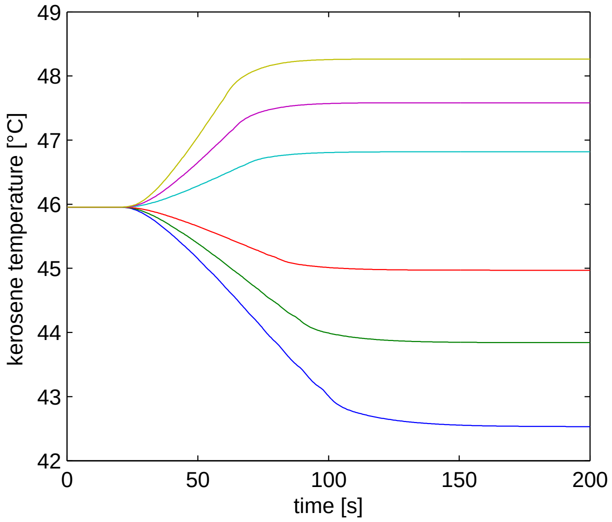


Fig. 4. Step responses of the outlet kerosene temperature on the step changes of the control input, where input change +10% is represented by cyan line, −10% is represented by red line, +20% is represented by magenta line, −20% is represented by green line, +30% is represented by yellow line, −30% is represented by blue line

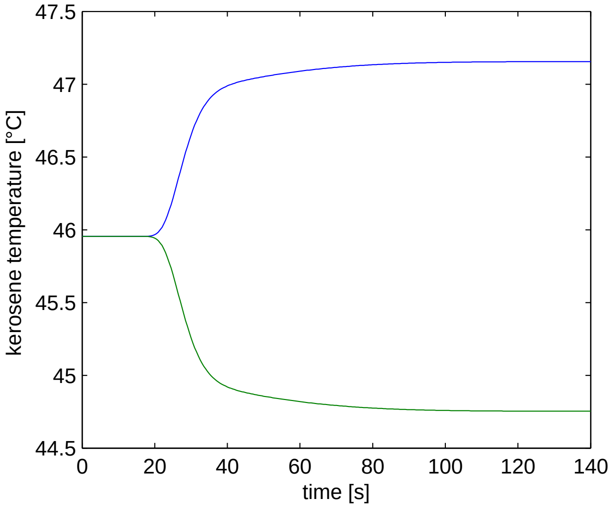


Fig. 5. Step responses of the outlet kerosene temperature on the step changes of the disturbance, where disturbance change +2% is represented by blue line, −2% is represented by green line

5. CONTROL OF THE TUBULAR HEAT EXCHANGER

For the third order model of the heat exchanger, four PI controllers were designed ($G_{R1} - G_{R4}$). The transfer function of the PI controller is in the form

$$G_R(s) = Z_R + \frac{Z_R}{T_I s} \tag{15}$$

where Z_R is the gain and T_I is the reset time of the controller (Bakošová et al. (2003)).

The controllers G_{R1}–G_{R3} were tuned using the experimental methods. The controller G_{R1} was designed for the model described by maximal values of identified parameters, the controller G_{R2} was designed for the model described by the minimal values of identified parameters and the controller G_{R3} was designed for the model described by the mean (nominal) values of parameters (Table 2). Because the heat exchanger can be represented also as a system with interval parametric uncertainty, a robust PI controller G_{R4} was tuned using the method described in Závacká et al. (2007). The parameters of designed controllers are listed in Table 4.

Table 4. Parameters of set-point tracking performances using the Smith predictor

controller	model parameters	Z_R	T_I	IAE
G_{R1}	*maximal*	8.44	49.14	1794
G_{R2}	*minimal*	11.78	83.07	2035
G_{R3}	*nominal*	9.98	25.52	4737
G_{R4}	*interval*	5.00	25.00	2150

Control of the heat exchanger without disturbance using the Smith predictor was simulated at first and all designed controllers were used in the predictor structure. The step change of the set-point was done at time $t = 1500$ s from 50°C to 40°C. The closed-loop control responses obtained using four designed controllers are shown in Figure 6, where the closed-loop control using G_{R1} is represented by the solid magenta line, using G_{R2} is represented by the cyan dash-dot line, using G_{R3} is represented by the red dotted line, using G_{R4} is represented by the green dashed line. The quality of the closed-loop control was evaluated using IAE (Mikleš and Fikar (2007)) quality criteria (16)

$$\text{IAE} = \int_0^\infty |e(t)|dt \tag{16}$$

Obtained values of IAE are enumerated in Table 4. The best value of the IAE was reached using controller G_{R1}.

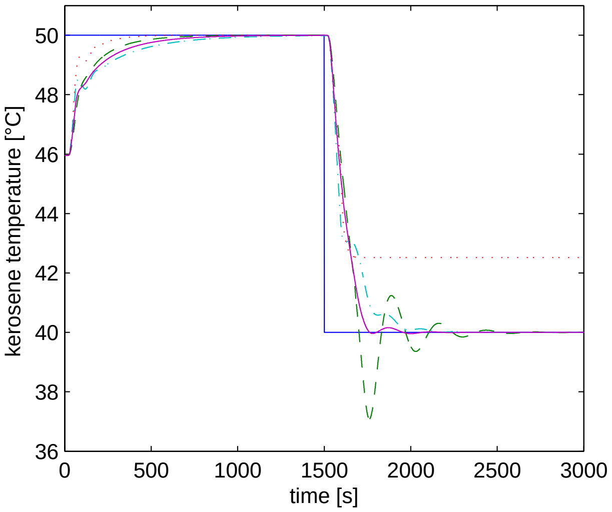


Fig. 6. Set-point tracking assured using G_{R1} (solid magenta line), G_{R2} (cyan dash-dot line), G_{R3} represented by the (red dotted line) and G_{R4} (green dashed line)

Then the control of the heat exchanger affected by disturbances was analyzed. The disturbance is represented by

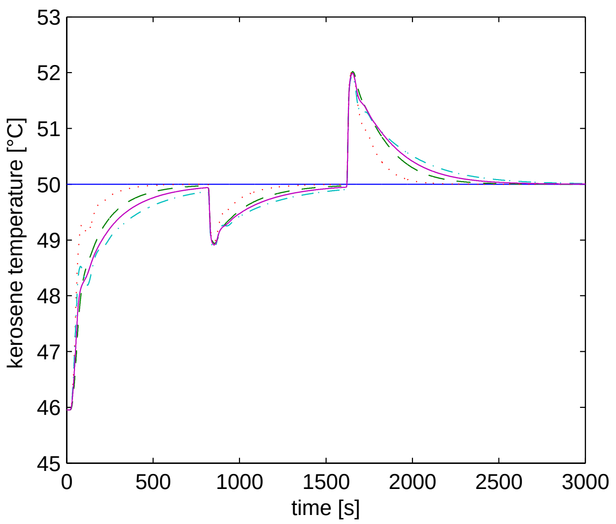


Fig. 7. Disturbance rejection without disturbance compensation assured using G_{R1} (solid magenta line), G_{R2} (cyan dash-dot line), G_{R3} represented by the (red dotted line) and G_{R4} (green dashed line)

the step change of the kerosene inlet temperature. This temperature decreased in 2°C at the time t = 800s and then the inlet kerosene temperature increased in 2°C at the time t = 1600s. Table 5 contains the controllers and the associated calculated values of IAE. The minimal value has been reached using G_{R3} controller. Figure 7 shows the closed-loop control responses obtained using the Smith predictor without disturbance compensation, where the closed-loop control using G_{R1} is represented by the solid magenta line, using G_{R2} is represented by the cyan dash-dot line, using G_{R3} is represented by the red dotted line, using G_{R4} is represented by the green dashed line.

Table 5. Parameters of disturbance rejection performances using the Smith predictor

controller	model parameters	Z_R	T_I	IAE
G_{R1}	*maximal*	8.44	49.14	1371
G_{R2}	*minimal*	11.78	83.07	1533
G_{R3}	*nominal*	9.98	25.52	740
G_{R4}	*interval*	5.00	25.00	1210

Then the modified Smith predictor with feed-forward disturbance compensation was applied. In Table 6, we can see the used controllers and the associated calculated values of IAE. The minimal value of IAE was reached also using G_{R3} controller. As can be seen in Table 6, the control performances generated by the controllers G_{R1} and G_{R3} lead to the higher values of IAE in comparison to the Smith predictor without disturbance compensation (Table 5). Figure 8 shows the closed-loop control response obtained using the modified Smith predictor with disturbance compensation, where the closed-loop control using G_{R1} is represented by the solid magenta line, using G_{R2} is represented by the cyan dash-dot line, using G_{R3} is represented by the red dotted line, using G_{R4} is represented by the green dashed line.

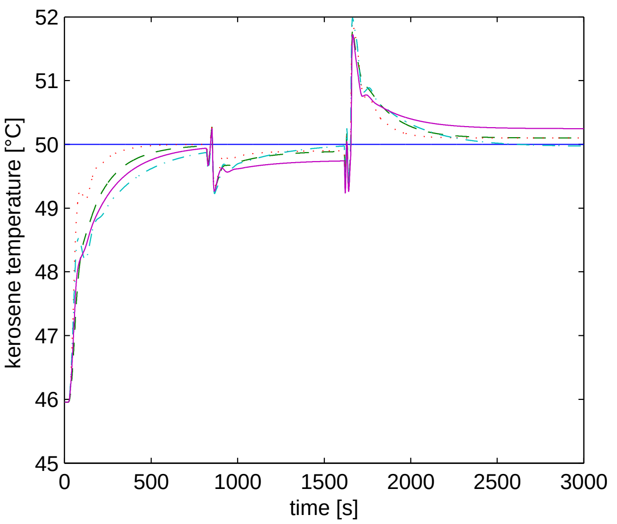


Fig. 8. Disturbance rejection with disturbance compensation assured using G_{R1} (solid magenta line), G_{R2} (cyan dash-dot line), G_{R3} represented by the (red dotted line) and G_{R4} (green dashed line)

6. CONCLUSION

The possibility to use the Smith predictor and the modified Smith predictor with feed-forward disturbance compensation for control of a time-delay system was studied in this paper. The controlled system was a tubular heat exchanger, which was a nonlinear time delay system. As the process was identified as a system with interval parametric uncertainty, the four PI controllers were designed for the Smith predictor and modified Smith predictor control structure. The three controllers were designed using experimental methods. They were tuned using the maximal, minimal and nominal process model parameters. The fourth PI controller was designed using robust control approach. Because of complicated dynamics of the controlled process, obtained simulation results are difficult to compare. But it can be stated, that using the robust controller in both control structures never led to the worst control response. Using the robust controller in the modified Smith predictor lead to the better disturbance compensation in comparison to the Smith predictor without disturbance compensation. Using the controller designed for the nominal values of process model parameters gave the best results in the task of disturbance rejection, but this controller led to the worst result in the task of setpoint tracking.

In the next work the heat exchanger with counter-current of cooling medium will be studied. The obtained results will be compared to the results obtained using the heat exchanger with co-current of cooling medium. The studied

Table 6. Parameters of disturbance rejection performances using modified the Smith predictor

controller	model parameters	Z_R	T_I	IAE
G_{R1}	*maximal*	8.44	49.14	1422
G_{R2}	*minimal*	11.78	83.07	1198
G_{R3}	*nominal*	9.98	25.52	790
G_{R4}	*interval*	5.00	25.00	1116

control methods will be applied to the control of a real model of the heat exchanger.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency VEGA of the Slovak Republic under the grants 1/0537/10, VV-0029-07.

REFERENCES

Bakošová, M., Ľ. Čirka, and Fikar, M. (2003). *Automatic control fundaments (in Slovak).* STU Publishing House, Bratislava.

Dostál, P., Gazdoš, F., and Bobál, V. (2008). Design of controllers for time delay systems part II: Integrating and unstable systems. *Journal of Electrical Engineering*, 59(1), 3–8.

Huzmezan, M., Gough, W., Dumont, G., and Kovac, S. (2002). Time delay integrating systems: A chalenge for process control industries. A practical solutions. *Control Engineering Practise*, 10(10), 1153–1161.

Mascolo, S. (2006). Modeling the internet cognestion control using a Smith controller with input shaping. *Control Engineering Practise*, 14(4), 425–435.

Mikleš, J. and Fikar, M. (2007). *Process Modelling, Identification, and Control.* Springer Verlag, Berlin Heidelberg.

Normey-Rico, J., Bordons, C., and Camacho, E. (1997). Improving the robustness of dead-time compensating PI controllers. *Control Engineering Practise*, 5(6), 801–810.

Normey-Rico, J. and Camacho, E. (1999). A Smith predictor based generalized predictive controller for mobile robot path tracking. *Control Engineering Practise*, 7, 729–740.

Normey-Rico, J. and Camacho, E. (2008). Dead-time compensators: A survey. *Control Engineering Practise*, 1(16), 407–428.

Šulc, B. and Vítečková, M. (2004). *Theory and praxis of control system design (in Czech).* ČVUT Publishing House, Prague.

Závacká, J., Bakošová, M., and Vaneková, K. (2007). Control of systems with parametric uncertainties using a robust PI controller. *AT&P Journal*, (Plus2), 84–87.

Robust Control of a Hydraulic System with Unstructured Uncertainties

M. Karšaiová M. Bakošová A. Vasičkaninová

*Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology,
Slovak University of Technology, Radlinského 9, 812 37 Bratislava, Slovakia
Tel. + 421 2 59 325 362; e-mail: {maria.karsaiova, monika.bakosova, anna.vasickaninova}@stuba.sk*

Abstract: The paper presents simulation results obtained by robust control of a system of three serially connected tanks. The method used for robust controller design is based on the small gain theorem. The robust PID controller is designed that assures the stability of the closed-loop control system for a certain range of unstructured uncertainties.

Key words: unstructured uncertainty, robust controller, robust stability, small gain theorem

1. INTRODUCTION

Uncertainty arises when some aspect of the system model is not completely known at the time of analysis and design. The typical example of a structured uncertainty is the value of a parameter which may vary according to operating conditions. The unstructured uncertainty can be caused by simplified modelling, when it is used to avoid very detailed and complex models. The other reasons for unstructured uncertainties are process non-linearity, changes of operating conditions and external disturbances. Dynamic systems with unstructured uncertainties are widely used to model physical systems.

The small gain theorem (Green and Limebeer, 1994) is a tool for robust controller design for systems with unstructured uncertainty (Karafyllis and Zhong-Ping, 2007). The small gain theorem states that stable systems can be connected to form a stable closed-loop if the loop gain product is less than unity. It is the basis for the general robust stability results.

The paper describes the robust PID controller design for three serially connect tanks. The process is modelled as a system with unstructured additive uncertainty and the robust controller design is based on the small gain theorem. The designed robust controller is tested by simulations.

2. ROBUST STABILITY

Suppose that the transfer function of an uncertain continuous-time system with additive unstructured uncertainty has the form

$$\begin{aligned} G(s) &= G_0(s) + G_{\Delta A}(s) \\ G_{\Delta A}(s) &= W_A(s)\Delta_A(s) \end{aligned} \tag{1}$$

where $G_0(s)$ is the nominal model, $W_A(s)$ is the weight function and $\Delta_A(s)$ is a category of uncertainties that satisfies the condition $\left|\Delta_A(j\omega)\right| \le 1$ for $\forall\omega$.

The task is to find a robust controller for control of the system (1). The design method is based on the small gain theorem (Green and Limebeer, 1994, Veselý and Harsanyi, 2007) and uses the fact that if a feedback loop consists of stable systems and the loop-gain product is less than unity, then the feedback loop is internally stable. The other basis for the design is a fixed point theorem known as the contraction mapping theorem (Khalil, 1996).

According to the small gain theorem, following conditions have to be satisfied: the controller with the transfer function $G_R(s)$ stabilizes the nominal model and for the open-loop transfer function $L(s)$, the condition given in (2) also holds.

$$\begin{aligned} &L(s) = G(s)G_R(s) \\ &\left|L(j\omega)\right| < 1 \end{aligned} \tag{2}$$

The family of the controlled system transfer functions $G(s)$ creates a set, in which $G_0(s)$ is the transfer function of the nominal system and $G_k(s)$ is a transfer function from the set $G(s)$, which differs from $G_0(s)$. Then, the value $l_A(\omega)$ can be calculated as the maximal value of modules as it is shown in (3)

$$\begin{aligned} &l_A(\omega) = max\left|G_k(j\omega) - G_0(j\omega)\right| \\ &\omega \in (0,\infty), \quad k = 1,2..... \end{aligned} \tag{3}$$

The characteristic equation of the closed loop with uncertain controlled system is

$$1 + G_R(s)G(s) = 0 \tag{4}$$

and after the substitution (1) into (4), we obtain

$$\left[1+G_R(s)G_0(s)\right]\left[1+V_0(s)\frac{G_{\Delta A}(s)}{G_0(s)}\right]=0 \qquad (5)$$

where $V_0(s)$ is the closed-loop transfer function with the nominal model and has the form

$$V_0(s)=\frac{G_R(s)G_0(s)}{1+G_R(s)G_0(s)} \qquad (6)$$

The closed loop must be stable. The small gain theorem requires satisfying also the second condition. It follows from (5) that for the second term in (5) the following condition holds

$$\left[1+V_0(s)\frac{G_{\Delta A}(s)}{G_0(s)}\right]=0 \qquad (7)$$

Then after the substitution $s = j\omega$ we obtain

$$\left|V_0(j\omega)\frac{G_{\Delta A}(j\omega)}{G_0(j\omega)}\right|<1 \qquad (8)$$
$$\forall\omega\in(0,\infty)$$

The conditions $\left|\Delta_A(j\omega)\right|=1$ and $\left|W_A(j\omega)\right|=l_A(\omega)$ represent the worst cases and so, it is possible to rewritten (8) to the form

$$\left|V_0(j\omega)\right|<\frac{\left|G_0(j\omega)\right|}{l_A(\omega)} \qquad (9)$$

Robust controller design is then based on finding parameters of the transfer function $V_0(s)$, the choice of the structure of the robust controller and calculation of the controller parameters.

3. ROBUST PID CONTROLLER DESIGN

A robust PID controller was designed for the process represented by three serially connected tanks. The controlled variable is the liquid level h_3 in the 3rd tank and the manipulated variable is the flow rate of the inlet stream of water q. The inputs, outputs and parameters of the nominal model in a steady state are represented by following values: inlet flow rate q^s =1m^3 min^{-1}, steady-state value of level h_3^s = 0.444 m, valve constants k_{11} = 1 m$^{2.5}$ min^{-1}, k_{22} = 1.5 m$^{2.5}$ min^{-1}, k_{33} = 1.5 m$^{2.5}$ min^{-1} and cross-section areas of tanks F_1 = 0.5 m^2, F_2 = 0.1 m^2, F_3 = 0.1 m^2. Unstructured uncertainties result from simplification of the mathematical model using linearization and changes of the valve constant $k_{33.}$ The family of the transfer functions in three operating points is following

$$G_0(s)=\frac{G_{0n}(s)}{G_{0d}(s)}=\frac{0.9}{0.008s^3+0.186s^2+1.18s+1} \qquad (10)$$

$$G_1(s)=\frac{G_{1n}(s)}{G_{1d}(s)}=\frac{1.18}{0.01s^3+0.21s^2+1.2s+1} \qquad (11)$$

$$G_2(s)=\frac{G_{2n}(s)}{G_{2d}(s)}=\frac{0.69}{0.006^3+0.16s^2+1.16s+1} \qquad (12)$$

where $G_0(s)$ is the nominal model, $G_1(s)$ is the model obtained with k_{33} = 1.3 m$^{2.5}$ min^{-1} and $G_2(s)$ is the model obtained for k_{33}= 1.7 m$^{2.5}$ min^{-1}. Figure 1 shows the function $l_A(\omega)$ which was determined using (3).

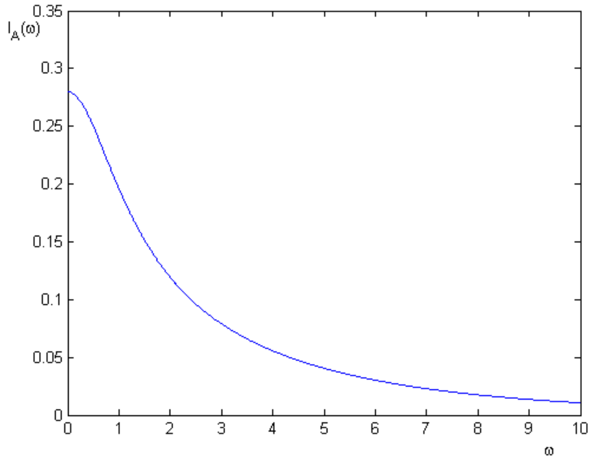


Fig. 1. Dependence of maximal values of modules on the frequency ω

The structure of robust PID controller was selected in the form:

$$G_R(s)=\frac{G_{Rn}(s)}{G_{Rd}(s)}=\frac{1}{K}\frac{C(s)}{s}=\frac{1}{K}\frac{c_1s^2+c_0s+c_{-1}}{s} \qquad (13)$$

The transfer function $V_0(s)$ is in the form

$$V_0(s)=\frac{G_{0n}(s)}{G_{0n}(s)+D_{0d}(s)Ks} \qquad (14)$$

Parameter K is an optional parameter and the function $V_0(s)$ has to satisfy (9). The polynomial $D_{0d}(s)$ is optional, too, and the following equation has to be satisfied

$$G_{0d}(s) = D_{0d}(s)C(s) \tag{15}$$

Parameters d_1, c_1, c_0, c_{-1} are calculated from the following equation

$$\begin{gathered} 0.008s^3 + 0.18s^2 + 1.18s + 1 = \\ (d_1 s + 1)\left(c_1 s^2 + c_0 s + c_{-1}\right) \end{gathered} \tag{16}$$

and their values are: $d_1 = 1.1765$, $c_1 = 0.0068$, $c_0 = 0.0035$, $c_{-1} = 1$. The transfer function $V_0(s)$ has the form

$$V_0(s) = \frac{1}{d_1 K s^2 + K s + 1} \tag{17}$$

and it is affected by the choice of the parameters K. Figure 2 illustrates the dependence of $\frac{|G_0(j\omega)|}{l_A(\omega)}$ and the dependence of $|V_0(j\omega)|$ on ω calculated for various values of K (blue lines). It is clear from Figure 2 that the boundary value of K is $K = 0.11$.

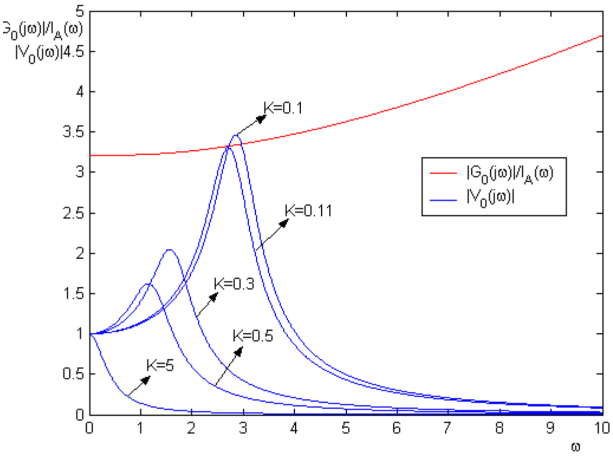


Fig. 2. Amplitude characteristics of V_0 for various K

The system of three serially connected tanks was controlled using three robust PID controllers designed by the small gain theorem for various feasible values of the parameter K. The transfer functions of found controllers are:

$$G_R(s) = 0.0117 + \frac{3.333}{s} + 0.0227s, \quad K = 0.3 \tag{18}$$

$$G_R(s) = 0.007 + \frac{2}{s} + 0.136s, \quad K = 0.5 \tag{19}$$

$$G_R(s) = 0.0007 + \frac{0.2}{s} + 0.0014s, \quad K = 5 \tag{20}$$

Designed controllers were tested by simulation experiments for the nominal model with $k_{33} = 1.5$ (the index 0), the model with $k_{33} = 1.3$ (the index 1) and the model with $k_{33} = 1.7$ (the index 2). The set point was $w = 0.4$ and it changed at the time 40 min to $w = 0.6$. The simulation results obtained using the PID robust controllers (18) – (20) are presented in Figs. 3 – 5.

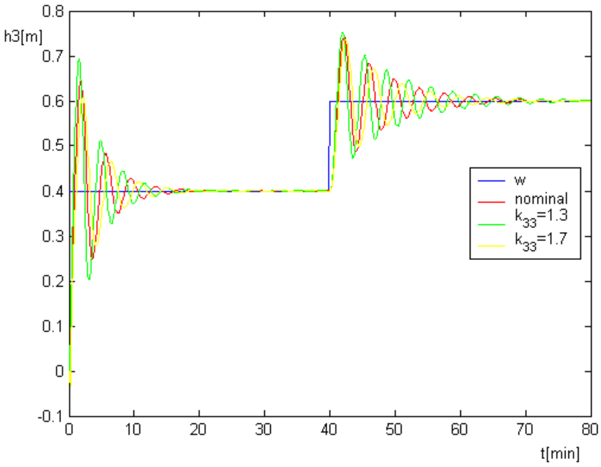


Fig. 3. Control responses of the system of three serially connected tanks with the controller (18)

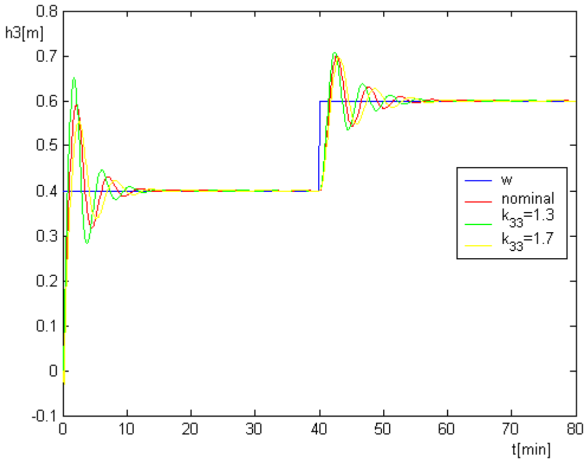


Fig. 4. Control responses of the system of three serially connected tanks with the controller (19)

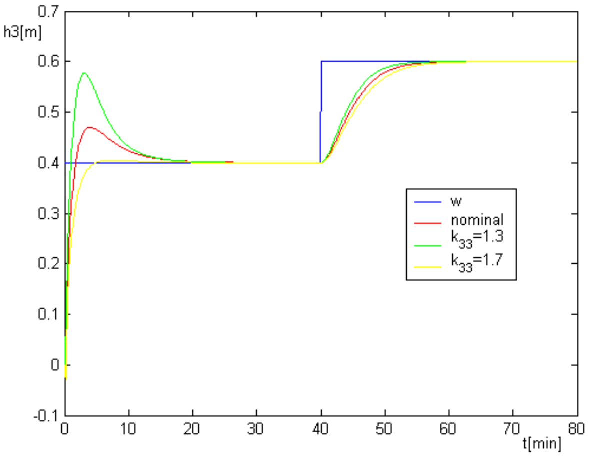


Fig. 5. Control responses of the system of three serially connected tanks with the controller (20)

The control responses obtained using three robust controllers were compared by evaluation of the IAE performance indexes. Their values are summarizes in the Table 1.

Table 1 Comparison of robust controllers using IAE performance indexes

	IAE$_0$	IAE$_1$	IAE$_2$
K=0.3	1.817	2.0541	1.7609
K=0.5	1.26	1.308	1.294
K=5	1.8218	2.1527	1.6828

One PID controller was designed for K higher then 0.11 to show that such choice leads to unstable control responses. The choice was $K = 0.1$ and the inequality (9) is not satisfied in this case. The transfer function of the found controller is

$$G_R(s) = 0.035 + \frac{10}{s} + 0.068s, \quad K = 0.1 \tag{21}$$

The simulation results obtained using the PID controller (21) are presented in Fig. 6.

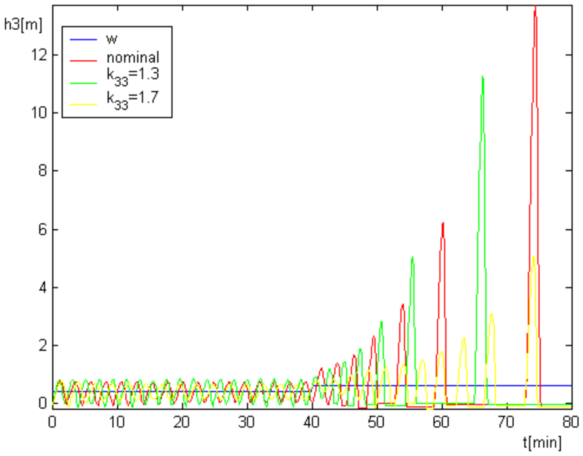


Fig. 6. Control responses of the system of three serially connected tanks with the controller (20)

4. CONCLUSION

Obtained simulation results confirmed, that it is possible to assure good control responses of controlled processes with unstructured uncertainties using robust PID controllers. The optional parameter K used in the controller synthesis depends on the controlled system and on the amplitude of unstructured uncertainties. The stability and quality of the control response depends on the value of optional parameter K and it is important to find its boundary value.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0537/10, 1/0095/09.

REFERENCES

Green, M. and D.N.J. Limebeer (1994). Linear robust control. Prentice Hall, New Jersey.

Karafyllis, I. and J. Zhong-Ping (2007). A small-gain theorem for a wide class of feedback systems with control application. *SIAM Journal on Control and Optimization* **46**, 1483-1517.

Khalil, H.K. (1996). Nonlinear systems. Prentice Hall, New Jersey.

Veselý, V. and Harsányi, L. (2007). Robust control fo dynamic systems (in Slovak). STU, Bratislava.

Virtual laboratory of process control

M. Kalúz *, Ľ. Čirka *, M. Fikar *

* *Slovak University of Technology in Bratislava*
Faculty of Chemical and Food technology
e-mail: {martin.kaluz, lubos.cirka, miroslav.fikar}@stuba.sk
url: http://www.kirp.chtf.stuba.sk

Abstract: This paper describes new PID control features, which have been implemented in our virtual Flash laboratory that was originally designed for simulations of technological plants. We discuss methods of discrete PID implementation and show new functions of the virtual laboratory.

Keywords: Virtual Laboratory, Process Control, Simulation

1. INTRODUCTION

Nowadays, in the field of automation and process control it is essential to keep abreast of information technologies. Virtual laboratories became a very popular way how to improve education opportunities in fields of automation, process control, physics, mathematics, and others. These laboratories are usually publicly accessible through Internet and thus can be used by a wider group of students and other interested people. The most commonly used software technologies to create and run online computer simulations are Java, Flash, PHP, ASP, JavaScript, and development environments like Easy Java Simulations (EJS), Virtual Reality Modeling Language (VRML), LabVIEW, and MATLAB connected technologies.

At our department we try to give students practical experience with measurements using real technological devices. But it is not possible to provide them these opportunities anytime, because our capabilities depend on disponibility of real plants at our department. In past, we decided to develop an Internet virtual laboratory to handle this issue.

The virtual laboratories from the different fields of science can be found all over the web. The Virtual Laboratory of Evolutionary Computing, developed by students from FEI STU in Bratislava, provides simulations in field of technical computing, automation, and process control. Most of their simulations are based on MATLAB software with graphical user interfaces built on VRML (Virtual Reality Modeling Language). This virtual laboratory is freely accessible through Internet and can be found on web site `http://ural.elf.stuba.sk/vrlab/`.

Malki and Matarrita (2002) created the virtual laboratory for process control based on LabVIEW environment. They used virtual model software with PID control of second-order plants.

Another interesting project is Virtual engineering/science course, which is a virtual laboratory created by developers from the Department of Chemical Engineering at John Hopkins University, Baltimore, USA. Virtual laboratory is based on Java technology (Java Applets) and includes models such as a robotic arm, heat conduction in materials, spatial distribution of sound, diffusion process, and some statistical and computing programs. Models are available at Web page `http://www.jhu.edu/~virtlab/virtlab.html`.

Our original virtual Flash laboratory, described in Kalúz (2010) and Čirka et al. (2010), was created using software programming platform Adobe Flash, and it provides virtual simulations of technological plants like liquid storage tank system, tube heat exchanger, and continuous stirred-tank reactor. We think that virtual laboratory including only simulations of system dynamics is not sufficient for actual education needs, so we have decided to improve it with features of process control. Our extensions for simulation applications in virtual laboratory contain feedback closed-loop control with PID controllers. Students and other users can access the virtual laboratory through e-learning system Moodle, located on web page of our department.

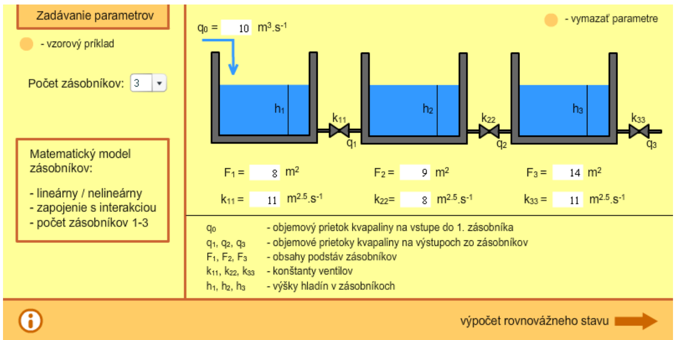
Recently, we have also created a virtual laboratory (Kalúz et al., 2010) based on Java Server Pages technology, including computing applications with MATLAB functions and simulations of storage tank system and heat exchangers.

2. OVERVIEW OF THE ORIGINAL APPLICATIONS

Our original applications in virtual laboratory course are simulations, based on computation of mathematical models of technological plants. The applications were created as Flash SWF files, published in regular Web page of Moodle course. We created three applications, presenting technological plants most commonly used in education at our department. The first application provides simulation of tank storage system dynamics (Fig. 1). It contains two general mathematical models of plant, the non-linear and linear (obtained by linearization). Each of the models consists optionally of 1-3 ordinary differential equations (ODE), depending on number of tanks in system. The second simulation contains model of a tube heat exchanger (Fig. 2), which is discretized to five segments. Mathematical model consists of five linear ODEs. The last application is simulation of continuous stirred-tank reactor dynamics

(Fig. 3), with mathematical model containing both types of ODEs (linear and non-linear).

So far, all applications are in Slovak language only, but we plan to provide interface with more languages as well.



Zadávanie parametrov

○ - vzorový príklad

Počet zásobníkov: 3 ▾

Matematický model zásobníkov:
- lineárny / nelineárny
- zapojenie s interakciou
- počet zásobníkov 1-3

○ - vymazať parametre

q_0 = 10 $m^3.s^{-1}$

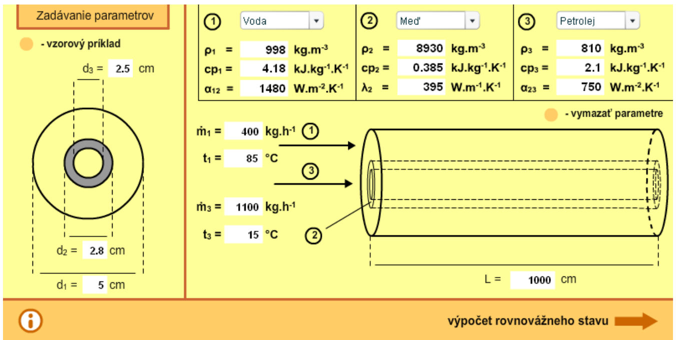
F_1 = 8 m^2 F_2 = 9 m^2 F_3 = 14 m^2

k_{11} = 11 $m^{2.5}.s^{-1}$ k_{22}= 8 $m^{2.5}.s^{-1}$ k_{33} = 11 $m^{2.5}.s^{-1}$

q_0 - objemový prietok kvapaliny na vstupe do 1. zásobníka
q_1, q_2, q_3 - objemové prietoky kvapaliny na výstupoch zo zásobníkov
F_1, F_2, F_3 - obsahy podstáv zásobníkov
k_{11}, k_{22}, k_{33} - konštanty ventilov
h_1, h_2, h_3 - výšky hladín v zásobníkoch

výpočet rovnovážneho stavu

Fig. 1. The first screen of Flash application for liquid storage tank system, showing forms for input parameters, which are used for computation of mathematical model and steady state of system



Zadávanie parametrov

○ - vzorový príklad

d_3 = 2.5 cm
d_2 = 2.8 cm
d_1 = 5 cm

① Voda ▾
ρ_1 = 998 $kg.m^{-3}$
cp_1 = 4.18 $kJ.kg^{-1}.K^{-1}$
α_{12} = 1480 $W.m^{-2}.K^{-1}$

② Meď ▾
ρ_2 = 8930 $kg.m^{-3}$
cp_2 = 0.385 $kJ.kg^{-1}.K^{-1}$
λ_2 = 395 $W.m^{-1}.K^{-1}$

③ Petrolej ▾
ρ_3 = 810 $kg.m^{-3}$
cp_3 = 2.1 $kJ.kg^{-1}.K^{-1}$
α_{23} = 750 $W.m^{-2}.K^{-1}$

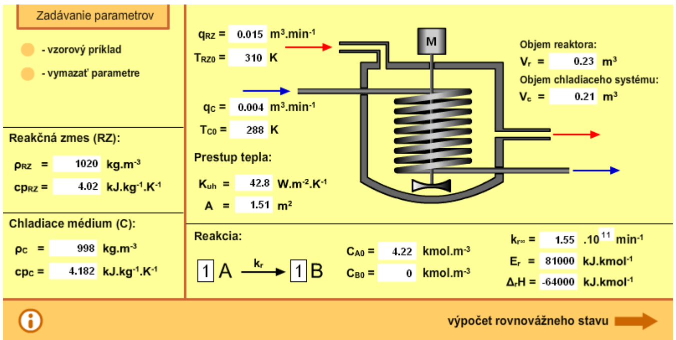
○ - vymazať parametre

$\dot{m}_1$ = 400 $kg.h^{-1}$ ①
t_1 = 85 °C
$\dot{m}_3$ = 1100 $kg.h^{-1}$
t_3 = 15 °C

L = 1000 cm

výpočet rovnovážneho stavu

Fig. 2. The first screen of Flash application for tube heat exchanger



Zadávanie parametrov

○ - vzorový príklad
○ - vymazať parametre

Reakčná zmes (RZ):
ρ_{RZ} = 1020 $kg.m^{-3}$
cp_{RZ} = 4.02 $kJ.kg^{-1}.K^{-1}$

Chladiace médium (C):
ρ_C = 998 $kg.m^{-3}$
cp_C = 4.182 $kJ.kg^{-1}.K^{-1}$

q_{RZ} = 0.015 $m^3.min^{-1}$
T_{RZ0} = 310 K
q_C = 0.004 $m^3.min^{-1}$
T_{C0} = 288 K

Prestup tepla:
K_{uh} = 42.8 $W.m^{-2}.K^{-1}$
A = 1.51 m^2

Objem reaktora:
V_r = 0.23 m^3
Objem chladiaceho systému:
V_c = 0.21 m^3

Reakcia:
$\boxed{1}\,A \xrightarrow{k_r} \boxed{1}\,B$

C_{A0} = 4.22 $kmol.m^{-3}$
C_{B0} = 0 $kmol.m^{-3}$

$k_{r\infty}$ = 1.55 $.10^{11}$ min^{-1}
E_r = 81000 $kJ.kmol^{-1}$
Δ_rH = -64000 $kJ.kmol^{-1}$

výpočet rovnovážneho stavu

Fig. 3. The first screen of Flash application for continuous stirred-tank reactor

When user starts application, he has to fill input parameters for mathematical model computation. The application includes a script that checks the correct format for all input parameters. After this procedure, application will compute and display steady state of plant. Before dynamics simulation starts, user fills simulation parameters like duration and steps for input variables. In this phase, another script checks the correct format of parameters. Then user pushes the button for executing system simulation. If all input parameters are correctly filled and all necessary setting are done, the application switches to simulation screen. When the simulation is running, the screen provides observation of plant states (numerically and graphically). After the simulation is finished, the application switches to another screen, where user can collect data in several structure types (text field, XML, MATLAB array).

3. MATHEMATICAL IMPLEMENTATION OF PID CONTROLLER

Mathematical model of simulated plant is in form of ordinary differential equations, therefore it must be solved by appropriate numerical method. During the development of original virtual laboratory for system dynamics simulation, we chose 4th order Runge-Kutta method to solve ODE problem. For the PID controller, we implement it in discrete form. To get appropriate discrete controller from a standard expression of the PID, one possibility is to transform derivative and integral components from continuous-time to discrete form (Bobál et al., 2005). The standard form is as follows

$$u(t_k) = K_p\left(e(t_k) + \frac{1}{T_i}\int_0^{t_k} e(\tau)\mathrm{d}\tau + T_d\frac{\mathrm{d}e(t_k)}{\mathrm{d}t}\right) \qquad (1)$$

The first-order derivative is approximated by backward finite difference

$$\frac{\mathrm{d}e(t_k)}{\mathrm{d}t} = \frac{e(t_k) - e(t_{k-1})}{\Delta t} \qquad (2)$$

The integral term is discretized with a sampling time Δt to finite summation

$$\int_0^{t_k} e(\tau)\mathrm{d}\tau = \sum_{i=1}^{k} e(t_i)\Delta t \qquad (3)$$

Final expression for implementation of the PID controller (4) is obtained by differentiating $u(t)$, using first and second differences.

$$\begin{aligned} u(t_k) = u(t_{k-1}) + K_p\Bigg[&\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t}\right)e(t_k) \\ &- \left(1 + \frac{2T_d}{\Delta t}\right)e(t_{k-1}) + \frac{T_d}{\Delta t}e(t_{k-2})\Bigg] \end{aligned} \qquad (4)$$

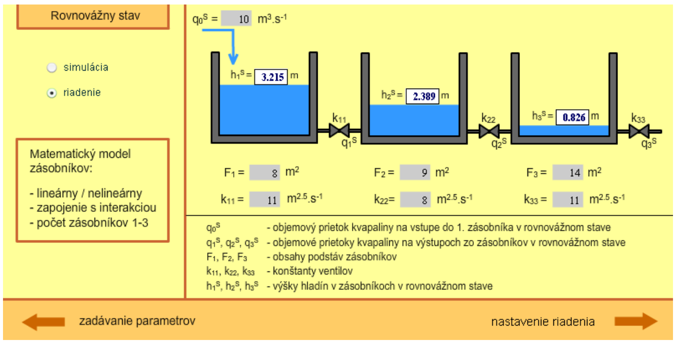
4. OVERVIEW OF THE NEW FEATURES

The first screens of applications are the same as in the original ones (Fig. 1, 2, and 3). User has to setup the parameters for mathematical model computation. Mathematical model of steady state for each dynamical system used in applications is defined as a system of algebraic equations.

The steady state observation (Fig. 4) now features two new radio buttons (choice between simulation of dynamics and control). When simulation button is checked, application switches to its original part. To use new features of PID control, user has to check the process control button.

The control setup screen (Fig. 5) defines types of mathematical model (this feature is available only for storage tank system, where user can choose between linear and non-linear model), manipulated and output variable (MV and OV), controller structure (P, PI, or PID), controller parameters, and set points of MV.

When user pushes the button for simulation of process control, the application executed script that runs simple



○ simulácia
● riadenie

Matematický model zásobníkov:

- lineárny / nelineárny
- zapojenie s interakciou
- počet zásobníkov 1-3

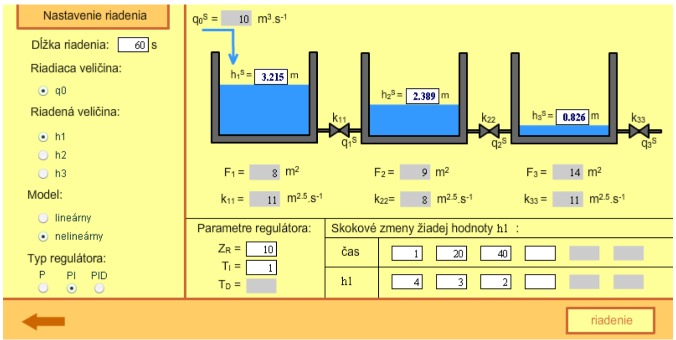
q_0^S = 10 $m^3.s^{-1}$

h_1^S = 3.215 m
h_2^S = 2.389 m
h_3^S = 0.826 m

F_1 = 8 m^2 F_2 = 9 m^2 F_3 = 14 m^2
k_{11} = 11 $m^{2.5}.s^{-1}$ k_{22}= 8 $m^{2.5}.s^{-1}$ k_{33} = 11 $m^{2.5}.s^{-1}$

q_0^S - objemový prietok kvapaliny na vstupe do 1. zásobníka v rovnovážnom stave
q_1^S, q_2^S, q_3^S - objemové prietoky kvapaliny na výstupoch zo zásobníkov v rovnovážnom stave
F_1, F_2, F_3 - obsahy podstáv zásobníkov
k_{11}, k_{22}, k_{33} - konštanty ventilov
h_1^S, h_2^S, h_3^S - výšky hladín v zásobníkoch v rovnovážnom stave

Fig. 4. Screen of Flash application showing steady state of system and choice between simulation and process control



Dĺžka riadenia: 60 s

Riadiaca veličina:
● q0

Riadená veličina:
● h1
○ h2
○ h3

Model:
○ lineárny
● nelineárny

Typ regulátora:
P ○ PI ● PID ○

q_0^S = 10 $m^3.s^{-1}$

h_1^S = 3.215 m
h_2^S = 2.389 m
h_3^S = 0.826 m

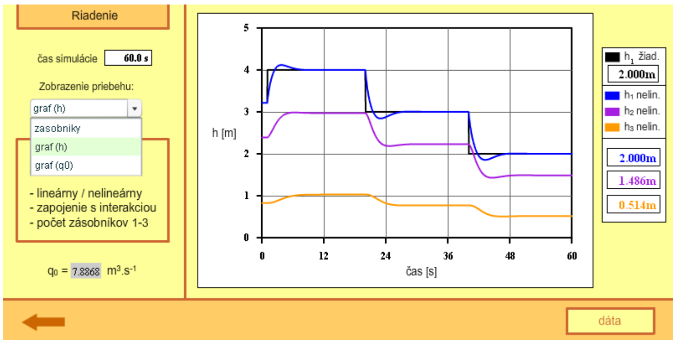
F_1 = 8 m^2 F_2 = 9 m^2 F_3 = 14 m^2
k_{11} = 11 $m^{2.5}.s^{-1}$ k_{22}= 8 $m^{2.5}.s^{-1}$ k_{33} = 11 $m^{2.5}.s^{-1}$

Parametre regulátora:
Z_R = 10
T_I = 1
T_D =

Skokové zmeny žiadej hodnoty h1 :

čas	1	20	40			
h1	4	3	2			

Fig. 5. Setup screen for OV/MV, controller type (P, PI, PID), controller parameters and setpoint step timing



čas simulácie 60.0 s

Zobrazenie priebehu: graf (h)

- lineárny / nelineárny
- zapojenie s interakciou
- počet zásobníkov 1-3

q_0 = 7.8868 $m^3.s^{-1}$

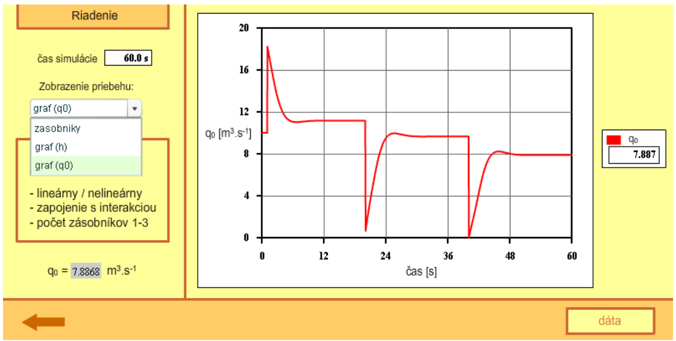
h_1 žiad. 2.000m
2.000m
1.486m
0.514m

Fig. 6. Application interface, showing graph for levels of liquid in tanks and controlled variable

pre-simulation of control to get information about its behavior. This is important for setting up the graph scaling in simulation screen.

The simulation screen appears if all parameters are correctly filled. User can observe simulation as animated scheme of plant, where application shows only periodically updated values of plant states, or as graphs of MVs (Fig. 6) and OVs (Fig. 7).

After simulation of plant control is finished, the application switches to screen with results (Fig. 8). On this screen, user can collect data from simulation in several structures and formats. Application provides data as plain text field, XML structure, and MATLAB formatted array.



čas simulácie 60.0 s

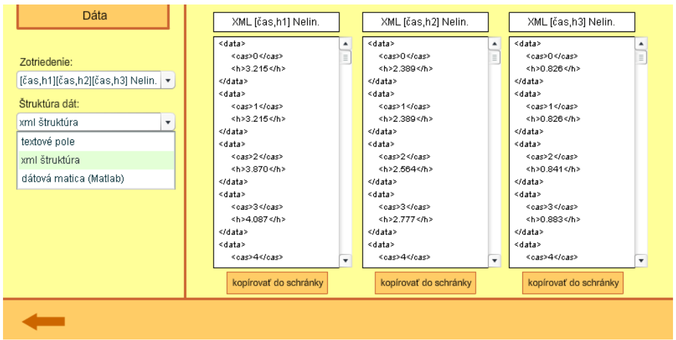
Zobrazenie priebehu: graf (q0)

- lineárny / nelineárny
- zapojenie s interakciou
- počet zásobníkov 1-3

q_0 = 7.8868 $m^3.s^{-1}$

q_0 7.887

Fig. 7. Application interface, showing graph for MV during simulation of control



Zotriedenie: [čas,h1][čas,h2][čas,h3] Nelin.

Štruktúra dát: xml štruktúra

XML [čas,h1] Nelin.

```
<data>
   <cas>0</cas>
   <h>3.215</h>
</data>
<data>
   <cas>1</cas>
   <h>3.215</h>
</data>
<data>
   <cas>2</cas>
   <h>3.870</h>
</data>
<data>
   <cas>3</cas>
   <h>4.087</h>
</data>
<data>
   <cas>4</cas>
```

XML [čas,h2] Nelin.

```
<data>
   <cas>0</cas>
   <h>2.389</h>
</data>
<data>
   <cas>1</cas>
   <h>2.389</h>
</data>
<data>
   <cas>2</cas>
   <h>2.564</h>
</data>
<data>
   <cas>3</cas>
   <h>2.777</h>
</data>
<data>
   <cas>4</cas>
```

XML [čas,h3] Nelin.

```
<data>
   <cas>0</cas>
   <h>0.826</h>
</data>
<data>
   <cas>1</cas>
   <h>0.826</h>
</data>
<data>
   <cas>2</cas>
   <h>0.841</h>
</data>
<data>
   <cas>3</cas>
   <h>0.883</h>
</data>
<data>
   <cas>4</cas>
```

Fig. 8. Results from simulation can be exported in several structures (text field, XML, MATLAB array)

5. CONCLUSIONS

We have extended our virtual Flash laboratory with new features for PID control. Users can now either simulate of control process models. This new features have been added to all recently developed simulation applications, containing models of storage tank system, tube heat exchanger, and continuous stirred-tank reactor.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency of the Slovak Republic under the grant 1/0071/09.

The paper is supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Bobál, V., Böhm, J., Fessl, J., and Macháček, J. (2005). *Digital Self-tuning Controllers. Algorithms, Implementation and Applications*. Springer-Verlag, London, U.K.

Čirka, Ľ., Kalúz, M., Kvasnica, M., and Fikar, M. (2010). Virtual laboratory. In *Proceedings of the 9th International Scientific - Technical Conference Process Control 2010*, C029a – 1–C029a – 8. University of Pardubice, Kouty nad Desnou, Czech Republic.

Kalúz, M. (2010). *Virtuálne laboratórium*. Master's thesis, ÚIAM FCHPT STU v Bratislave, Radlinského 9, 812 37 Bratislava.

Kalúz, M., Čirka, Ľ., and Fikar, M. (2010). MATLAB Builder JA in control engineering education at FCFT STU. In *Technical Computing Bratislava 2010*, volume 18, 1–5.

Malki, A.H. and Matarrita, A. (2002). Virtual labs for distance education classes. In *Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference*. American Society for Engineering Education, The University of Louisiana, Lafayette.

Bode Plots in Maxima Computer Algebra System

D. Gajdošík and K. Žáková

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava
(e-mail:katarina.zakova@stuba.sk)

Abstract: The aim of the paper is to demonstrate possibilities of open software environment Maxima in educational process at technical universities whereby our attention is dedicated to the teaching of Bode plots. The developed procedure for drawing its asymptotic approximation can be used both for checking results on the base of the entered system transfer function and also for self testing purposes. In addition, the results were used for the building of web application that will be used in frame of the subject Control Theory.

Keywords: computer algebra system, open source, Maxima, Bode plots, control theory

1. INTRODUCTION

Computer algebra systems are systems that enable to solve mathematical tasks either in numerical or symbolical way. Following Internet resources one can find several products that can be divided into two main groups: proprietary software that is a product of some company taking care about the whole development and support and open software developed usually by an enthusiastic person or group of persons that is available for free.

We decided to devote our attention to open alternative since it helps to solve problems connected with legal purchase of software that is used. In the case of proprietary products students usually use another version of software at the faculty and another version (very often illegal) at home. After finishing their study the proprietary software that is devoted to specific purposes like mathematical calculations are is too expensive for paying licences and keeping it regularly updated. Therefore they stop to use it. Open software can overcome these problems and therefore it could be good if students would know about its existence already during their stay at university.

Looking for the best alternative of computer algebra system (CAS) we had several requirements. The selected software should be open and enable at least

- numerical and symbolical calculations,
- simplification and processing of algebraic expressions,
- matrix computations,
- solution of a system of linear, nonlinear and differential equations,
- differentiate and integrate functions,
- graphical presentations of results,
- simple programming.

Except of that the selected CAS should have good ongoing support, cooperative community, continuing development and it would be welcomed if it could be supported both for Windows and Unix/Linux operating system. After considering several software environments (Axiom, Maxima, SymPy, Sage, Yacas, etc.) we decided to choose Maxima computer algebra system that can be an open alternative for such proprietary programs as Maple, Mathematica or Matlab with Symbolic Toolbox are.

2. BODE PLOTS

A Bode plot is a plot showing one of graphical representation of frequency response characteristics for linear time invariant system usually described by a transfer function. The sinusoidal transfer function of any linear system (obtained by substituting $j\omega$ for s in the transfer function of the system) is characterized by its magnitude and phase angle with frequency as parameter. Therefore the Bode plots are represented by two separate plots - one expressing the magnitude of the frequency response gain (magnitude vs. frequency plot) and another one expressing the frequency response phase shift (phase vs. frequency). Standard Bode plots are logarithmic on the frequency axis, and plot the magnitude in dB's (decibels) and phase in degrees.

Using computers drawing of plots can be done by simple entering many values for the frequency, calculating the magnitude and phase at each frequency and displaying them.

However, there are reasons to develop a method for sketching Bode diagrams manually. By drawing the plots by hand you develop an understanding about how the location of poles and zeros influences the shape of the plots. With this knowledge you can predict how a system behaves in the frequency domain by simply examining its transfer function. On the other hand, if you know the shape of transfer function that you want, you can use your knowledge of Bode diagrams to generate the transfer function.

Therefore our aim was not only to draw Bode plots of linear dynamical systems but to visualise also their asymptotic approximations. The main advantage of using Bode plots is that multiplication of magnitudes can be converted into addition.

To sketch asymptotic approximates of Bode diagram it is necessary to draw separate asymptotic curves for each of the transfer function factors

- gain,
- integral and derivative factors,
- first-order factors,
- quadratic factors.

The composite curve (both for magnitude and phase as well) is obtained by adding algebraically the individual curves. In order students could check their results, the developed Maxima procedure tries to offer results in the way as students obtain them manually. Actually, any change in the slope of the magnitude curve is made only at the break frequencies (break frequency or corner frequency is the frequency at which two asymptotes meet) of the transfer function. Therefore, instead of drawing individual magnitude curves and adding them up, it is possible to sketch the magnitude curve without sketching individual curves. In similar way, it is possible to proceed also in the case of the phase curve.

3. MAXIMA IMPLEMENTATION

Computation and visualisation of Bode diagram in Maxima can be realised very easily by using standard functions `bode_gain()` for magnitude plot and `bode_phase()` for phase plot. The input of both functions is a transfer function of a system. These functions were already created by other persons and there is no need to change them.

Our interest was first of all dedicated to the creation of Bode diagram asymptotic approximations. As it was already told the created procedure tries to demonstrate all steps of the manual graphical sketching that include

- transformation of the transfer function to the normalized form

$$F(s) = K\frac{s^p \prod(T_i s+1)\prod\left(\mathrm{T_k}^2\mathrm{s}^2+2\mathrm{b_k}\mathrm{T_k}\mathrm{s}+1\right)}{s^r \prod(T_j s+1)\prod\left(\mathrm{T_l}^2\mathrm{s}^2+2\mathrm{b_l}\mathrm{T_l}\mathrm{s}+1\right)}$$

- separation of all transfer function factors (gain, integral and derivative factors, first-order factors, quadratic factors)
- calculation of break frequencies
- visualisation of asymptotic approximation of magnitude and phase plots for each separate factor
- adding up the previous results and drawing the final asymptotic approximation for both Bode plots.

Similar to standard Bode functions the created Maxima function has one input – the transfer function of the system. The function output generates asymptotic approximations of frequency responses together with all break frequencies.

4. WEB APPLICATION

It is to say that Maxima is the software that was developed only for the use on local computer. Since we want to use it as an engine for driving our web application it was necessary to prepare tools that would enable to communicate with Maxima via Internet. This functionality was already realised and was described in our previous papers (see e.g. Magyar et all, 2009). The developed implementation enables to exchange commands and data between Maxima software installed on linux server and our web application only on the base of created php functions.

The front end of the application was created using standard technologies as HTML, CSS and JavaScript. For the data exchange the JSON format was used. For plotting of the graphical dependencies we used free Javascript plotting library for jQuery that is called jqPlot. It produces graphical plots of arbitrary datasets on the client-side.

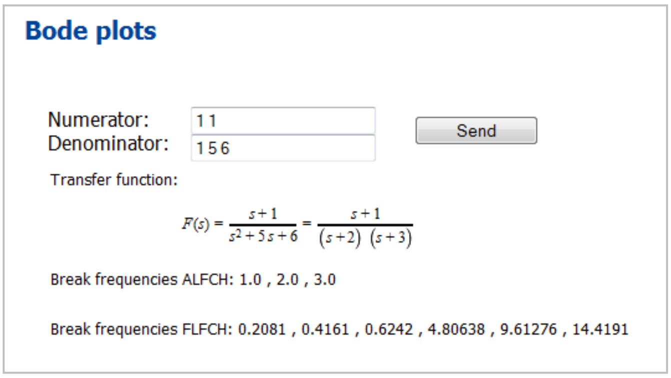
4.1 Basic functionality

The developed web application enables to draw asymptotic approximations of Bode plots. During university courses students learn to sketch these approximations in order they would be able to estimate behaviour of dynamical systems also without using computer.

The considered system is defined by the transfer function that is entered using coefficients of the transfer function numerator and denominator. In Fig. 1 the system with the transfer function

$$F(s) = \frac{\mathrm{s}+1}{\mathrm{s}^2+5\mathrm{s}+6} = \frac{\mathrm{s}+1}{(\mathrm{s}+2)(\mathrm{s}+3)} = \frac{1}{6}\frac{\mathrm{s}+1}{\left(\frac{\mathrm{s}}{2}+1\right)\left(\frac{\mathrm{s}}{3}+1\right)}$$

is started to be solved.


Bode plots

Numerator: 1 1
Denominator: 1 5 6

Transfer function:

$$F(s) = \frac{s+1}{s^2+5s+6} = \frac{s+1}{(s+2)\ (s+3)}$$

Break frequencies ALFCH: 1.0 , 2.0 , 3.0

Break frequencies FLFCH: 0.2081 , 0.4161 , 0.6242 , 4.80638 , 9.61276 , 14.4191

Fig. 1. Form for entering the input coefficients

After submitting the entered parameters, the user input is sent for processing to Maxima where break frequencies and corresponding slopes of asymptotes are calculated. The graphical result is displayed again in web browser (Fig.2). The advantage is that user can display resulting frequency response, final asymptotic approximation of Bode plots or only asymptotic approximation of chosen separate factor. In this way student can check all steps of own manual procedure.

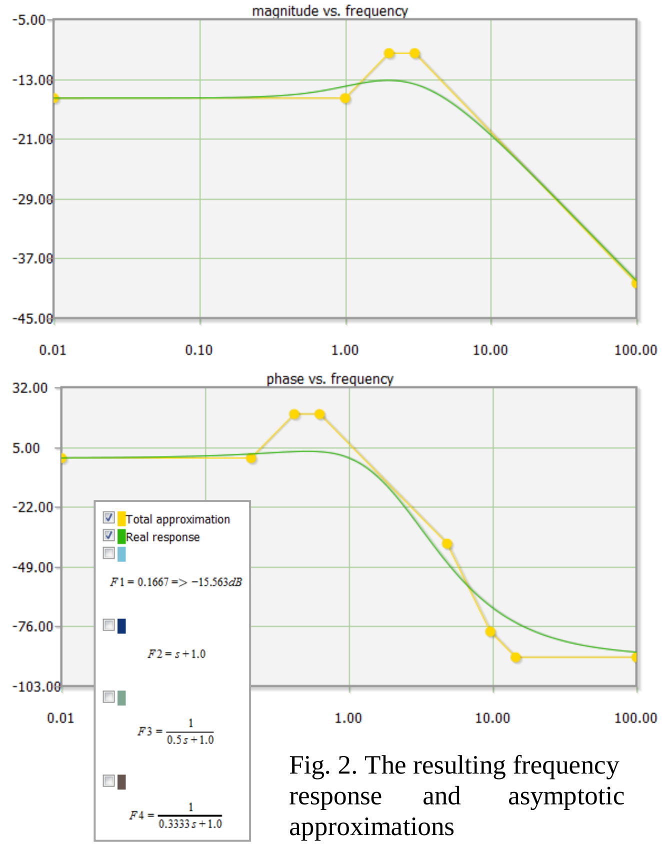


Fig. 2. The resulting frequency response and asymptotic approximations

4.2 *Self testing*

The developed application can also be used in other way. Students should be able not only to sketch Bode plots but also to analyse the model of the system according to sketched frequency characteristics. Therefore we prepared the modification of the application that can be used for self testing of students. Instead of entering parameters of the transfer function by student the parameters are determined randomly by computer. Then, the resulting asymptotic approximation of the magnitude Bode plot is visualised to the student (Fig.3). According to break frequencies student can determine time constants of the system and according to the vertical position of the characteristics the gain of the system can be found. In this way student can determine the resulting transfer function that is after entering to the web page compared with the originally generated system description (Fig.4). Maxima computer algebra system is able to compare both transfer functions without any problem.

In the case of self testing we decided to concentrate only on transfer functions with real poles and real zeros. The transfer functions containing complex roots are not considered because of the problematic determination of a system damping from the graph. From similar reason we do not consider systems with non minimal phase. The transfer function that is taken into account has the following form

$$F(s) = K\frac{\prod_{i=1}^{m}(T_i s + 1)}{s^r \prod_{j=1}^{n}(T_j s + 1)}$$

whereby the introduced parameters can achieve the following values: $K = 1,\ 100;\ T_i, T_j = 0,\ 0.125,\ 0.2,\ 0.25,\ 0.5,\ 1;$ $r = 0, 1, 2;\ m, n = 1, 2, 3.$

The parameters are restricted to the mentioned values because of their easier identification from the graphical presentation of the magnitude plot. We decided to use only values that can be read from the picture exactly and without any doubt. It is very convenient and useful if no values rounding has to be used. In addition, such restriction of values doesn't mean any limitation to the complexity of the task. It is more important to consider various structures of transient functions.

Following the considered values of parameters the considered transient function can achieve several forms e.g.

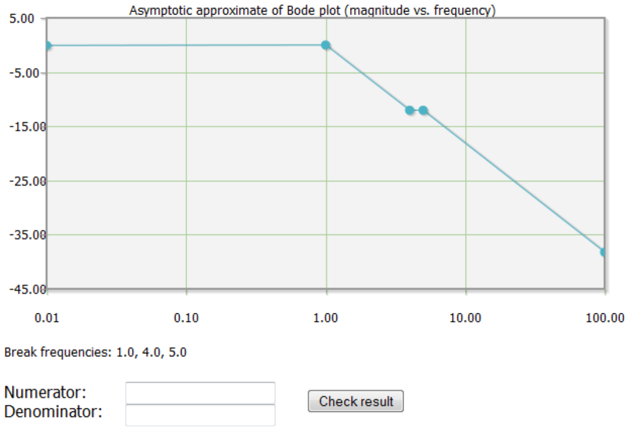
$$F_1(s) = \frac{K}{s(Ts+1)}$$

$$F_2(s) = \frac{K}{(Ts+1)^2}$$

$$F_3(s) = \frac{K}{(T_1 s+1)(T_2 s+1)(T_3 s+1)}$$

$$F_4(s) = \frac{K(T_1 s+1)}{s^2(T_2 s+1)}$$

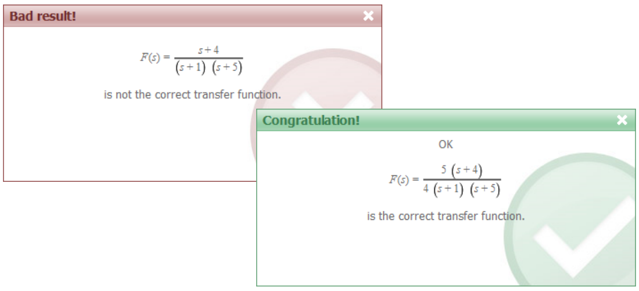
Of course, we didn’t introduce all possibilities. The number of transient functions should be sufficient for comprehension of the problem by student.



Break frequencies: 1.0, 4.0, 5.0

Numerator:
Denominator:

Fig. 3. Randomly generated asymptotic approximation of the magnitude Bode plot


Bad result!

$F(s) = \frac{s+4}{(s+1)\ (s+5)}$

is not the correct transfer function.

Congratulation!

OK

$F(s) = \frac{5\ (s+4)}{4\ (s+1)\ (s+5)}$

is the correct transfer function.

Fig.4. Verification of the result

5. CONCLUSIONS

In the paper we exploited a possibility to use Maxima software for solution of one problem from Control Theory.

The advantage is that the tool is available online via Internet for all interested users in any time they need. Since the presented topic is part of the study at probably all technical universities it presents the useful instrument in an educational process mainly in online and distance forms of education.

The last thing to mention is the fact that each version of the introduced application was prepared by regular students in frame of their individual projects. They like programming and together with it they learn control theory, too. The idea of "learning by doing" is in this case very actual and the results showed that it is good to continue in this direction also in future.

The application is a part of study materials that are available on the address:

http://obelix.urpi.fei.stuba.sk/~tar/pedagogika/tar1/

ACKNOWLEDGMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09 and by the Grant VEGA No. 1/0656/09. This support is very gratefully acknowledged.

Authors thank to Ľudovít Vörös, Tomáš Hojč and Ján Šovčík for their help with the problem algorithmization and programming.

REFERENCES

Cheever, E. (2005). What Bode Plots Represent, *url*: http://jegyzet.sth.sze.hu/ftp/!BSc/Szabalyozastechnika/BodePl.pdf

Jakab, F., Andoga, V., Kapova, L., Nagy, M. (2006). Virtual Laboratory: Component Based Architecture Implementation Experience, *Electronic computer and informatics*, September, Košice-Herľany, Slovakia.

Leão, C. P., A. E. Rodrigues (2004). Transient and steady-state models for simulated moving bed processes: numerical solutions. *Computers & Chemical Engineering* 28(9): 1725-1741.

Lutus P. (2007), Symbolic Mathematics Using Maxima, *url*: http://arachnoid.com/maxima/

Magyar, Z., T. Starý, L. Szolik, Ľ. Vörös, K. Žáková (2009). Modeling of Linear Dynamical Systems Using Open Tools. 10*th International Conference Virtual University*, December, Bratislava, Slovak Republic.

Ogata K. (1997). *Modern Control Engineering*, 3rd Edition, Prentice Hall London.

Resig J. and the jQuery Team (2009). jQuery Documentation. *url*: http://docs.jquery.com

Restivo, M. T., J. Mendes, A.M. Lopes, C.M. Silva, F. Chouzal (2009). A Remote Lab in Engineering Measurement, *IEEE Trans. on Industrial Electronics*, vol. 56, no.12, pp. 4436-4843.

Schauer, F., M. Ožvoldová, F. Lustig (2008). Real Remote Physics Experiments across Internet – Inherent Part of Integrated E-Learning. *Int. Journal of Online Engineering* (*iJOE*), 4, No 2.

Schmid, Chr. (2003). Internet - basiertes Lernen. *Automatisierungstechnik*, 51, No. 11, p. 485-493.

Turgeon, A. J. (2002) Implications of Web-Based Technology for Engaging Students in a Learning Society", *Journal of Public Service and Outreach* 2(2): 32-37.

Zolotová, I., M. Bakoš, L. Landryová (2007). Possibilities of communication in information and control systems. *Annals of the university Craiova, Series: Automation, Computers, Electronic and Mechatronic*, Vol.4(31), No.2, pp.163-168, ISSN 1841-062.

Žáková, K., M. Janotík (2004). Mathematical Modeling of Dynamic Systems: an interactive online lesson. 5*th int. conf.* "*Virtual University*", Bratislava, Slovakia.

Žáková, K. (2005). Control Theory - an interactive online course. 6*th int. conf.* "*Virtual University*", Bratislava, Slovakia.

Žáková, K., M. Sedlák (2006). Remote Control of Experiments via Matlab, *Int. Journal of Online Engineering* (*iJOE*), 2, No. 3.

jsMath: A Method of Including Mathematics in Web Pages, http://www.math.union.edu/~dpvc/jsMath/, Last modified: 10 Mar 2009.

Neuro-fuzzy Control of the Three Tank System

L. Blahová, J. Dvoran

Department of Information Engineering and Process Control,
Institute of Information Engineering, Automation and Mathematics,
Faculty of Chemical and Food Technology,
Slovak University of Technology in Bratislava
Radlinského 9, 812 37 Bratislava
(e-mail : lenka.blahova@stuba.sk)

Abstract: This paper presents the control design via the combination of the neural predictive controller and the neuro-fuzzy controller type of ANFIS. The neuro-fuzzy controller works in parallel with the predictive controller. This controller adjusts the output of the predictive controller, in order to enhance the predicted inputs. The performance of our proposal is demonstrated on the three tank system control problem with disturbance. Simulation results demonstrate the effectiveness and robustness of the proposed approach.

1. INTRODUCTION

The aim of process control is to achieve the target value of the given variable. This is mainly the task of the properly designed controller. The controller should also provide some flexibility in case an unexpected failure, change of conditions, etc.

Today, there are many methods for designing intelligent controllers, such as fuzzy control, neural networks or expert systems. Appropriate combinations of these methods offer a number of other design possibilities.

This paper describes the above mentioned combination of two methods of intelligent system controlling. By the parallel connection of predictive and neural-fuzzy controller, we aimed to obtain better results of the reference variable in terms of lowering its overshooting and reducing the control time. The designed system with two connected controllers was tested using a three tank system. The tank system introduces one of the nonlinear type of the chemical-technological processes.

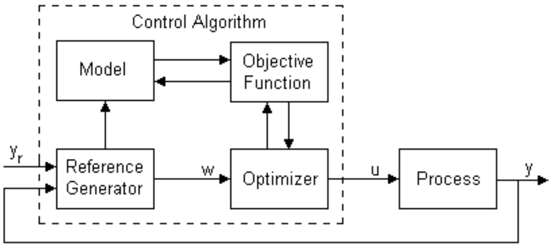


Fig. 1. Model-based predictive control scheme

2. PREDICTIVE CONTROL

MBPC (Model-Based Predictive Control) is a name of a several different control techniques (A. Vasičkaninová 2008). All are associated with the same idea. The prediction is based on the model of the process (Figure 1).

The controller uses a neural network model to predict future plant responses to potential control signals. An optimization algorithm then computes the control signals that optimize future plant performance. The neural network plant model is trained offline, in bath form, using any of the training algorithms. The controller, however, requires a significant amount of online computation, because an optimization algorithm is performed at each sample time to compute the optimal control input. The model predictive control method is based on the receding horizon technique. The neural network model predicts the plant response over a specified time horizon. The predictions are used by a numerical optimization program to determine the control signal that minimizes the following performance criterion over the specified horizon.

$$J(t,u(k)) = \sum_{i=N_1}^{N_2} (y_m(t+i) - y_r(t+i))^2 + \lambda \sum_{i=1}^{N_u} (\Delta u(t+i-1))^2 \qquad (1)$$

where N_1, N_2 and N_u define the horizons over the tracking error and the control increments are evaluated. The Δu variable is the tentative control signal, y_r is the desired response and y_m is the network model response. The λ value determines the contribution that the sum of the squares of the control increments has on the performance index.

The controller consists of the neural network plant model and the optimization block. The optimization block determines the values of *u* that minimize *J*, and then the optimal *u* is input to the plant.

Equation (1) is used in combination with input and output constraints:

$$\begin{array}{ccccc} u_{\min} & \leq & u & \leq & u_{\max} \\ \Delta u_{\min} & \leq & \Delta u & \leq & \Delta u_{\max} \\ y_{\min} & \leq & y & \leq & y_{\max} \\ \Delta y_{\min} & \leq & \Delta y & \leq & \Delta y_{\max} \end{array} \tag{2}$$

3. NEURO-FUZZY CONTROLLER

The neural predictive controller can be extended with neuro-fuzzy controller, connected in parallel (Figure 2).

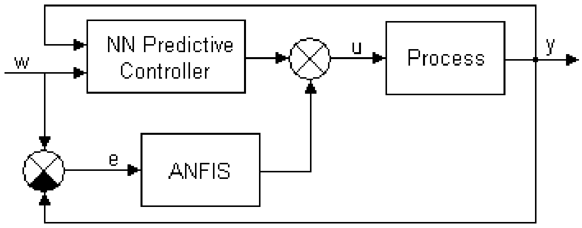


Fig. 2. Neuro-fuzzy control scheme

Neuro-fuzzy systems, which combine neural networks and fuzzy logic, have recently gained a lot of interest in research and application. A specific approach in neuro-fuzzy development is the ANFIS (Adaptive Network-based Fuzzy Inference System) (M. Agil 2007). ANFIS uses a feed forward network to search for fuzzy decision rules that perform well on a given task. Using a given input-output data set, ANFIS creates an Fuzzy Inference System for which membership function parameters are adjusted using a combination of a backpropagation and least square method. The ANFIS architecture of the first-order Takagi-Sugeno inference system is shown in Figure 3.

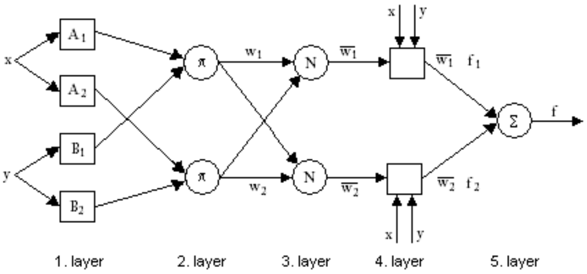


Fig. 3. System architecture ANFIS

4. EXPERIMENTAL

4.1 Process

We assume a non-linear system of three tanks shown in Figure 4 that is described by three sets of differential equations (Mikleš and Fikar 2007).

$$\frac{dh_1}{dt} = \frac{q_{01}}{S_1} - \frac{k_{11}}{S_1}\sqrt{h_1 - h_2} \tag{3}$$

$$\frac{dh_2}{dt} = \frac{q_{02}}{S_2} + \frac{k_{11}}{S_2}\sqrt{h_1 - h_2} - \frac{k_{22}}{S_2}\sqrt{h_2} \tag{4}$$

$$\frac{dh_3}{dt} = \frac{k_{22}}{S_3}\sqrt{h_2} - \frac{k_{33}}{S_3}\sqrt{h_3} \tag{5}$$

where S_1, S_2, S_3 [dm^2] are the cross-sectional areas of tanks, h_1, h_2, h_3 [dm] – heights of liquid in tanks, k_{11}, k_{22}, k_{33} [dm$^{2.5}$s^{-1}] – constants , q_{01} [dm^{3}s^{-1}] – inlet volumetric flow rate to the first tank, q_{02}, q_1 [dm^{3}s^{-1}] – inlet volumetric flow rate to the second tank, q_2 [dm^{3}s^{-1}] – inlet volumetric flow rate to the third tank, q_3 [dm^{3}s^{-1}] – outlet volumetric flow rate from the third tank and t [s] – time variable. The concrete values of the parameters are summarized in Table 1.

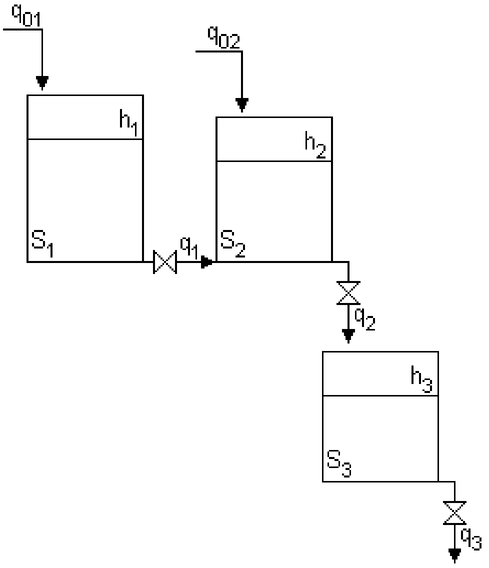


Fig. 4. Signification scheme of a three tank system

Table 1. Parameters of the tank system

Variable	Unit	Value
S_1	dm^2	3
S_2	dm^2	2.5
S_3	dm^2	2
k_{11}	dm$^{2.5}$s^{-1}	1.8
k_{22}	dm$^{2.5}$s^{-1}	1.3
k_{33}	dm$^{2.5}$s^{-1}	1.4
q_{01}	dm^{3}s^{-1}	1
q_{02}	dm^{3}s^{-1}	0.3

The height of liquid in the third tank h_3 is controlled variable and inlet volumetric flow rate to the first tank q_{01} is input variable. The process state variables are heights of liquid in tanks (h_1, h_2 and h_3).

4.2 Process control in the nominal state

Firstly, process was simulated with neural predictive controller (NNPC). To set this controller neural network process model was needed. Neural network model of three tanks system was trained offline based on non-linear process input and output data by Levenberg-Marquardt back propagation method. When optimization parameters were adjusted, tanks system was further controlled by NNPC controller.

Secondly, tanks system was controller with neuro-fuzzy controller (NFC) formed from neural predictive controller and ANFIS controller. ANFIS was trained by PID controller. PID parameters were designed by Strejc method (Bakošová et al. 2003) in five training periods. ANFIS have two inputs: set-point error *e* and derivation of set-point error *de*. Sixteen membership function bell shape were chosen for ANFIS input: nine for variable *e* and seven for variable *de* (Figure 5). The neural predictive and the neuro-fuzzy cotnroller were tested in MATLAB/SIMULINK ® environment using neural network toolbox and fuzzy logic toolbox. This experiment was designed to compare a neural predictive controller with neuro-fuzzy controller performance while controlling and nominal process.

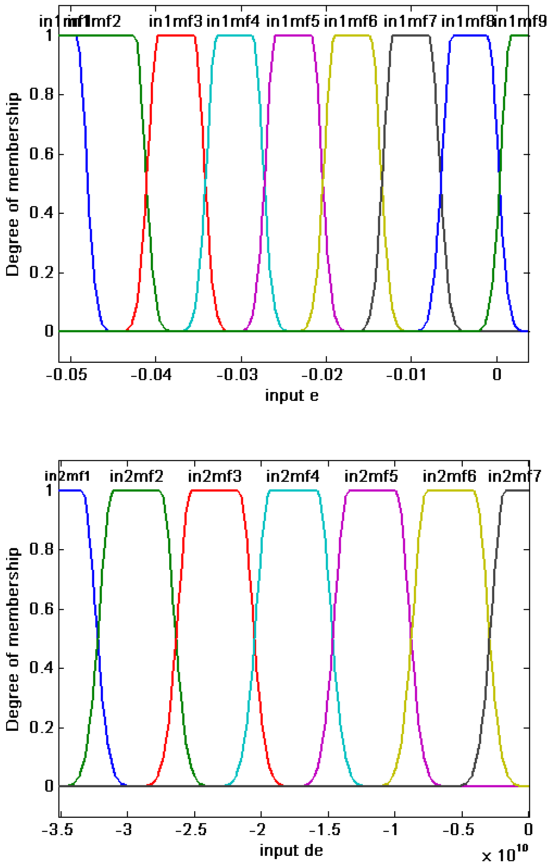


Fig.5. Membership functions for input variables *e* and *de*

For neural predictive controller had IAE criteria value 5.07 and for neuro-fuzzy controller had IAE criteria value 3.94. In Figure 6, set-point changes of the desired height profile were tracked with satisfactory result in both considered cases. However, it can be seen, that the controlled variable (h_3) profiles exhibit differences for both controllers compared. The neuro-fuzzy controller had more fainting performance that the neural predictive controller.

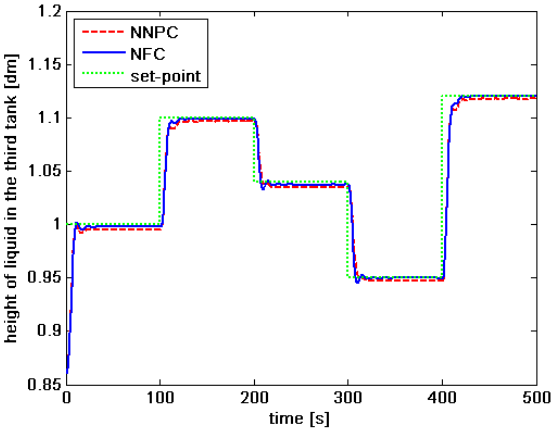


Fig.6. Comparison of NNPC and NFC performance for nominal plant

4.3 Process control in the perturbed state

Besides the good regulatory performance tested above, tracking abilities of controllers proposed in the presence of disturbances is of utmost importance. Disturbance was applied during the control curse and it was set as step change of inlet volumetric flow rate to the second tank (q_{02}). This disturbance was change in range ± 10% from nominal value q_{02}.

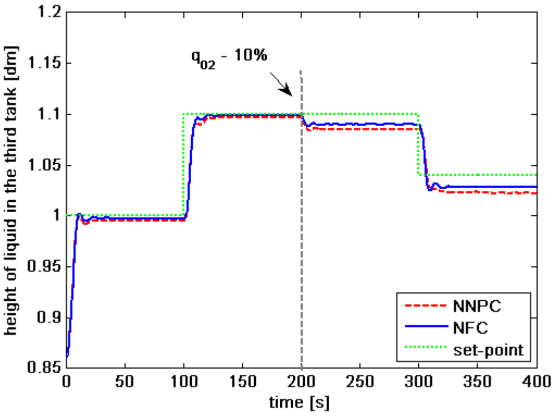


Fig.7. Comparison of NNPC and NFC performance for perturbed state – step change of q_{02} – 10% from nominal value

A comparison of the neural predictive controller and the neuro-fuzzy controller performance tested in the presence of process parameter is demonstrate in Figure 7,8 (the arrow is to show the time instants when disturbance was applied). IAE criteria values are shown in Table 2.

Table 2. Values of IAE criteria

Controller\Disturbance	q_{02} – 10% from nominal value	q_{02} + 10% from nominal value
NNPC	5.522	3.998
NFC	4.144	3.403

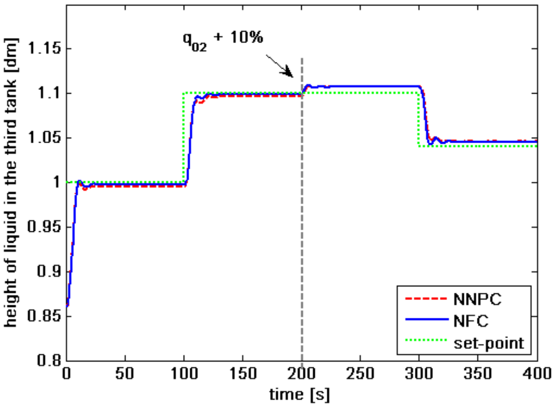


Fig.8. Comparison of NNPC and NFC performance for perturbed state – step change of q_{02} + 10% from nominal value

5. CONCLUSION

In this paper, we present intelligent control of a three tanks system. This intelligent control system is composed from two individual controllers: neural predictive controller and ANFIS controller.

The main goal of the resulting control system was to enhance a profile of height of liquid in the third tank by manipulating the inlet volumetric flow rate to the first tank. Simulation results and IAE criteria obtained demonstrated the usefulness and robustness of the proposed control system, and general advantages of the innovative technique in control application.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, 1/0537/10, 1/0095/11, and the Slovak Research and Development Agency under the project APVV-0029-07.

REFERENCES

M. Agil, I. Kita, A. Yano, and S. Nishiyama. Analysis and pridiction of flow from local source in a river basin using a Neuro-fuzzy modeling tool. In: *Jurnal of Enviromental Management*, 85, 215 – 223, (2007).

M. Bakošová, M. Fikar, Ľ. Čirka. *Automatic control fundamentals.* Laboratory exercises *(in Slovak)*, Vydavateľstvo STU, Bratislava, 2003.

J. Mikleš and M. Fikar. *Process Modelling, Identification and Control.* New York: Springer, Berlin Heidelberg (2007)

A. Vasičkaninová, M. Bakošová, A. Mészáros and J. Závacká. Model-based predictive control of a chemical reactor. In: *18th International Congress of Chemical and Process Engineering*, Orgit s.r.o., Praha, Czech Republic, 0623-1 – 0623-6 (2008)

Neuro-fuzzy Control of a Chemical Reactor with Uncertainties

A. Vasičkaninová, M. Bakošová, M. Karšaiová

Department of Information Engineering and Process Control,
Institute of Information Engineering, Automation and Mathematics,
Faculty of Chemical and Food Technology, Slovak University of Technology, Radlinského 9,
81237 Bratislava, Slovakia

Abstract: This work deals with the design and application of a neuro-fuzzy control of a chemical reactor. The reactor is exothermic one. There are two parameters with only approximately known values in the reactor. These parameters are the reaction enthalpies. Because of the presence of uncertainty in the continuous stirred tank reactor, the robust output feedback is designed. The simulation results confirm that fuzzy control is one of the possibilities for successful control of chemical reactors.

1. INTRODUCTION

It is well known that the control of chemical reactors often represents very complex problems (Luyben 2007), (Molnár et al. 2002). Continuous stirred tank reactors (CSTRs) are often used plants in chemical industry and especially exothermic CSTRs are very interesting systems from the control viewpoint (Bequette 1991). The dynamic characteristics may exhibit a varying sign of the gain in various operating points, the time delay as well as non-minimum phase behaviour. Various types of disturbances also affect operation of chemical reactors, operation of chemical reactors is corrupted by many different uncertainties. Some of them arise from varying or not exactly known parameters, as e.g. reaction rate constants, reaction enthalpies or heat transfer coefficients (Antonelli and Astolfi 2003). All these problems can cause poor control response or even instability of classical closed-loop control systems.

Effective control of CSTRs requires application of some of advanced methods, as e. g. robust control (Gerhard et al. 2004), (Tlacuahuac et al. 2005). Robust control has grown as one of the most important areas in modern control design since works by (Doyle 1981), (Zames 1983) and many others.

Soft computing is a collection of methodologies like fuzzy system, neural networks and genetic algorithm, designed to tackle imprecision and uncertainty involved in a complex nonlinear system. Recent reviews on soft computing around the world (Dote and Ovaska 2001) indicate that the number of soft computing based engineering applications is increasing.

Fuzzy system has been known to provide a framework for handling uncertainties and imprecision by taking linguistic information from human experts. Fuzzy logic controllers have the advantages over the conventional controllers: they are cheaper to develop, they cover a wider range of operating conditions, and they are more readily customizable in natural language terms. FLCs have been implemented successfully in a variety of applications (Shapiro 2004), (Hayward and Davidson 2003), (Peri and Simon 2005).

Fuzzy controllers are more robust than PID controllers because they can cover a much wider range of operating conditions than PID can, and can operate with noise and disturbances of different nature. Given the dominance of conventional PID control in industrial applications, it is significant both in theory and in practice if a controller can be found that is capable of outperforming the PID controller with comparable ease of use. Some of PID fuzzy controllers are quite close to this aim (Ying 2000). The simplest and most usual way to implement a fuzzy controller is to realize it as a computer program on a general purpose computer.

One popular soft computing method is neuro-fuzzy technique which is a hybrid combination of artificial neural networks (ANN) and fuzzy inference system (FIS). Adaptive Network based Fuzzy Inference System (ANFIS) (Jang 1993), (Jang et al. 1997) is such a neuro-fuzzy technique. A clustering algorithm partitions a data set into several groups such that the similarity within a group is larger than among groups (Jang et al. 1997). The idea of data grouping, or clustering, is simple in its nature and is close to the human way of thinking (Jain and Dubes 1988). A more recent overview can be found in a collection of (Bezdek and Pal 1992), (Backer 1995).

2. FUZZY CONTROL

Classic control theory is usually based on mathematical models which describe the behaviour of the process under consideration. The main aim of fuzzy control is to simulate a human expert (operator), who is able to control the process by translating the linguistic control rules into a fuzzy set theory.

In 1965 Lotfi A. Zadeh introduced fuzzy sets, where a more flexible sense of membership is possible. The past few years have witnessed a rapid growth in the use of fuzzy logic controllers for the control of processes that are complex and badly defined. Most fuzzy controllers developed till now have been of the rule-based type (Driankov et al. 1993), where the rules in the controller attempt to model the operator´s response to particular process situations. An alternative approach uses fuzzy or inverse fuzzy model in process control (Babuška et al. 1995), (Jang 1995) because it is often much easier to obtain information on how a process responds to particular inputs than to record how, and why, an operator responds to particular situations. A review of the work on fuzzy control has been presented by Lee (Lee 1990).

Design of a simple fuzzy controller can be based on a three-step design procedure that builds on PID control: start with a PID controller, insert an equivalent, linear fuzzy controller and make it gradually nonlinear.

A fuzzy controller (Fig. 1) can include empirical rules, and that is especially useful in operator controlled plants. Take e.g. a typical fuzzy controller:

- if error is negative and change in error is negative then output is negative big
- if error is negative and change in error is zero then output is negative medium.

The collection of rules is called a rule base. The computer is able to execute the rules and compute a control signal depending on the measured inputs error and change in error. The inputs are most often hard or crisp measurements from some measuring equipment. A dynamic controller would have additional inputs, for example derivatives, integrals, or previous values of measurements backwards in time.

The block fuzzification converts each piece of input data to degrees of membership by a lookup in one or several membership functions. The rules may use several variables both in the condition and the conclusion of the rules. Basically a linguistic controller contains rules in the if-then format, but they can be presented in different formats.

The resulting fuzzy set must be converted to a number that can be sent to the process as a control signal. This operation is called defuzzification. There are several defuzzification methods. Output scaling is also relevant.

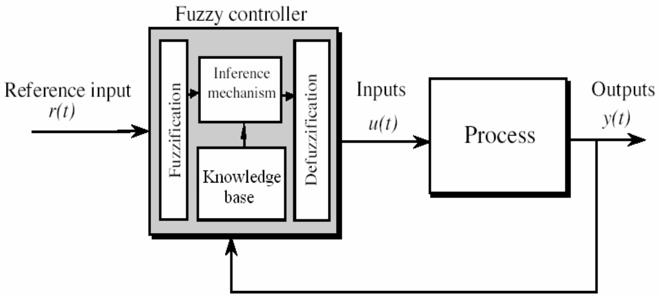


Fig. 1. Fuzzy control (Passino and Yurkovich 1998)

3. ADAPTIVE NETWORK BASED FUZZY INFERENCE SYSTEM (ANFIS)

The output sets can often be linear combinations of the inputs, or even a function of the inputs. The developed Fuzzy Logic Toolbox for the software package Matlab implements one of the hybrid schemes known as the ANFIS. ANFIS represents a Sugeno-type fuzzy system. Suppose the rule base of a Sugeno - Takagi fuzzy system as follows (Nauck et al. 1977), (Takagi et al. 1985):

$$\begin{gathered}\text{if } x_1 \text{ is } A_i \text{ and } x_2 \text{ is } B_i \\ \text{then } y = p_i x_1 + q_i x_2 + r_i,\ i = 1, ..., N \end{gathered} \tag{1}$$

The if-parts (antecedents) of the rules describe fuzzy regions in the space of input variables error e, its derivative de and the then-parts (consequents) are functions of the inputs, usually linear with consequent parameters p_i, q_i, r_i, y is an output variable, A_i, B_i are fuzzy sets.

ANFIS represents a useful neural network approach for the solution of function approximation problems. Data driven procedures for the synthesis of ANFIS networks may be based on the subtractive clustering technique (Chiu 1994) of the input-output space of a training set of numerical samples of the unknown function to be approximated.

In the ANFIS architecture, FIS is described in a layered, feedforward network structure (Fig. 6.). The parameters in layer 1 are called premise parameters and they are adjustable. The second layer represents the *T*-norm operators that combine the possible input membership grades in order to compute the firing strength of the rule. In the basic ANFIS method these parameters are not adjustable. The third layer implements a normalisation function to the firing strengths producing normalised firing strengths. The fourth layer represents the consequent parameters that are adjustable. The fifth layer represents the aggregation of the outputs performed by weighted summation. This is not adjustable.

3.1 Subtractive clustering

Subtractive clustering method is a method which extracts rules from supplied input-output training data. The idea of fuzzy clustering is to divide the data space into fuzzy clusters, each representing one specific part of the system behavior. After projecting the clusters onto the input space, the antecedent parts of the fuzzy rules can be found. The consequent parts of the rules can then be simple functions. In this way, one cluster corresponds to one rule of the TSK model. Several clustering methods are well known (Chiu 1994), (Yager and Filev 1994).

Let us consider a collection of n data points $\{x_1, x_2, ..., x_n\}$ in an M dimensional space. Each data point is a candidate for cluster centers, a density measure at data point $\mathbf{x}_i$ is defined as

$$P_k = \sum_{j=1}^{N} exp\left(-\alpha\left\|x_k - x_j\right\|^2\right) \tag{2}$$

where $\alpha = \dfrac{\gamma}{(r_a)^2}$, P_k is the new potential-value of each examined point, α is the weight between i-data to j-data, x is the data point, γ is variables (commonly set 4), r_a is a cluster radius, it is a positive constant that represents the radius of data neighborhood.

A data point will have a high density value if it has many neighboring data points. The first cluster center x_{c1} is chosen as the point having the largest density value P_{c1}. Next, the density measure of each data point x_i is revised as follows:

$$P'_k = P_k - P_{c1}\exp\left(-\beta\left\|x_k - x_{c1}\right\|^2\right) \tag{3}$$

where $\beta=\gamma/(r_b)^2$, $r_b = r_a\eta$, r_b is a positive constant which defines a neighborhood that has measurable reductions in density measure. Therefore, the data points near the first cluster center x_{c1} will have significantly reduced density measure. P_{c1} is the new potential-value data as cluster centre, β is the weight of i-data to cluster centre, η is the quash factor, usually set 1,5, r_i is the distance between cluster centre.

When the potential of all data points have been revised according to (3), the data point with highest remaining potential is selected as the second cluster center. We then further reduce the potential of each data point according to their distance to the second cluster center. The process is repeated until the potential of the points reaches the stopping criterion $P'_k < \varepsilon P_{c1}$, where ε is the reject ratio, usually set 0,15.

4. SIMULATION AND RESULTS

4.1 *Chemical reactor*

Consider a continuous-time stirred tank reactor (CSTR) with the first order irreversible parallel exothermic reactions according to the scheme $A \xrightarrow{k_1} B$, $A \xrightarrow{k_2} C$, where B is the main product and C is the side product. The dynamic mathematical model of the reactor is obtained by mass balances of reactants, enthalpy balance of the reactant mixture and enthalpy balance of the coolant. Assuming ideal mixing in the reactor and other usual simplifications (Ingham et al. 1994), (Vasičkaninová and Bakošová 2006), the simplified nonlinear dynamic mathematical model of the chemical reactor consists of five differential equations

$$\frac{dc_A}{dt} = \frac{q}{V}c_{Av} - \frac{q}{V}c_A - k_1c_A - k_2c_A \tag{4}$$

$$\frac{dc_B}{dt} = \frac{q}{V}c_{Bv} - \frac{q}{V}c_B + k_1c_A \tag{5}$$

$$\frac{dc_C}{dt} = \frac{q}{V}c_{Cv} - \frac{q}{V}c_C + k_2c_A \tag{6}$$

$$\frac{dT}{dt} = \frac{q}{V}T_v - \frac{q}{V}T - \frac{Ak}{V\rho c_p}\left(T - T_c\right) - \frac{h_1k_1 + h_2k_2}{\rho c_p}c_A \tag{7}$$

$$\frac{dT_c}{dt} = \frac{q_c}{V_c}T_{vc} - \frac{q_c}{V_c}T_c + \frac{Ak}{V_c\rho_c c_{pc}}\left(T - T_c\right) \tag{8}$$

with initial conditions $c_A(0)$, $c_B(0)$, $c_C(0)$, $T(0)$, $T_c(0)$. The reaction rate coefficients are non-linear functions of the reaction temperature being defined by the Arrhenius relations

$$k_i = k_{i0}e^{-\frac{E_i}{RT}},\ i = 1,2 \tag{9}$$

Here, c are concentrations, T are temperatures, V are volumes, ρ are densities, c_p are specific heat capacities, q are volumetric flow rates, h are reaction enthalpies, A is the heat transfer area, k_i is the heat transfer coefficient, k_{i0} is the pre-exponential factor, E is the activation energy and R is the universal gas constant. The subscript c denotes the coolant and the superscript s denotes the steady-state values in the main operating point.

The values of constant parameters and steady-state inputs of the chemical reactor are summarized in Table 1. Model uncertainty of the over described reactor follows from the fact that there are two physical parameters in this reactor, the reaction enthalpies, which values are known within following intervals (Table 2). The nominal values of these parameters are mean values of theirs intervals.

Table 1. Constant parameters and steady-state inputs of the chemical reactor

Variable	Unit	Value
q	m^3min^{-1}	0,015
V	m^3	0,23
V_C	m^3	0,21
ρ	kg m^{-3}	1020
ρ_C	kg m^{-3}	998
c_p	kJ kg^{-1} K^{-1}	4,02
c_{pc}	kJ kg^{-1} K^{-1}	4,182
A	m^2	1,51
k	kJ m^{-2} min^{-1} K^{-1}	42,8
k_{10}	min^{-1}	$1{,}55\times10^{11}$
k_{20}	min^{-1}	$4{,}55\times10^{25}$
E_1/R	K	9850
E_2/R	K	22019
c_{Av}	kmol m^{-3}	4,22
c_{Bv}	kmol m^{-3}	0
c_{Cv}	kmol m^{-3}	0
T_v	K	328

T_{vc}	K	298
$q^s{}_c$	m_3 min^{-1}	0,004
T^s	K	363,61
$T^s{}_c$	K	350,15
$c^s{}_A$	kmol m^{-3}	0,4915
$c^s{}_B$	kmol m^{-3}	2,0042
$c^s{}_C$	kmol m^{-3}	1,7243

Table 2. Uncertain parameters of the chemical reactor

Variable	Unit	Value
$-h_{1min}$	kJ kmol^{-1}	8,4×10^4
$-h_{1max}$	kJ kmol^{-1}	8,8×10^4
$-h_{2min}$	kJ kmol^{-1}	1,62×10^4
$-h_{2max}$	kJ kmol^{-1}	2,02×10^4

4.3 Neuro-fuzzy controller of the chemical reactor

In this paper, ANFIS and subtractive clustering method were used to design fuzzy controller. The design procedure is conducted in two stages: first subtractive clustering is applied to extract fuzzy model from experimental data; then ANFIS is applied to improve the fuzzy model performance.

Sugeno-type neuro-fuzzy inference system was generated in the form:

$$\begin{aligned} &if\ \ e \text{ is } A_i \text{ a } \frac{\mathrm{d}e}{\mathrm{d}t} \text{ is } B_i \text{ a } \int e \text{ is } C_i \\ &then\ \ f_i = p_i\, e + q_i \frac{\mathrm{d}e}{\mathrm{d}t} + r_i \int e + s_i, \quad i = 1,\ldots,8 \end{aligned} \qquad (10)$$

where e is the control error, p_i, q_i, r_i, s_i are consequent parameters. The symmetric Gaussian function (*gaussmf* in MATLAB) (11) was chosen as the membership function. The Gaussian function μ depends on two parameters σ and c as it is seen in (8), where x represents e, de/dt *or* ∫e.

$$\mu(x;\, \sigma,\, c) = e^{\left[\frac{-(x-c)^2}{2\sigma^2}\right]} \qquad (11)$$

The parameters σ and c for *gaussmf* are listed in the Table 3. For obtaining of these parameters, it was necessary to have the input data sets. These data were obtained by simulation of experimental PID controllers. The consequent parameters in the control input rule (10) are listed in Table 4. Figure 2 demonstrates the Takagi-Sugeno fuzzy inference system. Figure 3 shows the structure of Anfis.

Table 3. Parameters of the Gaussian curve membership functions

e		de		$\int e$	
σ_i	c_i	σ_i	c_i	σ_i	c_i
2,91	-4,72	0,44	-0,37	23,9	-56,33
2,95	2,71	0,50	0,79	23,9	0,0013

Table 4. Consequent parameters

p_i	q_i	r_i	s_i
-3,1×10^{-3}	-1,6×10^{-4}	-6,7×10^{-3}	8,3×10^{-3}
-3,1×10^{-3}	-2,6×10^{-4}	-5,4×10^{-3}	2,4×10^{-3}
-2,9×10^{-3}	-1,5×10^{-4}	-6,1×10^{-3}	5,6×10^{-3}
-3,3×10^{-3}	-2,5×10^{-4}	-7,0×10^{-3}	3,9×10^{-3}
-3,2×10^{-3}	-2,6×10^{-4}	-6,7×10^{-3}	2,3×10^{-3}
-3,3×10^{-3}	-2,2×10^{-4}	-7,0×10^{-3}	4,8×10^{-3}
-3,1×10^{-3}	-2,6×10^{-4}	-6,6×10^{-3}	3,4×10^{-3}
-3,3×10^{-3}	-2,2×10^{-4}	-6,3×10^{-3}	4,0×10^{-3}

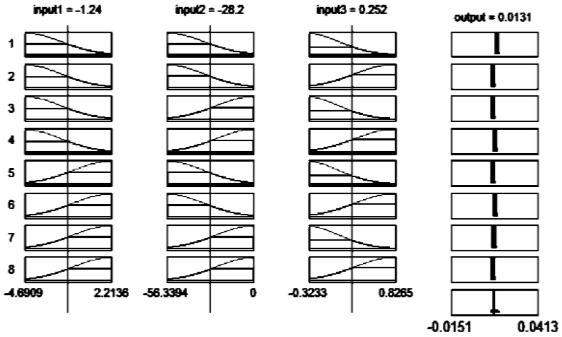


Fig. 2. Fuzzy inference system

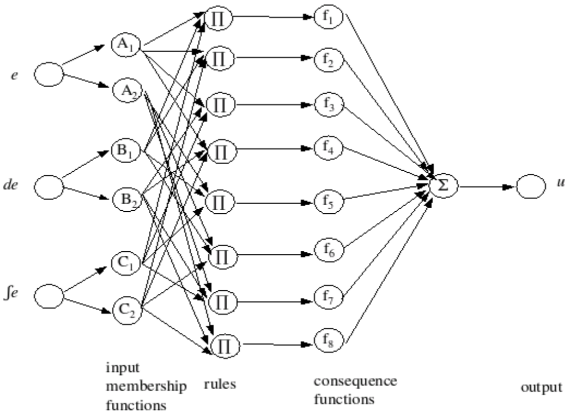


Fig. 3. Structure of Anfis

4.4 Control of the chemical reactor

The reactions in the described reactor are exothermic ones and the heat generated by the chemical reactions is removed by the coolant in the jacket of the tank. The measured output is temperature of the reaction mixture T, the coolant flow rate q_c is chosen as the control input. The control objective is to keep the temperature of the reacting mixture close to a desired value.

The steady state behaviour of the chemical reactor with nominal values and also with 4 combinations of minimal and maximal values of 2 uncertain parameters was studied at first. The maximal concentration of the main product B is obtained by temperature T=355 K (Fig. 4).

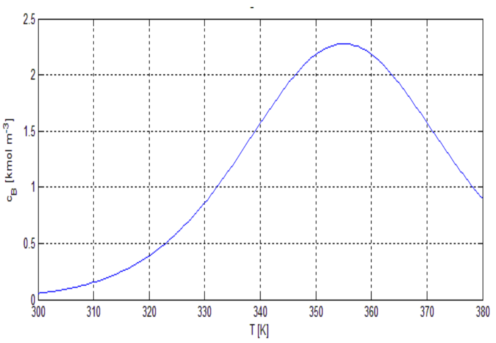


Fig. 4. Concentration of the main product B in the dependence on the *T*

The open-loop behaviour of the reactor was also studied. Simulation results obtained for the nominal model and also for 4 vertex systems are shown in Figure 5, 0 – nominal system, 1 – h_{1min}, h_{2min}, 2 – h_{1max},, h_{2max}, 3 – h_{1max}, h_{2min}, 4 – h_{1min}, h_{2max}.

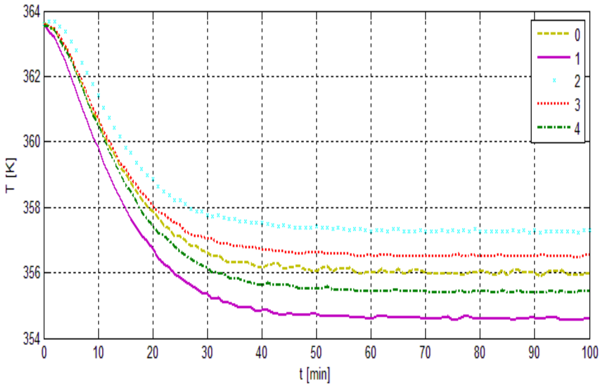


Fig. 5. Open-loop response of the CSTR: 0 - nominal system, 1, 2, 3, 4 - vertex systems

In Figure 6 the neuro-fuzzy control of the reacting mixture temperature and the reference trajectory obtained for the nominal model and for 4 vertex systems are shown. The control inputs are presented in Figure 7. The controller is fast and the overshoots are minimal.

In praxis, it is necessary to work with noisy signals, the white noise was added to the controlled output. Figures 8, 9 present the simulation results of the fuzzy control of the chemical reactor in the case when disturbances affect the controlled process. Disturbances were represented by temperature changes in the feed temperature of the reaction mixture. Following disturbances were loaded: *T* decreased by 5 K at t=200 min and increased by 10 K at t=200 min.

The neuro-fuzzy PID controller attenuates disturbances very fast and the overshoots caused by disturbances are minimal.

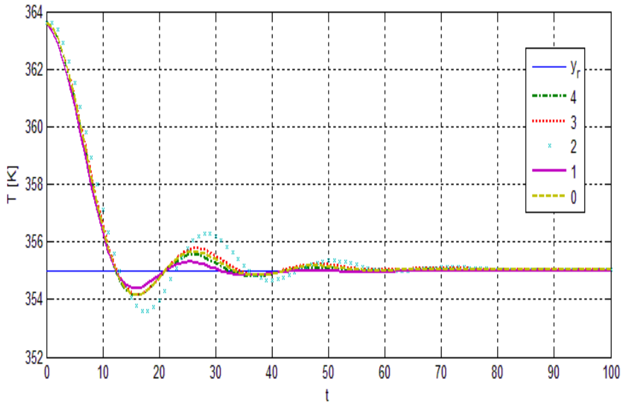


Fig. 6. Control of the CSTR: 0 - nominal system, 1, 2, 3, 4 - vertex systems

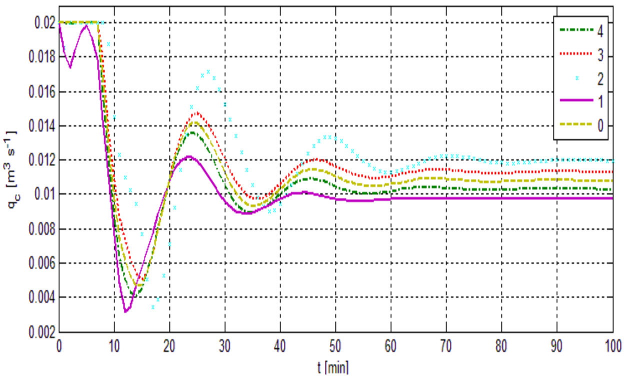


Fig. 7. Control inputs to the CSTR: 0 - nominal system, 1, 2, 3, 4 - vertex systems

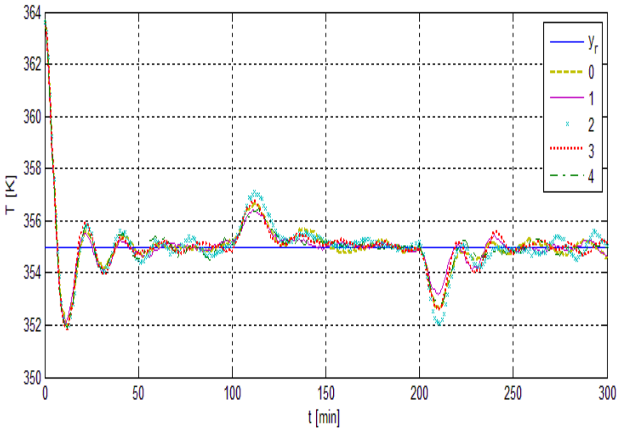


Fig. 8. Control of the CSTR with disturbances and noisy signals: 0 - nominal system, 1, 2, 3, 4 - vertex systems

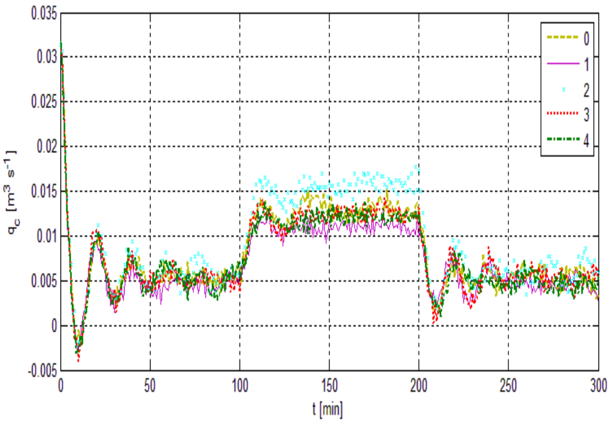


Fig. 9. Control inputs to the CSTR with disturbances and noisy signals: 0 - nominal system, 1, 2, 3, 4 - vertex systems

5. CONCLUSIONS

In this paper, the neuro-fuzzy control is applied to the exothermic CSTR with uncertain parameters. Simulations confirmed that robust neuro-fuzzy controllers can be successfully used for control of CSTRs with uncertainties and disturbances, even though CSTRs are very complicated systems from the control point of view. All simulations were done using MATLAB.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0537/10, 1/0095/11.

6 REFERENCES

Antonelli, R., A. Astolfi, A. (2003). Continuous Stirred Tank Reactors: Easy to Stabilise? *Automatica*. **Vol. 39.** No. 10. pp. 817-1827.

Babuška, R., J. Sousa, H.B. Verbruggen. (1995). Model-Based Design of Fuzzy Control Systems. *Proceedings of the EUFIT´95*. 837-841

Backer, E. (1995). Computer-assisted Reasoning in Cluster Analysis. New York. Prentice Hall.

Bequette, B.W. (1991). Nonlinear control of chemical processes: a review. *Industrial Engineering Chemical Research*. **30**. 1391–1413.

Bezdek, J. and S. Pal. (1992). Fuzzy Models for Pattern Recognition. New York. IEEE Press.

Dote, Y. and S.J. Ovaska. (2001). Industrial applications of soft computing: a review. *Proceedings of the IEEE*. pp. 1243-1265.

Doyle, J. C. and G. Stein. (1981). Multivariable Feedback Design - Concepts for a Classical Modern Synthesis. *IEEE Transactions on Automatic Control*. 26. 4–16.

Driankov, D., H. Hellendom. (1993). An Introduction to Fuzzy Control. Springer Verlag. London.

Gerhard, J., M. Monningmann and W. Marquardt. (2004). Robust stable nonlinear control and design of a CSTR in a large operating range. In *Proceedings of the 7th International Symposium on Dynamics and Control of Process Systems*. Massachusetts. IFAC. pp. 92.

Hayward, G. and V. Davidson. (2003). Fuzzy Logic Applications. *Analyst*. **Vol.128**. pp. 1304-1306.

Ingham, J., I.J. Dunn, E. Heinzle and J.E. Přenosil. (1994). Chemical Engineering Dynamics. VCH Verlags-gesellschaft. Weinheim.

Jain A. and R. Dubes. (1988). Algorithms for Clustering Data. Englewood Cliffs, Prentice Hall.

Jang, R.J.-S. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference System. *IEEE Trans. Syst. Man. Cybern.*, **23** (3). pp. 665-684.

Jang, R.J.-S., C.T. Sun. (1995). Neuro-Fuzzy Modeling and Control. *The Proceedings of the IEEE* **83**. 378-406.

Jang, R.J.-S., C.T. Sun and E. Mizutani. (1997). Neuro-Fuzzy and Soft Computing; Computational Approach to Learning and Machine Intelligence. Prentice Hall.

Lee, C. C. (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controllers. *IEEE Trans. Syst. Man. Cybern.*, SMC **20**, p. 404.

Luyben, W.L. (2007). Chemical reactor design and control. John Wiley & Sons. New Jersy. ISBN 978-0-470-09770-0

Molnár, A., J. Markoš and Ľ. Jelemenský. (2002). Accuracy of Mathematical Model with Regard to Safety Analysis of Chemical Reactors. *Chemical Papers*. **56**. 357–361.

Nauck, D., F. Klawonn and R. Kruse. (1977). Foundations of neuro-fuzzy systems. John Wiley & Sons. Great Britain.

Passino, K.M., S. Yurkovich. (1998). Fuzzy Control. Addisson Wesley. Ed. Ohio.

Peri, V.M. and D. Simon. (2005). Fuzzy Logic Control for an Autonomous Robot. North American Fuzzy Information Processing Society. NAFIPS 2005 Annual Meeting. pp. 337- 342.

Shapiro, A.F. (2004). Fuzzy Logic in Insurance, Insurance: Mathematics and Economics. **Vol. 35.** No.2. pp. 399-424.

Takagi, K. and M. Sugeno. (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Syst. Man. Cybern*. **15**. pp. 116-132.

Tlacuahuac, A.F., J. Alvarez, E.S. Guerra and G. Oaxaca. (2005). Optimal Transition and Robust Control Design for Exothermic Continuous Reactors. *AIChE Journal*. **Vol. 51**. no. 3. pp. 895-908.

Ying, H. (2000). Fuzzy Control and Modeling, Analytical Foundations and Applications. USA: Institute of Electrical and Electronic Engineers Inc.

Vasičkaninová, A. and M. Bakošová. (2006). Fuzzy modelling and identification of the chemical technological processes. In *Proc. 7. Int. Scientific-Tehnical Conf. Process Control 2006*. University of Pardubice. Kouty nad Desnou.

Zames, G. and B.A. Francis. (1983). Feedback, Minimax Sensitivity and Optimal Robustness. *IEEE Transactions on Automatic Control*. **28**. pp. 586–601.

Transplant Evolution for Optimization of general Controllers

Jindřich Petrucha
European Polytechnical Institute
Kunovice,
Osvobození 699, 686 04 Kunovice,
Czech Republic
e-mail:petrucha@edukomplex..cz

Pavel Ošmera
Institute of Automation and Computer
Science
Brno University of Technology
Faculty of Mechanical Engineering
Brno, Czech Republic
osmera@fme.vutbr.cz

Milos Seda
Institute of Automation and Computer
Science
Brno University of Technology
Faculty of Mechanical Engineering
Brno, Czech Republic
seda@fme.vutbr.cz

ABSTRACT

This paper describes a new method of evolution that is named Transplant Evolution (TE). None of the individuals of the transplant evolution contains genotype. Each individual of the transplant evolution contains only phenotype. Reproduction methods as crossover and mutation work and store only the phenotype. The hierarchical structure of grammar-differential evolution that is used for finding optimal structures and parameters of general controllers is described.

Categories and Subject Descriptors

D.3.2

General Terms

Algorithms, Design

Keywords

Transplant evolution, grammatical-differential evolution, object trees, hierarchical structures, algebraic reducing of trees, crossover by linking.

1. INTRODUCTION

The aim of this paper is to describe a new optimization method that can create control equations of general regulators. For this type of optimization a new method was created and we call it Two-Level Transplant Evolution (TLTE). This method allowed us to apply advanced methods of optimization, for example direct tree reducing of tree structure of control equation. The reduction method was named Arithmetic Tree Reducing (ART). For optimization of control equations of general controllers is suitable combine two evolutionary algorithms. Main goal in the first level of TLTE is the optimization of structure of general controllers. In the second level of TLTE the concrete parameters are optimized and the unknown abstract parameters in structure of equations are set. The method TLTE was created by combination of Transplant Evolution method (TE) [1,2,3,8,9,10] and Differential Evolution method (DE) [7]. The Transplant Evolution (TE) optimizes structure of solution with unknown abstract parameters and the DE optimizes the parameters in this structure. The parameters are real numbers. The real numbers are not easy find directly in TE without DE. For evaluation of quality of found control equation are described new methods, which allow us evaluate their quality. It can be used in the case when the simulation of control process cannot be finished. In results are shown some practical application. In all results we received the control equation that reached better quality of control process, than classical PSD controllers and Takahashi`s modification of PSD controller.

2. THE PRESENTATION OF OBJECT TREE STRUCTURES

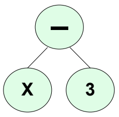


Fig. 1. Function block

The phenotype representation of the individual is stored in the object tree structure. Each of nodes in the tree structure, including the sub-nodes, is an object that is specified by a terminal symbol and the type of terminal symbols. All nodes are independent and correctly defined mathematical functions that can be calculated, e.g. the function x-3, shown on Fig. 1, is a tree structure containing a functional block (sub-tree).

Creating the object tree is a key part of GEOS, which this method differs from other evolutionary algorithms. When the object tree is generated, similar methods to a traditional grammatical evolution are used. But the GEOS does not store the genotype, because the production rules are selected by randomly generated genes that are not saved in chromosomes of individuals. The final GEOS's individual contains only phenotype expressed in an object tree structure.

The algorithm of GEOS uses a generative grammar [4,5,6] whose translation process starts from the initial symbol *S* and continues randomly with using the rules of defined grammar [2]. The basic procedure of the translation algorithm is shown on Fig. 2 where is explain to why is unnecessary to store the genotype.

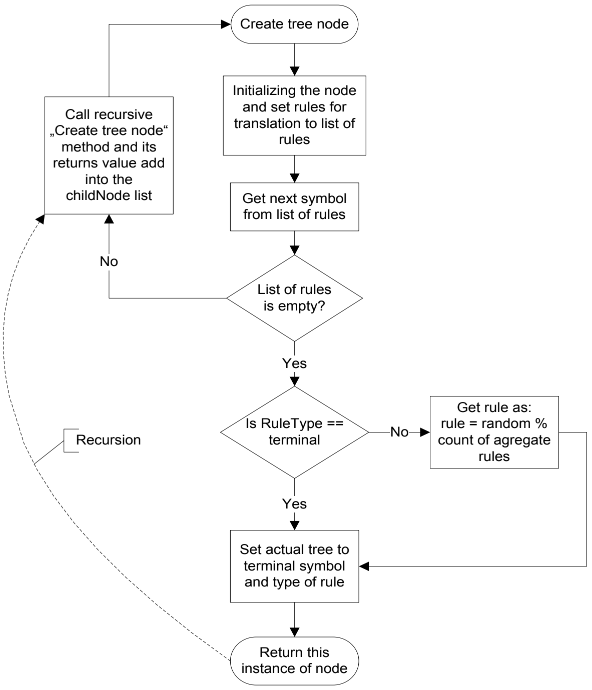

Create tree node
Initializing the node and set rules for translation to list of rules
Get next symbol from list of rules
List of rules is empty?
No
Yes
Call recursive „Create tree node“ method and its returns value add into the childNode list
Is RuleType == terminal
No
Get rule as: rule = random % count of agregate rules
Yes
Set actual tree to terminal symbol and type of rule
Return this instance of node
Recursion

Fig. 2 Flowchart creation of object tree

3. CROSSOVER

The crossover is a distinctive tool for genetic algorithms and is one of the methods in evolutionary algorithms that are able to acquire a new population of individuals. For crossover of object trees can be used following methods:

Crossover the parts of object trees (sub-trees)

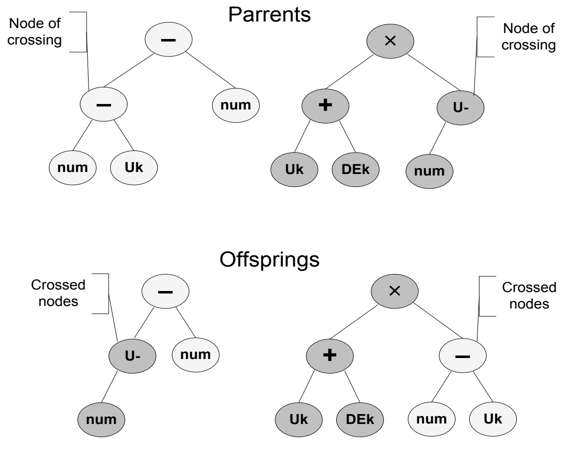


Fig. 3. Classical Crossover (CC)

The method of crossover object trees is based on the selection of two parents from the population and changing each other part of their sub-trees. For each of the parents cross points are randomly selected and their nodes and sub-trees are exchanged. This is the principle of creating new individuals into subsequent population as is shown on Fig. 3.

Crossover by linking trees or sub-trees

This method, as well as the previous one, is based on the crossover of two parents who are selected from the previous population. But the difference is in the way how the object trees are crossed. This method, unlike the previous one, does not exchange two randomly selected parts of the parents but parts of individuals are linked together with new and randomly generated node. This node will represent a new root of the tree structure of the individual. This principle is shown on Fig. 4.

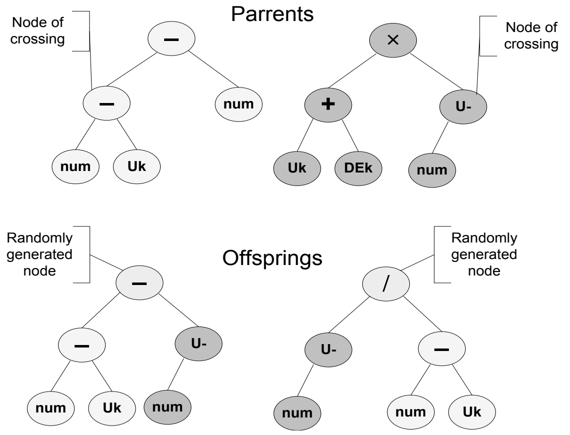


Fig.4. Crossover by linking method

4. MUTATION

Mutation is the second of the operators to obtain new individuals. This operator can add new structures, which are not included in the population so far. Mutation is performed on individuals from the old population. In the selected individual are randomly chosen nodes which are then subjected to mutation. The mutation operator can be subdivided into two types:

- Non-structural Mutation (NM)
- Structural Mutation (SM)

Non-structural Mutation (NM)

Non-structural mutations do not affect the structure of already generated individual. In the individual who is selected for mutation, chosen nodes of object sub-tree are further subjected to mutation. The mutation will randomly change chosen nodes, whereas used grammar is respected. For example it means that mutated node, which is a function of two variables (i.e. + - × ÷) cannot be changed by node representing function of one variable or only a variable, etc. see Fig. .

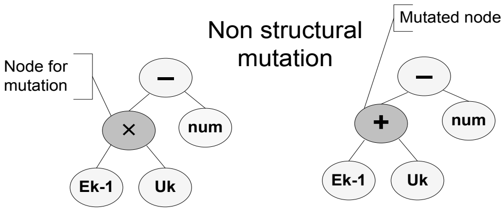


Fig. 5. Nonstructural mutation

Structural Mutation (SM)

Structural mutations, unlike non-structural mutations, affect the tree structure of individuals. Changes of the sub-tree by extending or shortening its parts depend on the method of structural mutations. Structural mutation can be divided into two types: Structural mutation which is extending an object tree structure (ESM) and structural mutation which is shortening a tree structure (SSM). This type of mutation operator can be subdivided into two types:

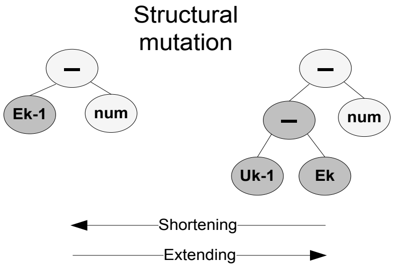


Fig. 6. Structural mutation

Extending Structural Mutation (ESM)

In the case of the extending mutation, a randomly selected node is replaced by a part of the newly created sub-tree that respects the rules of defined grammar (see fig. 3). This method obviously does not always lead to the extension of the sub-tree but generally this form of the mutation leads to extension of sub-tree. (see Fig.).

Shortening Structural Mutation (SSM)

Conversely the shortening mutation replaces a randomly selected node of the tree, including its child nodes, by node which is described by terminal symbol (i.e. a variable or a number). This type of mutation can be regarded as a method of indirectly reducing the complexity of the object tree (see Fig.).

The complexity of the tree structure can be defined as the total number of objects in the tree of individual.

5. DIRECT TREE REDUCTION

The minimal length of an object tree is often one of the factors required in the optimal problem solution. This requirement can be achieved in several ways:

- By penalizing the part of the individual fitness which contains a complex object tree,
- Method of targeted structural mutation of individual (see SSM),
- The direct shortening of the tree using algebraic adjustments - algebraic reducing tree (ART).

The last-mentioned method can be realised by the GEOS, where all of individuals does not contain the genotype, and then a change in the phenotype is not affected by treatment with genotype. The realisation of above mentioned problem with individual, which use genotype would be in this case very difficult. This new method is based on the algebraic arrangement of the tree features that are intended to reduce the number of functional blocks in the body of individuals (such as repeating blocks "unary minus", etc.). The method described above is shown on Fig. and Fig. .

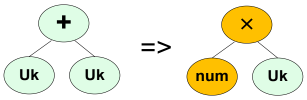


Fig. 7. ART – substitution of nodes

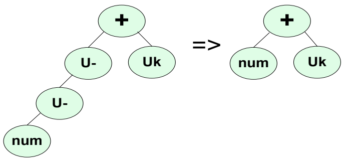


Fig. 8. ART – reduction multiple unary minus

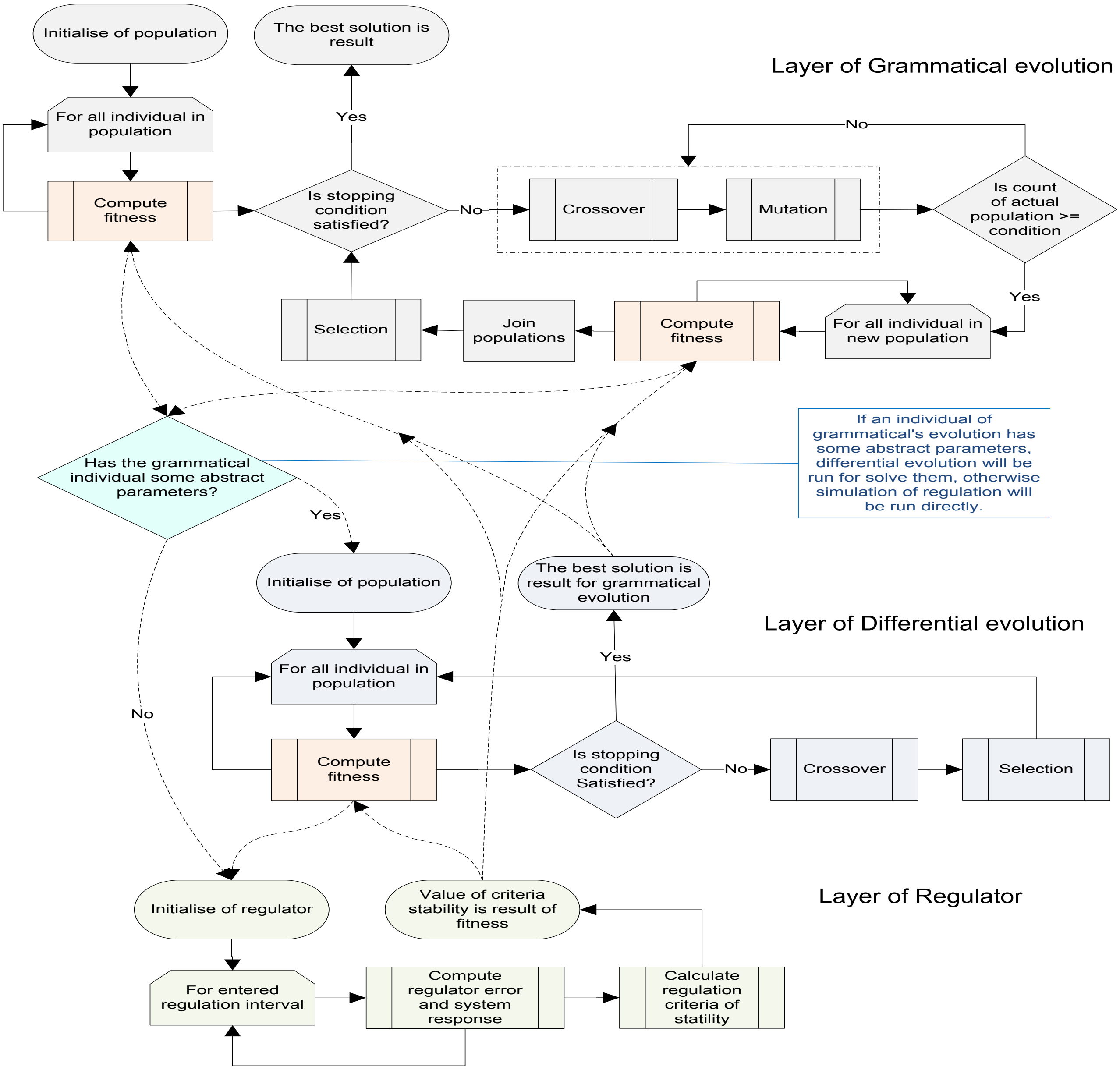

Initialise of population
The best solution is result
Layer of Grammatical evolution
For all individual in population
Yes
No
Compute fitness
Is stopping condition satisfied?
No
Crossover
Mutation
Is count of actual population >= condition
Yes
Selection
Join populations
Compute fitness
For all individual in new population
If an individual of grammatical's evolution has some abstract parameters, differential evolution will be run for solve them, otherwise simulation of regulation will be run directly.
Has the grammatical individual some abstract parameters?
Yes
Initialise of population
The best solution is result for grammatical evolution
Layer of Differential evolution
Yes
For all individual in population
No
Compute fitness
Is stopping condition Satisfied?
No
Crossover
Selection
Initialise of regulator
Value of criteria stability is result of fitness
Layer of Regulator
For entered regulation interval
Compute regulator error and system response
Calculate regulation criteria of statility

Fig. 9. Flowchart of TE (GDEOS) for controller

In view of the object tree complexity of the individual and also for subsequent crossover is preferable to have a function in the form $x = 3a$ than $x = a + a + a$, or more generally $x = n \times A$. Another example is the shortening of the function $x = -(-a)$, where is preferable to have the form $x = a$ (it is removing redundant marks in the object tree individual). The introduction of algebraic modifications of individual phenotype leads to the shorter result of the optimal solution and consequently to the shorter presentation of the individual, shortening the time of calculation of the function that is represented in object tree and also to find optimal solutions faster because of higher probability of crossover in the suitable points with higher probability to produce meaningful solutions. The essential difference stems from the use of direct contraction of trees, which leads to significantly shorter resulting structure than without using this method.

6. HIERARCHICAL STRUCTURE OF TE (GDEOS) FOR OPTIMISATION OF THE CONTROLLER

The hierarchical structure of the transplant evolution can be used for optimisation of the structure and parameters of a general controller. This structure contains three layers. First two layers (GE + DE) are contained in TE. Those two layers are used for optimisation of the structure and parameters of general controller. The third layer which is named layer of controller is used for computation of fitness in TE.

At the beginning of GDEOS an initial population is created (see Fig. 2) and then fitness of individuals is calculated. In the case of finding the optimal solution in the first generation, the algorithm is terminated, otherwise creates a new population of individuals by crossover and mutation operators, with the direct use of already created parent's object tree structures (it is analogy as transplantation of already created organs, without necessary know-ledge of DNA – "Transplant Evolution (TE)"). If the result of GDEOS needs some numerical parameters (for example *num* in (Weisser 2010)), the second level with Differential Evolution (DE) is used for optimization their parameter setting. The DE gives better results in finding optimal values of unknown numerical parameters that are expressed in the form of real numbers, then in the GE. Due to the use of GDEOS for optimization of controllers in the next stage of calculation of fitness is model of controller used which is represented by the equation in incremental form (recurrent algorithm). Quality of controller is determined depending on the type of criterial function (see equation 3). For fitness calculation are various criterial functions used. Basic criterion is linear control area, quadratic control area, linear or quadratic control area extended with overshoot, oscillation of action value of the controller.

The flowchart of TE (GDEOS) for a controller is shown on Fig.

7. RESULTS

The TE and TE + ART methods for optimization of equation for general controller were compared.

The resulting form of the recurrent equation of general controller without using the direct method shortening of the tree (ART) is following (equation 1):

$$\begin{aligned}&u_k=((((((E_k-(-((E_k+E_k))))\times3)\times2)+E_{k-3})-(-(2)))-(-(((((((E_{k-3}+((((((E_k\times3)\\&+(E_k+(((dE_{k-1}+E_k)\times2)\times1.63)))\times2)\times2)+(-(((E_{k-4}+(E_k-(-(3))))+E_{k-2}))))\\&\times3))+(-(((E_{k-4}+(E_k-E_k))+E_k-2))))\times2)+((dE_{k-1}+(4.47-((((E_k-(-\\&((((((((((E_k\times3)+(E_k+(((dE_{k-1}+E_k)\times2)\times2)))\times2)\times2)\times2)+(-(3.61)))+E_k)\\&+E_k)+(-((3+E_{k-2})))))))+((E_k\times2)+E_k))+E_k)-(-(((E_k+2)\times(3\times((E_{k-4}+2)\quad+E_{k-2}))))))))-((((E_k+((((E_k+2)+E_k)+E_k)+(E_k/(-((6.88-((dE_{k-1}+(1.79-E_{k-3}))-\\&2)))))))\ \times2)\times3)+(-(((E_{k-4}+((E_k+E_k)\times2))\ +E_k))))))+E_k)-(-(E_{k-2})))))\end{aligned} \tag{1}$$

The resulting form of the recurrent optimization algorithm in the case with using the direct method of contraction tree is following (equation 2):

$$u_k = (E_{k-3} - E_k \times 1.93) \times 33.97 + E_{k-1} + E_{k-2} \tag{2}$$

As you can see, the resulting lengths of recurrent equation of the general controller, is shorter in case of using TE + ATR then TE without ART.

Bellow is shown result of optimisation parameters of PSD controllers and optimisation of the structure and parameters of general controllers. The parameters of PSD controllers were optimised with using DE and structure and parameters of general controller were optimised with using TE + ART method.

The basic criterion of minimal integral control area was used as criterial function for optimisation of PSD or general controllers, (see equation 3)

$$J = \int_0^{\infty} |e(t) - e(\infty)|\, dt \tag{3}$$

On the Fig. and Fig. are result of optimisation of PSD controller and general controller to control the identical system with 5 second time delay.

On the Fig. is shown regulatory process of PSD controller. The parameters of PSD controller were optimised with using DE.

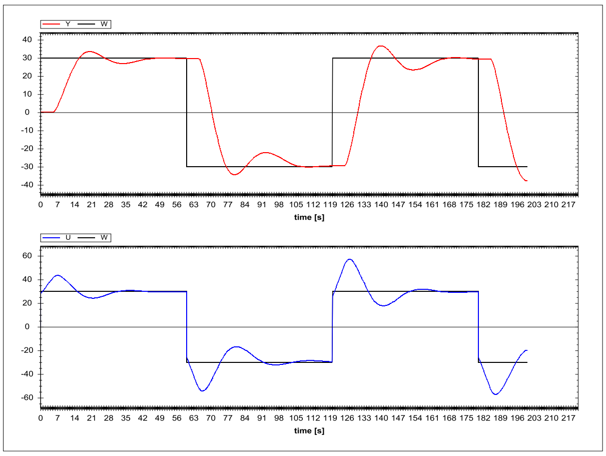


Fig.10. Regulatory process of PSD controller for second order system with 5s time delay

(Top figure shows the system response and on the bottom part of the figure is shown the action output of controller)

On the Fig. is shown regulatory process of general controller. The structure and parameters of this controller was optimised with using TE (GDEOS) + ART method. The equation of general controller is following (see equation 4).

$$u_k = E_k \times 23.01 + 10.91 \times E_{k-3} + E_{k-1} \times (-33.91) + U_{k-1} \tag{4}$$

On the Fig. and

Fig. are result of optimisation of PSD controller and general controller to control the identical system with 2 second time delay.

On the Fig. is shown regulatory process of PSD controller. The parameters of PSD controller were optimised with using DE.

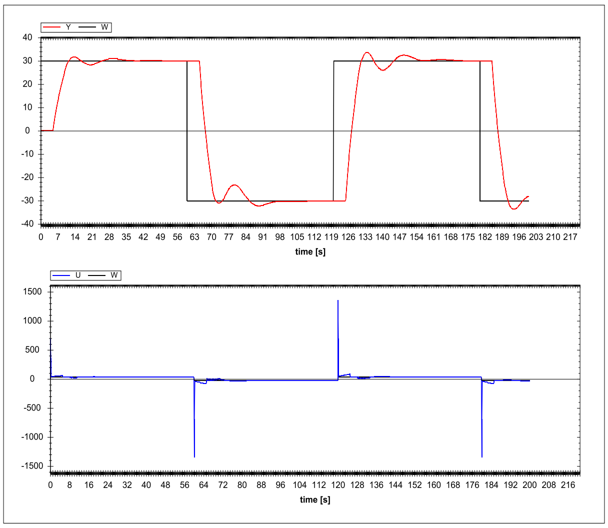


Fig. 11 Regulatory process of general controller for second order system with 5s time delay

(Top figure shows the system response and on the bottom

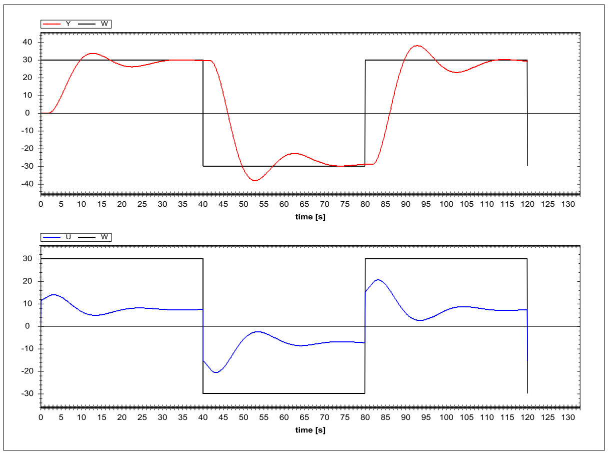


Fig. 12 Regulatory process of PSD controller for second order system with 2s time delay

(Top figure shows the system response and on the bottom part of the figure is shown the action output of controller)

On the Fig.13 is shown regulatory process of general controller. The structure and parameters of this controller was optimised with using TE (GDEOS) + ART method. The equation of general controller is following (see equation 5).

$$u_k = (32.55 \times E_{k-5}) + (58.79 \times E_k) + 7.08 + (-89.97 \times E_{k-2}) \qquad (5)$$

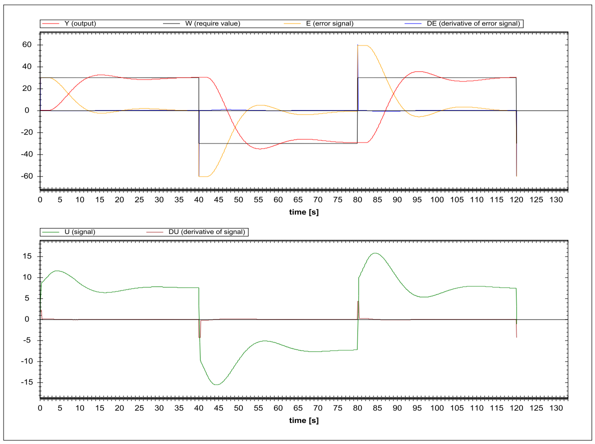


Fig. 13. Regulatory process of general controller for second order system with 2s time delay

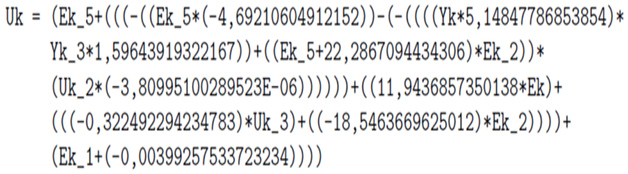
(Top figure shows the system response and on the bottom part of the figure is shown the action output of the controller)

We tested the TLTE method for optimization of recurrent equation of general controllers. There is some results of optimization for one following system:.

Integral system with transport delay

$$G_S(S) = \frac{5}{s(9s+1)} e^{-2s} \qquad (5)$$

In Fig.14 we compare 3 types of controllers. There is one PSD controller marked PSD_DE and two general controllers marked General_DE and General_TLTE. The curve marked PSD_DE is PSD controller. Parameters (Kr, Ti, Td) of this controller were optimized by Differential Evolution (DE). The curve marked General_DE is for general controller which has the control equation in PSD equation form, but parameters q0, q1, q2 were optimized directly by DE. The curve marked General_TLTE is for general controller with general control equation that was optimized by Two-Level Transplant Evolution (TLTE). As you can see, the best result gives the General_TLTE. In this case we receive the recurrent control equation with following form:



```
Uk = (Ek_5+(((-((Ek_5*(-4,69210604912152))-(-((((Yk*5,14847786853854)*
     Yk_3*1,59643919322167))+((Ek_5+22,2867094434306)*Ek_2))*
     (Uk_2*(-3,80995100289523E-06))))))+((11,9436857350138*Ek)+
     (((-0,322492294234783)*Uk_3)+((-18,5463669625012)*Ek_2))))+
     (Ek_1+(-0,00399257533723234))))
```

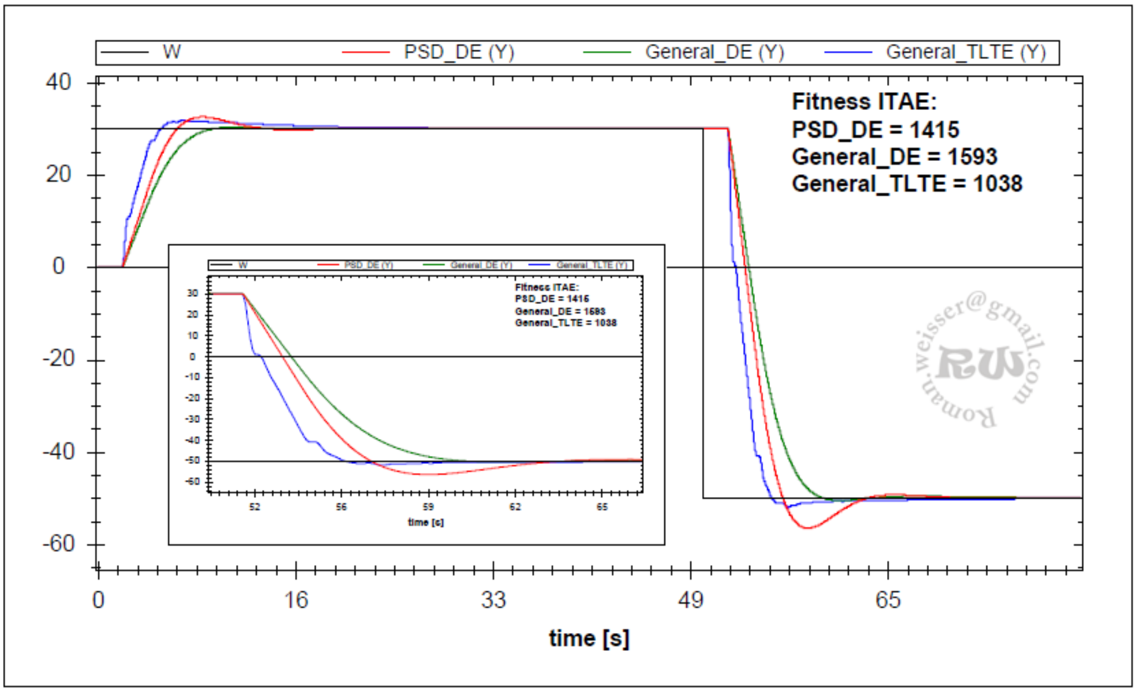


Fig.14. Step response for integration system with time delay

8. CONCLUSION

The Two-Level Transplant Evolution (TLTE) was successfully use for automatic generation of control programs of general controllers. We tested this algorithm on many problems, only one example was described in this paper. We hope that this new method of controller design will be use in practice, not only for simulation.

Although we are at early stages of experiments, but it seems that it is possible to use parallel grammatical evolution with backward processing to generate combinatorial logic circuits. The grammatical algorithm can be outperformed with algorithms, which are designed specifically for this purpose.

9. ACKNOWLEDGMENTS

This work has been supported by Czech Ministry of Education No: MSM 00216305529 Intelligent Systems in Automation and GA ČR No: 102/09/1668.

10. REFERENCES

[1] Koza J.R. 1992: Genetic Programming: On the Programming of Computers by Means of Natural Selection, The MIT Press

[2] Kratochvíl O. and Ošmera P. and Popelka O. 2009: Parallel grammatical evolution for circuit optimization, in Proc. WCECS, World Congress on Engineering and Computer Science, San Francisco, 1032-1040.

[3] Li Z. and Halang W. A. and Chen G. 2006: Integration of Fuzzy Logic and Chaos Theory; paragraph: Osmera P.: Evolution of Complexity, Springer, 527 – 578.

[4] O'Neill M. and Ryan C. 2003: Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language Kluwer Academic Publishers.

[5] O'Neill M. and Brabazon A. and Adley C. 2004: The Automatic Generation of Programs for Classification Problems with Grammatical Swarm, Proceedings of CEC, Portland, Oregon, 104 – 110.

[6] Piaseczny W. and Suzuki H. and Sawai H. 2004: Chemical Genetic Programming – Evolution of Amino Acid Rewriting Rules Used for Genotype-Phenotype Translation, Proceedings of CEC, Portland, Oregon, 1639 - 1646.

[7] Price K. 1996. Differential evolution: a fast and simple numerical optimizer, Biennial Conference of the North American Fuzzy Information Processing Society, NAFIPS, IEEE Press, New York, NY, 524-527.

[8] Rukovanský I. Optimization of the throughput of Computer Network Based on Parallel EA. In Proceedings of the World Congress on Engineering and Computer Science WCECS 2009, San Francisco, CA, Oct. 20-22, 2009, Vol. II, pp. 1038-1043

[9] Weisser R., Ošmera P., Matoušek R., Transplant Evolution with Modified Schema of Differential Evolution: Optimization Structure of Controllers. In International Conference on Soft Computing MENDEL. Brno : MENDEL, 2010.

[10] Weisser R., Ošmera P., Šeda, M., Kratochvíl, O. Transplant Evolution for Optimization of General Controllers. In European Conference on Modelling and Simulation. 24th. Kuala Lumpur (Malaysia) : ECMS 2010. s. 250 -- 260.

Tests of Various Types of Residuals in Regression Diagnostics

M. Javůrek I. Taufer

*Department of Process Control, Faculty of Electrical Engineering and Informatics,
The University of Pardubice, Studentská 95, 532 10 Pardubice.
e-mail: Milan.Javurek@upce.cz; Ivan.Taufer@upce.cz;*

Abstract: Approximation of experimental data by means of an analytical or general mathematical dependence is performed most frequently by the regression method using the least squares approach. The quality of curve fitting is evaluated on the basis of analysis of resulting set of residuals which, however, can be defined in various ways. This paper deals with suitability tests of the individual types from the standpoint of curve fitting quality of the regression dependence.

1. INTRODUCTION

Regression is one of most common and most favored approximation methods of experimental dependences. The principle consists in optimization (i.e. minimization) of the users function, most often in the form of the least squares, which expresses closeness of curve fitting of the regression and experimental dependence. The function fitted may be known in analytical form, where the parameters have direct physical meaning, or various types of mathematical dependences are used. The basic classification of regression methods is done according to the parameters of fitted dependences, i.e. linear regression and non-linear regression are known. While the linear regression is evaluated according to univocal formulas, the so-called normal equations, the course and results of non-linear regression are affected by a number of factors, such as initial assessment of parameters, the adopted optimizing method, interdependence of individual parameters etc. Therefore, the parameters found can be neither correct nor accurate, particularly in the cases where even their approximate values are unknown. The non-linear regression offers relatively few tools for verification of the found parameters. If several calculations are performed with different models, their comparison is carried out by means of Akaike Information Criterion (AIC), mean quadratic error of prediction (MEP), the value of users function (residual-square-sum, RSC) etc. (MELOUN, 2011). However, if we have only a single calculation, then one of the few available tools is the analysis of set of residuals. It is quite paradoxical that most PC programs (even the commercial ones. such as STATISTICA) do not include this analysis; only in algorithmic regime they provide the values of parameters with their standard deviations. No further verification is performed, and the curve fitting quality of regression dependence and experimental dependence cannot be evaluated.

If the conditions of application of regression method are fulfilled (the data do not exhibit heteroscedasticity, supernormality, multicollinearity, autocorrelation, outliers, the model is significant), then the set of residuals should exhibit normal distribution, which can be proved on the basis of calculated values of central moments, Pearson's test, sign tests and other tests. However, residuals can be defined in various ways, and their information abilities differ.

2. DEFINITION OF VARIOUS TYPES OF RESIDUALS

2.1 Classical Residuals $\hat{e}_i$ (MELOUN, 2011)

These residuals are defined as a difference between the calculated values and the experimental ones. They are correlated, do not have constant dispersion, and they need not correctly indicate deviated points.

2.2 Normalized Residuals $\hat{e}_{Ni}$ (MELOUN, 2011)

In this case the normalization consists in division of the value of classical residual by the value of standard deviation of the whole set. The set of residuals should have normal distribution with zero mean value and standard deviation is equal to one. The values higher/lower than the triple of standard deviations are considered as outliers.

However, the mathematical analysis shows that the dispersion D($\hat{e}_{Ni}$) = (1-H_{ii}) is neither constant nor unit, so the recommended elimination of the values exceeding the interval of the triple of standard deviation need not be correct.

2.3 Standardized Residuals $\hat{e}_{Si}$ (MELOUN, 2011)

They also should exhibit normal distribution with constant dispersion; they are defined as follows:

$$\hat{e}_{Si} = \frac{\hat{e}_i}{\hat{\sigma}\sqrt{1-H_{ii}}} \qquad (1)$$

where $\hat{\sigma}$ stands for standard deviation

H_{ii} are diagonal elements of projection matrix

Their properties are almost identical with the classical ones.

2.4 Jack-Knife Residuals $\hat{e}_{Ji}$ (MELOUN, 2011)

If in Eq. (1) we use, instead the overall standard deviation, its estimate obtained with omitting of the i-th point:

$$\hat{e}_{Ji} = \sqrt{\frac{n-m-1}{n-m-\hat{e}_{Si}}} \qquad (2)$$

where n stands for the number of measurements

m is the number of parameters determined

Under the assumption of normality of errors, these residuals exhibit Student distribution with n-m-1 degrees of freedom. These residuals are used for identification of outliers points.

2.5 Predicted Residuals $\hat{e}_{Pi}$ (MELOUN, 2011)

These are defined as follows:

$$\hat{e}_{Pi} = y_i - x_i\,\mathbf{b}_{(i)} = \frac{\hat{e}_i}{1-H_{ii}} \qquad (3)$$

where x is/are independent variable(s) and y is the dependent variable quantity

$\mathbf{b}_{(i)}$ are estimates of parameters obtained by the least squares method from all points except the i-th point

3. OTHER DIAGNOSTIC TOOLS

3.1 Cook's Distance D_i (MELOUN, 2011)

This, in fact, is the Euclidean distance between the vector of prediction of independent variable obtained by the least squares method and the same vector obtained with elimination of the i-th point. Cook's distance expresses the effects of the i-th point upon the estimates of parameters only. It is defined as follows:

$$D_i = \frac{\hat{e}_{Si}}{m}\,\frac{H_{ii}}{1-H_{ii}} \qquad (4)$$

3.2 Atkinson's Distance A_i (MELOUN, 2011)

This is used in order to increase the sensitivity of regression to extreme points. It is defined as follows:

$$A_i = \left|\hat{e}_{Ji}\right| \sqrt{\frac{n-m}{m}\,\frac{H_{ii}}{1-H_{ii}}} \qquad (5)$$

3.3 Distances of Likelihood

This quantity is the difference of logarithms of credibility function using all point and that obtained with elimination of the i-th point. If its value is higher than the quantile $\chi^2_{1-\alpha}(m+1)$ of distribution, then the given point is considered as influential.

3.4 Summary Characteristics of Properties of Whole Set of Residuals

Original data			
Type of residuals	Classical	Normalized	Standard.
Users function	0.0030	1.7373	29.9600
Arithm. mean	0.0000	-0.0430	0.3816
Stand. deviation	0.0101	0.2368	0.9236
Mom. coeff. of skew.	-0.2084	-3.0775	-1.2577
Mom. coeff. of curt.	0.0298	11.6552	2.3077
R-factor	0.0006	0.0136	0.0563
Type of residuals	Jack-Knife	Predicted	
Users function	32.2588	0.0035	
Arithm. mean	-0.0058	0.0000	
Stand. deviation	1.0369	0.0108	
Mom. coeff. of skew.	-0.2908	-0.1922	
Mom. coeff. of curt.	0.3215	-0.0634	
R-factor	0.0584	0.0006	
Other criteria			
AIC	-227.0000		
MEP	0.0001		
Heteroscedasticity	yes		
Normality	yes		
Autocorrelation	not		
Sign test	negat.		

Table 1. Characteristics of set of residuals of starting data

The following characteristics have also been used for determination of validity of the basic presumptions of application of the least squares method from the whole set of residuals (the classical residuals were always used):

- Cook–Weisberg's test of heteroscedasticity
- Jarque–Berr's test of normality
- Wald's test of autocorrelation
- Sign test

Description of these tests is somewhat complicated and time-consuming; therefore, see Ref. (MELOUN, 2011).

The above-mentioned diagnostic tools (except for 3.4) are ordinarily used only for detection of significant (extreme, outliers) points, which from the standpoint of overall view of the quality of curve fitting of regression model is of not very high significance. If the model is not suitable, then the search for these points has no meaning either. In most cases, the closeness of curve fitting is evaluated by means of classical residuals, but their information effectiveness is very small; the calculation of statistical moments expresses neither the outliers points not the trends in curve fitting. In this case, the tests given in section 3.4 come in useful and, furthermore, testing of suitability of the whole model by means of F-test, or Student's test of significance of the individual parameters. The two last mentioned tests have not been taken into account in this paper, since the model was known. Also Hamilton's R-factor has very good information ability:

$$R = \sqrt{\frac{RSC}{\sum y_i^2}} \qquad (6)$$

Classical residuals			
First point			
Multiple of characteristics of original set	1. point +1 s	1. point + 2s	1. point + 3s
Users function	0.9596	3.1891	11.1415
Arithmetic mean	–0.8662	2.4045	5.1910
St. deviation	0.9796	1.7858	3.3903
M. coeff. of skew.	1.2973	–12.1966	–19.1828
M. coeff. of curt.	4.7594	350.2452	655.3301
R-factor	0.9796	1.7858	3.3378
Other criteria			
AIC	–273.2000	–237.2000	–199.7000
MEP	0.0001	0.0004	0.0014
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.
Second point			
Multiple of characteristics of original set	1. point +1s, 2. point -1s	1. point +2s, 2. point -2s	1. point +3s, 2. point -3s
Users function	1.0754	3.6223	12.1170
Arithmetic mean	0.0000	2.8989	0.6816
St. deviation	1.0370	1.9032	3.4809
M. coeff. of skew.	2.0439	–9.8283	–17.4348
M. coeff. of curt.	7.9841	317.6662	606.9347
R-factor	1.0370	1.9032	3.4809
Other criteria			
AIC	–269.8000	–233.4000	–197.1000
MEP	0.0001	0.0005	0.0016
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.

Table 2. Various variants of calculation for classical residuals

Normalized residuals			
First point			
Multiple of characteristics of original set	1. point +1 s	1. point + 2s	1. point + 3s
Users function	0.4125	29.8402	112.5996
Arithmetic mean	0.4781	–3.6503	–7.2397
St. deviation	0.6469	5.5122	10.8850
M. coeff. of skew.	0.7344	–1.6724	–1.6877
M. coeff. of curt.	0.8122	2.3804	2.4561
R-factor	0.6422	5.4626	10.6110
Other criteria			
AIC	–273.2000	–237.2000	–199.7000
MEP	0.0001	0.0004	0.0014
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.
Second point			
Multiple of characteristics of original set	1. point +1s, 2. point -1s	1. point +2s, 2. point -2s	1. point +3s, 2. point -3s
Users function	0.8572	33.3538	121.3035
Arithmetic mean	0.6204	–3.3425	–6.7443
St. deviation	0.9342	5.8382	11.1266
M. coeff. of skew.	1.2930	–1.5192	–1.6224
M. coeff. of curt.	1.6985	2.1565	2.3071
R-factor	0.9259	5.7752	11.0136
Other criteria			
AIC	–269.8000	–233.4000	–197.1000
MEP	0.0001	0.0005	0.0016
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.

Table 3. Various variants of calculation for normalized residuals

If the value of R-factor is not higher than the uncertainty of measurement, then the curve fitting can be considered as a good one. However, the use of classical residuals could be replaced by another type, which is more sensitive and better reflects the deviation in the fitting of regression dependence. Hence in the subsequent text we will try to select the more suitable type out of the available types of residuals.

4. TESTING

The testing of individual types of residuals was performed on a simulated example of linear dependence. The values of dependent variable were calculated for the given values of independent variable and given parameters of straight line. The calculated values were loaded with errors exhibiting normal distribution. The obtained data were evaluated by linear regression, and the central moments of the set of residuals were calculated. In the next step, the value of dependent variable of the first point was increased stepwise by adding one, two and three multiples of standard deviation from the first set, and again the individual sets of data were evaluated by linear regression, and central moments of the sets of residuals were calculated. These values were referenced to the values of central moments of the original set – i.e. to find out to what extent the change of one point will make itself felt in the change of characteristics of the set of residuals. In subsequent step, also changes of the second point were added to the changes of the first point in the opposite direction as compared with the first point.

Apart from the tests described, we also carried out calculations with the sets in which the described changes had been realized in central part of the dependence; however, this

Standardized residuals			
First point			
Multiple of characteristics of original set	1. point +1 s	1. point + 2s	1. point + 3s
Users function	17.2142	1.0464	1.0595
Arithmetic mean	–0.0003	0.0145	0.0165
St. deviation	1.0810	1.1068	1.1312
M. coeff. of skew.	0.2081	–2.0917	–3.2168
M. coeff. of curt.	0.0453	4.7456	8.6059
R-factor	0.9991	1.0229	1.0293
Other criteria			
AIC	–273.2000	–237.2000	–199.7000
MEP	0.0001	0.0004	0.0014
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.
Second point			
Multiple of characteristics of original set	1. point +1s, 2. point -1s	1. point +2s, 2. point -2s	1. point +3s, 2. point -3s
Users function	1.0034	1.0487	1.0606
Arithmetic mean	–0.0024	0.0112	0.0139
St. deviation	1.0838	1.1080	1.1143
M. coeff. of skew.	0.3467	–1.6840	–2.9227
M. coeff. of curt.	0.1070	4.2868	7.9575
R-factor	1.0017	1.0241	1.0298
Other criteria			
AIC	–269.8000	–233.4000	–197.1000
MEP	0.0001	0.0005	0.0016
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.

Table 4. Various variants of calculation for standardized residuals

Jack-Knife residuals			
First point			
Multiple of characteristics of original set	1. point +1 s	1. point + 2s	1. point + 3s
Users function	1.0026	2.3131	9.0801
Arithmetic mean	1.0198	–21.0248	–70.0348
St. deviation	1.0013	1.5164	3.0392
M. coeff. of skew.	1.2606	–14.4374	–18.0003
M. coeff. of curt.	1.4778	64.8356	89.5911
R-factor	1.0013	1.5209	3.0133
Other criteria			
AIC	–273.2000	–237.2000	–199.7000
MEP	0.0001	0.0004	0.0014
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Second point			
Sign test	negat.	negat.	negat.
Multiple of characteristics of original set	1. point +1s, 2. point -1s	1. point +2s, 2. point -2s	1. point +3s, 2. point -3s
Users function	1.0122	2.0193	6.2412
Arithmetic mean	1.7396	–16.2757	–52.5175
St. deviation	1.0060	1.4182	2.4812
M. coeff. of skew.	1.8302	–12.6178	–17.8036
M. coeff. of curt.	1.6265	56.6179	86.6017
R-factor	1.0061	1.4210	2.4982
Other criteria			
AIC	–269.8000	–233.4000	–197.1000
MEP	0.0001	0.0005	0.0016
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.

Table 5. Various variants of calculation for Jack-Knife residuals

almost did not make itself felt in the evaluation of the regression process, hence it was not tested any more. This fact is connected with different interdependence of parameters throughout the course of regression dependence, e.g., see Ref. (MELOUN, 1984).

5. CONCLUSION

The tests performed unambiguously show that the normalized residuals (Tab. 3) are most suitable for evaluation of closeness of fit. The performed changes in analyzed data are most clearly reflected by the statistical characteristics of the sets of normalized residuals. Of course, these changes apply to such criteria as are the value of users function, the first and the second central moment and the R-factor. However, the parameters characterizing the form of probability distribution do not substantially change, which again speaks in favor of this type of residuals.

Similarly it is possible to evaluate the predicted residuals (Tab. 6), but in this case the variability is lower than that in the case of normalized residuals.

The Jack-Knife residuals (Tab. 5) can be placed behind the predicted residuals: the variability was still lower here. However, this type is very useful for guessing of significant points.

In the case of classical residuals (Tab. 2) the first group of criteria changes only little – hence it is problematic to evaluate changes of fitting – and the second group (the 3rd and the 4th central moments) are changed very markedly, which means that this type is absolutely unsuitable for evaluation of the quality of fitting.

Predicted residuals			
First point			
Multiple of characteristics of original set	1. point +1 s	1. point + 2s	1. point + 3s
Users function	0.9561	3.4940	12.5057
Arithmetic mean	0.0956	–8.0159	–17.0309
St. deviation	0.9778	1.8691	3.5919
M. coeff. of skew.	1.3179	–14.1426	–21.2923
M. coeff. of curt.	–1.0926	–181.0766	–318.7822
R-factor	0.9778	1.8692	3.5363
Other criteria			
AIC	–273.2000	–237.2000	–199.7000
MEP	0.0001	0.0004	0.0014
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.
Second point			
Multiple of characteristics of original set	1. point +1s, 2. point -1s	1. point +2s, 2. point -2s	1. point +3s, 2. point -3s
Users function	1.0829	3.9852	13.6282
Arithmetic mean	0.8011	–6.6097	–14.9265
St. deviation	1.0406	1.9962	3.6915
M. coeff. of skew.	2.3294	–11.3779	–19.3415
M. coeff. of curt.	–4.1581	–162.9381	–294.3271
R-factor	1.0406	1.9963	3.6916
Other criteria			
AIC	–269.8000	–233.4000	–197.1000
MEP	0.0001	0.0005	0.0016
Heteroscedasticity	yes	yes	yes
Normality	yes	yes	not
Autocorrelation	not	not	not
Sign test	negat.	negat.	negat.

Table 6. Various variants of calculation for predicted residuals

Variability was almost absent in the case of standardized residuals (Tab. 4): hence their application is utterly meaningless.

The values of criteria for comparison of the quality of ftting between the individual sets (i.e. AIC and MEP) clearly reproduce the worsening conditions of the calculation. However, this is the only piece of information obtained from these characteristics. Interestingly, the values of both AIC and MEP are better for the first two variants of calculation as compared with the basic set – this is due to the fact that the first point in the basic set has a lower experimental value of dependent variable as compared with the predicted value; therefore, during the changes the regression improves at the beginning.

The remaining characteristics (the tests of heteroscedasticity, distribution normality, autocorrelation, and the sign test) possess relatively low information ability, and their application can only be tentative. They can be successfully replaced by a map of distribution of residuals around the zero value, where the trends such as heteroscedasticity, normality, and autocorrelation or sign alternation can be evaluated much more objectively by mere inspection.

The problem has been dealt with in the framework of the research project MŠM 0021627505 „Control, optimization and diagnostics of complex systems“.

REFERENCES

Meloun, M., Javůrek, M. Multiparametric Curve Fitting VIII. The Reliability of Dissociation Constants Estimated by Analysis of Absorbance-pH Curves. *Talanta 32(10)*, (1985), pp. 973-986, ISSN 0039-9140

Meloun, M., Militký, J. *Statistical Data Analysis*.Cambridge, UK, Woodhead Publishing, Ltd. 2011, ISBN 978-0-85709-109-3, 900p.

Predictive Control Using Neural Network Applied on Semi-batch Reactor

L. Macků * D. Sámek**

** Tomas Bata University in Zlín,
Faculty of Applied Informatics,
Department of Electronics and Measurements.
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
(e-mail: macku@fai.utb.cz)
** Tomas Bata University in Zlín,
Department of Production Engineering.
Nam. T.G.Masaryka 275, 762 72 Zlín, Czech Republic.
Faculty of Humanities.
Mostní 5139, 760 01 Zlín, Czech Republic
(e-mail: samek@ft.utb.cz)*

Abstract: The article deals with the control of the semi-batch reactor that is used in chromium sludge processing. To simulate the real process a mathematical model including reaction kinetics was used. The parameters of the achieved model were obtained and verified by experiments. The control of the semi-batch reactor is difficult by common control methods because of the strongly exothermic chemical reaction. A model predictive control using artificial neural network is applied to the temperature control problem. The system control is generally complicated because of its nonlinearities.

1. INTRODUCTION

Although the leather industry is environmentally important as a user of by-products of the meat industry, it is perceived as a consumer of resources and producer of pollutants.

The most serious problem, which is now of a great importance, is chrome-tanned solid waste. One of the numerous possible solutions of the problem of chrome-tanned waste is its enzymatic dechromation. A chromium filter cake containing not only the alkali, but also a non-hydrolyzed protein is obtained. This fact can be used for the production of regenerated tanning chromium salts (Kolomazník at al. 2007).

Chromium filter cake (chromium sludge) processing can be done in a semi-batch reactor. Batch reactors provide flexible means of producing high value-added products in specialty chemical, biotechnical, and pharmaceutical industries. To realize the production objectives, these batch reactors have to be operated optimally in a precise fashion. However, due to the following characteristics: 1. intrinsic nonlinearity; 2. lack of steady-state operating conditions; 3. uncertainties in reaction dynamics, or modeling error; 4. unknown disturbances; 5. constraints on process variables; 6. and limited on-line measurement information, the optimization and control of batch reactors present some of the most interesting and challenging problems for both academia and industry in the process control arena (Hua at al. 2004).

Due to the complexity of the reaction mixture and the difficulties to perform on-line composition measurements, control of batch and fed-batch reactors is essentially a problem of temperature control. The temperature profile in batch reactors usually follows three-stages (Bouhenchir at al. 2006): (i) heating of the reaction mixture until the desired reaction temperature, (ii) maintenance of the system at this temperature and (iii) cooling stage in order to minimize the formation of by-products. Any controller used to control the reactor must be able to take into account these different stages.

2. PROCESS MODEL

In this paper, a fedbatch reactor model is used to study model predictive control method application. The model input data comes from a real process - the chromium waste recycling process (Macků 2003), (Janáčová 2006). Let us consider single input – single output (SISO) system of chemical exothermic semi-batch reactor. The mathematical model of this system can be written by equations (1)-(4).

$$\frac{\mathrm{d}\,m(t)}{\mathrm{d}\,t} = F_I \tag{1}$$

$$\frac{\mathrm{d}\,a(t)}{\mathrm{d}\,t} = \frac{F_I}{m(t)} - A \cdot e^{-\frac{E}{R \cdot T(t)}} \cdot a(t) \tag{2}$$

$$\frac{\mathrm{d}\,T(t)}{\mathrm{d}\,t} = \frac{F_I \cdot c_I \cdot T_I}{m(t) \cdot c} + \frac{A \cdot e^{-\frac{E}{R \cdot T(t)}} \cdot \Delta H_r \cdot a(t)}{c} - \\ - \frac{K \cdot S \cdot T(t)}{m(t) \cdot c} + \frac{K \cdot S \cdot T_C(t)}{m(t) \cdot c} \tag{3}$$

$$\frac{\mathrm{d}T_C(t)}{\mathrm{d}t}=\frac{F_C\cdot T_{CI}}{m_C}+\frac{K\cdot S\cdot T(t)}{m_C\cdot c_C}-\\-\frac{K\cdot S\cdot T_C(t)}{m_C\cdot c_C}-\frac{F_C\cdot T_C(t)}{m_C} \quad (4)$$

where *m* is the total weight of reaction components in the reactor, *a* is the mass concentration of the reaction component in the reactor, *c* = 4500 J.kg.K^{-1} is the specific heat capacity of the reactor content, *T* is the temperature of the reactor content. F_I, T_I = 293.15 K and c_I = 4400 J.kg.K^{-1} is the reaction component input mass flow rate, temperature and specific heat capacity. F_C = 1 kg.s^{-1}, T_{CI} = 288.15 K, T_C, c_C = 4118 J.kg.K^{-1} and m_C = 220 kg is the cooling water mass flow rate, input temperature, output temperature, specific heat capacity and weight of the cooling water in the cooling system of the reactor, respectively. Other constants: A = 219.588 s^{-1}, E = 29967.509 J.mol^{-1}, R = 8.314 J.mol^{-1}.K^{-1}, ΔHr = 1392350 J.kg^{-1}, K = 200 kg.s^{-3}.K^{-1}, S = 7.36 m^2.

3. MODEL PREDICTIVE CONTROL

The task was to control the in-reactor temperature *T* by reaction component dosing F_I. The desired value of temperature *T* was 370K and the maximum value shouldn't exceed 373K. The actuating variable F_I was from the interval <0,3> kg.s^{-1}.

The basic idea of model predictive control (MPC) is to use a model to predict the future output trajectory of a process and compute a series of controller actions to minimize the difference between the predicted trajectory and a user-specified one, subject to constraints (Garcia at al. 1989), (Camacho 2004), Fig. 1.

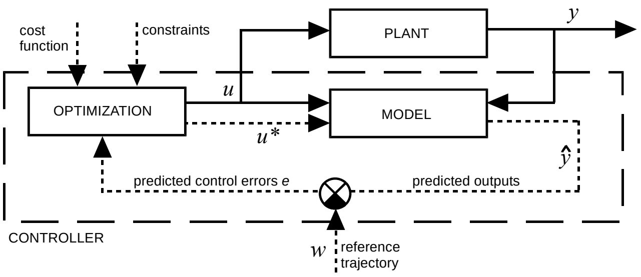


Fig. 1. The basic scheme of model predictive control.

Generally we can say that MPC uses a predictor network (ANN) as the plant model in order to get its output predictions. The controller then calculates the control input that will optimize the performance criterion over a specified future time horizon (Zhang 2008). Typical form of the performance criterion *J* is as follows:

$$J=\lambda\sum_{j=N_1}^{N_2}\left[y_r(k+j)-\hat{y}(k+j)\right]^2+\\+\rho\sum_{j=1}^{N_u}\left[u_t(k+j-1)-u_t(k+j-2)\right]^2 \quad (5)$$

where N_1, N_2 and N_u define horizons over which the tracking error and the control increments are evaluated. The u_t variable is the tentative control signal, y_r is the desired response and $\hat{y}$ is the predictor response. The λ and ρ parameters determine the contribution that the particular sum has on the performance index.

The selection of predictor is a key question in the model predictive control (Mazinan 2008). Because the controlled system is nonlinear, an artificial neural network (ANN) was selected (Volosencu 2009). After many simulations and tests the multilayered feed-forward neural network with three layers was chosen as the best solution from the wide group of artificial neural networks. From the figure 2 can be seen that as a transfer function the hyperbolic tangent was used in the both hidden layers, while in the output layer the linear function was applied. The ANN predictor used five last values of the system output and the controlled signal as an input. The ANN based predictor was trained offline using offline prepared identification data.

The minimization of the performance function is in the linear MPC typically provided by quadratic programming (Tondel at al. 2003), (Kouvaritakis at al. 2002). Nevertheless, because of the nonlinearity of the predictor and the usage of constraints it was necessary to apply a numerical optimization method. Therefore, the Levenberg-Marquart method, which is implemented in the Matlab Optimization Toolbox (Venkataraman 2009), was used in this paper.

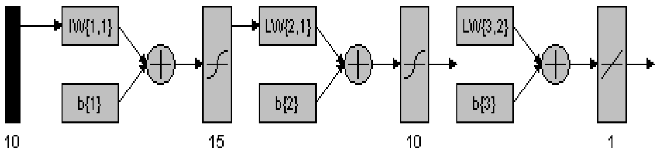


Fig. 2. The based on artificial neural network.

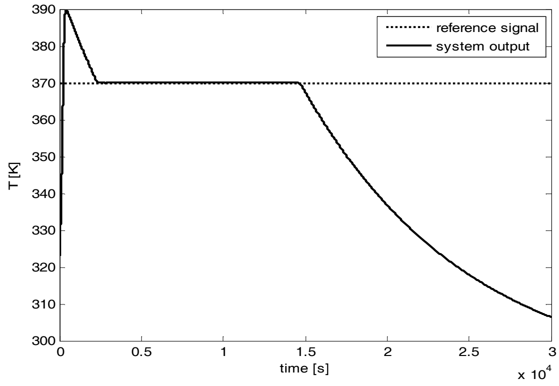


Fig. 3. The in-reactor temperature development – MPC1.

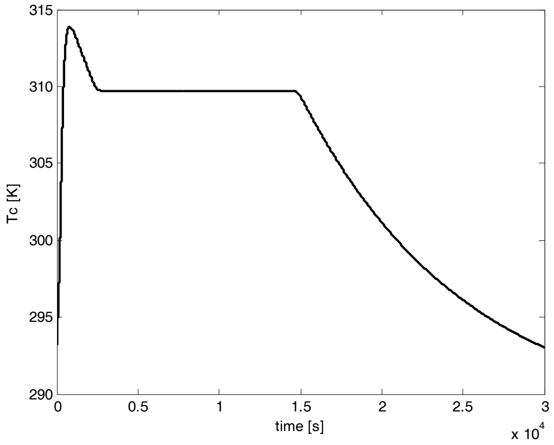


Fig. 4. The temperature in the cooling system – MPC1.

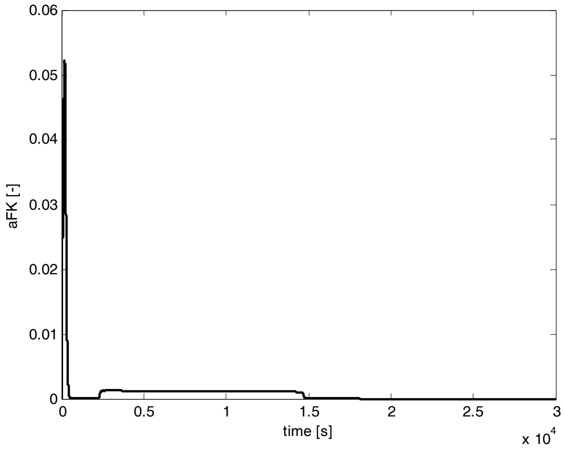


Fig. 5. The in-reactor chromium sludge concentration development – MPC1.

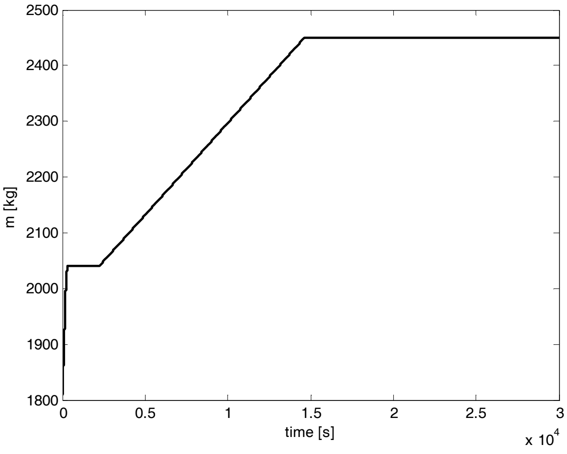


Fig. 6 The mass of reaction mixture – MPC1.

In the figures 3, 4, 5 and 6 there are presented results of selected simulation of control using MPC controller with the criterion function (6). The presented simulation used the following settings of the controller: λ =1000, ρ =100000, N_1=1, N_2=8, N_u=8. However, this “standard” approach does not provide satisfactory performance in case of this semi-batch plant. The time of the batch must be as short as possible because of the economical reasons. But it is impossible to obtain fast batch without overshoot of temperature by any combination of controller parameters. The increase of ρ parameter can reduce the temperature overshoot but in the cost of long batch time.

Therefore, the third part to the criterion function (5) was added in order to reduce the speed of dosing (control signal $\underline{u}$). The γ parameter determines the influence of nominal values of future control signal on the cost function (6). Results obtained using this cost function is in the following text denoted as MPC2. The settings of the controller were: λ=1000, ρ=10000, γ =10000, N_1=1, N_2=8, N_u=8. As can be seen from figures 7, 8, 9 and 10, the controller has permanent control error. In order to show this negative behaviour more clearly, it is assumed in the MPC2 that there is unlimited amount of the chromium sludge (batch input).

$$\begin{aligned} J = \lambda \sum_{j=N_1}^{N_2} \left[y_r(k+j) - \hat{y}(k+j)\right]^2 + \\ + \rho \sum_{j=1}^{N_u} \left[u_t(k+j-1) - u_t(k+j-2)\right]^2 \\ + \gamma \sum_{j=1}^{Nu} u_t(k+j) \end{aligned} \qquad (6)$$

It can be deduced from MPC2 results that the size of the control signal had to be penalized in the beginning of the batch only. Thus, the criterion function (6) was modified into the form defined by equations (7) and (8). Then, the γ parameter was during the control gradually decreased up to zero in order to avoid the permanent control error. In other words, the third sum in the beginning of the control has the maximum value, and after initial phase it equals to zero. The γ_c parameter determines the speed of the decrement in γ.

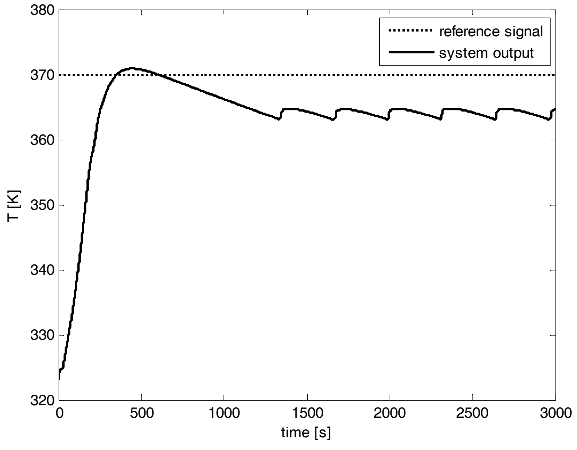


Fig.7 The in-reactor temperature development – MPC2.

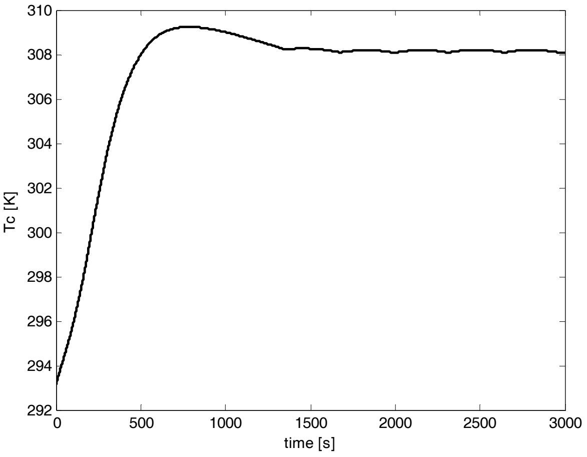


Fig.8 The temperature in the cooling system – MPC2.

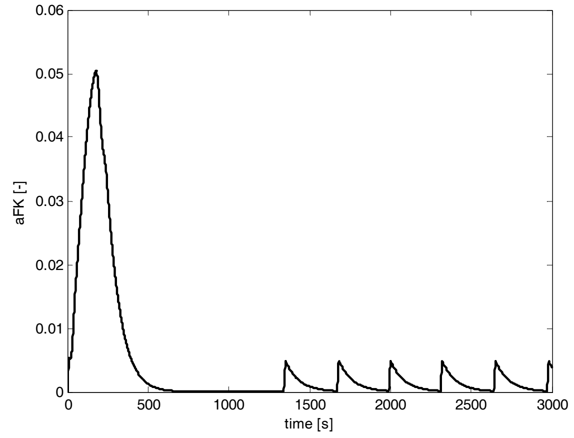


Fig.9 The in-reactor chromium sludge concentration development – MPC2.

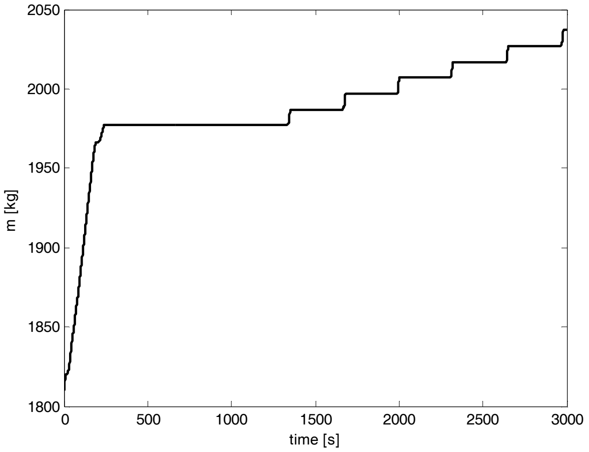


Fig.10 The mass of reaction mixture – MPC2.

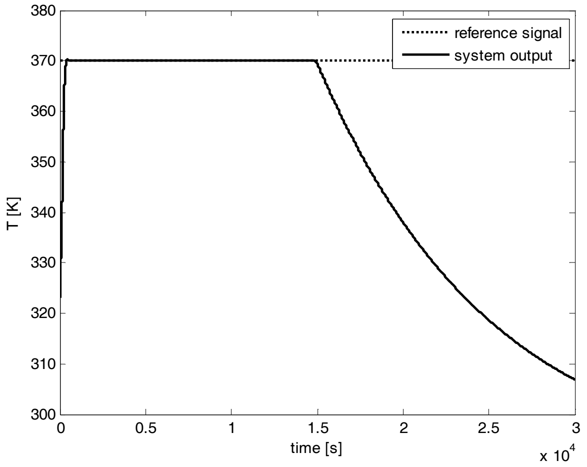


Fig.11 The in-reactor temperature development – MPC3.

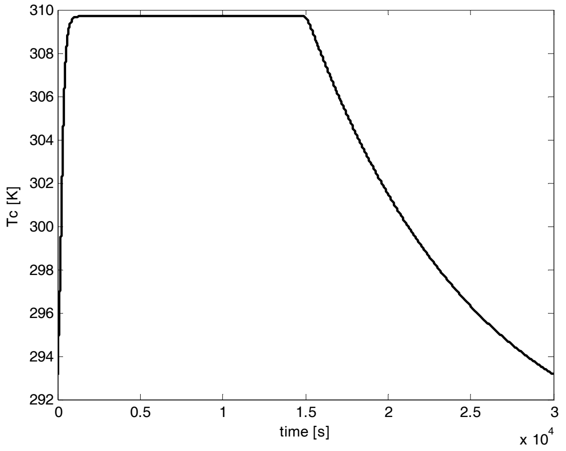


Fig.12 The temperature in the cooling system – MPC3.

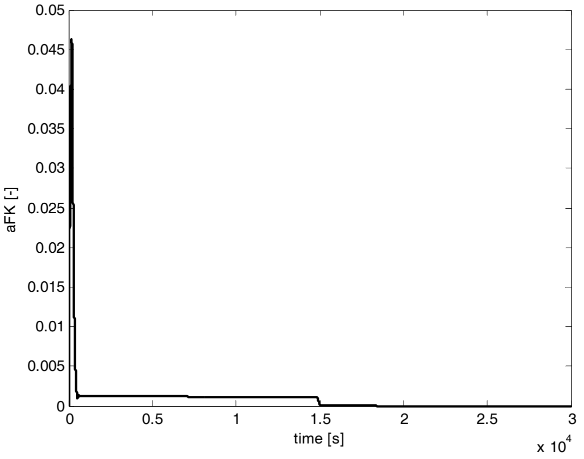


Fig.13 The in-reactor chromium sludge concentration development – MPC3.

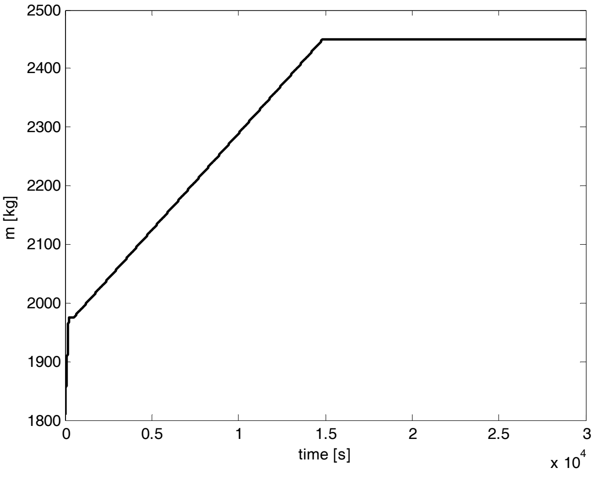


Fig.14 The mass of reaction mixture – MPC3.

$$J = \lambda \sum_{j=N_1}^{N_2} \left[y_r(k+j) - \hat{y}(k+j) \right]^2 +$$

$$+ \rho \sum_{j=1}^{N_u} \left[u_t(k+j-1) - u_t(k+j-2) \right]^2 \quad (7)$$

$$\gamma(k) \sum_{j=1}^{N_u} u_t(k+j)$$

$$\gamma(k) = \gamma(k-1) - \gamma_c \quad (8)$$

The controller with cost function defined by equations (7) and (8) was tested in simulation MPC3 with the following settings: λ =1000, ρ =100000, γ=10000, γ_c =200, N_1=1, N_2=8, N_u=8. As can be seen from the figures 11 – 14, the MPC3 results were: the upper-most in-reactor temperature T reached 370.78 K, the maximum chromium sludge concentration a was 0.0461 and the total batch time made 25499 seconds.

The maximum and minimum actuating variable values were 0.9375 kg.s^{-1} or 0 kg.s^{-1} respectively. The steady state actuating variable value made approximately 0.031 kg.s.

4. CONCLUSION

The best control performance was obtained by MPC3, but simulation of this method is quite hardware demanding today. The simulation using CPU 2500 MHz computer took almost 2 hours. As can be seen, the MPC can solve even such difficult task as nonlinear system, chemical reactor is.

ACKNOWLEDGMENTS

The work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under grant MSM 7088352102. This support is gratefully acknowledged.

REFERENCES

Able, B.C. (1956). Nucleic acid content of micro-scope. *Nature* **135**, 7-9.

Able, B.C., R.A. Tagg and M. Rush (1954). Enzyme-catalyzed cellular transanimations. In: *Advances in Enzymology* (A.F. Round, Ed.). 3rd ed.. **Vol. 2.** pp. 125-247. Academic Press. New York.

Keohane, R. (1958). *Power and Interdependence: World Politics in Transitions*. Little, Brown & Co.. Boston.

Kolomazník, K., Adámek, M., Uhlířová, M. (2007). Potential Danger of Chromium Tanned Wastes, *In Proceedings of the 5th IASME/WSEAS International Conference on Heat Transfer, Thermal Engineering and Environment*, IASME/WSEAS, p. 137-141.

Hua, X., Rohani, S., Jutan, A. (2004). Cascade closed-loop optimization and control of batch reactors, *Chemical Engineering Science*, Vol. 59, p. 5695 – 5708.

Bouhenchir, H., Cabassud, M., Le Lann, M.V. (2006). Predictive functional control for the temperature control of a chemical batch reactor. *Computers and Chemical Engineering*. Issue 30, p. 1141-1154.

Macků, L. (2003). *Control design for the preparation of regenerate for tanning*, Ph.D. Thesis, UTB in Zlin.

Janacova, D., Kolomaznik, K., Mokrejs, P., Vasek, V. (2006). Optimization of enzymatic hydrolysis of leather waste, In Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, WSEAS, p. 345-348.

Garcia, C.E., Prett, D.M., Morari, M., Model predictive control: theory and practice – a survey, *Automatica*, Vol. 25, No. 3, pp. 335-348.

Camacho, E. F., Bordons, C. (2004). *Model Predictive Control in the Process Industry*. Springer - Verlag.

Zhang J., Wang, W. (2008). Synthesis of explicit model predictive control system with feasible region shrinking, *In 8th WSEAS Conf. on Robotics, Control and Manufacturing Technology*, WSEAS.

Mazinan, A. H., Sadati, N. (2008). Multiple Modeling and Fuzzy Predictive Control of a Tubular Heat Exchanger System, *WSEAS TRANSACTIONS on SYSTEMS and CONTROL*, Vol. 3, No. 4, pp. 249 – 258.

Volosencu, C. (2009). Identification of Non-Linear Systems, Based on Neural Networks, with Applications at Fuzzy Systems, *In Proceedings of the 10th WSEAS International Conference on Automation & Information*, WSEAS.

Tondel, P., Johansen, T. A., Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions, *Automatica*, Vol. 39, No. 3, pp. 489 – 497.

Kouvaritakis, B., Cannon, M., Rossiter, J. A. (2002). Who needs QP for linear MPC anyway?, *Automatica*, Vol. 38, No. 5, pp. 879 – 884.

Venkataraman, P. (2009). Applied Optimization with MATLAB Programming. John Wiley & Sons, Inc.

Pole placement controller with compensator adapted to semi-batch reactor process

D. Novosad* L. Macků**

** Tomas Bata University in Zlín,*
Faculty of Humanities.
Mostní 5139, 760 01 Zlín, Czech Republic
(e-mail: novosad@fhs.utb.cz)
*** Tomas Bata University in Zlín,*
Faculty of Applied Informatics, Department of Electronics and Measurements.
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
(e-mail: macku@fai.utb.cz)

Abstract: This paper deals with the modelling and control of semi-batch reactor used for chromium sludge regeneration process. A comparison of three process control approaches is presented. Usual PID controller without online identification (OI) and adaptive PID controller were adapted to semi-batch rector process in our previous studies. In this study the two-degrees-of-freedom (2DOF) controller is developed for the same reactor control.

1. INTRODUCTION

Batch and semi-batch reactors are widely used in chemical, biotechnical, and pharmaceutical industries. To obtain the desired product quality during the production period an accurate temperature control is required. The temperature profile in batch and semi-batch reactors usually follows three stages (Bouhenchir et al. 2010): (i) heating of the reaction mixture up to the desired reaction temperature, (ii) maintenance of the system at this temperature and (iii) cooling stage in order to minimize the formation of by-products. Any controller used to control the reactor must be able to take into account these different stages.

In the literature some papers have been published which discuss the control of a batch or semi-batch reactor. A global linearization control strategy was applied, with online state and parameter estimation for a polymerization reactor (Beyer et al 2008). However, the authors concluded that the implementation of the proposed method is still difficult due to the missing support of required mathematical functions. The other approach was used in the next study (Cho et al. 2008), where the authors applied a dual-mode control improved by iterative learning technique. Simulations showed that the proposed method can enhance the conventional DM control with modest efforts. For rapid and suitable reference-trajectory tracking a self-adaptive predictive functional control algorithm by Škrjanc was recommended (Škrjanc 2007). This approach was successful in a reactor with switching between cold and hot water in the inlet. Neural network was applied to similar system (Wu et al. 2010) to accommodate the online identification of a nonlinear system. The authors found this strategy effective in identification and control of a class of time-varying-delayed nonlinear dynamic systems. Neural networks are often presented as a good method to reach useful results in batch processes. Some authors recommended using a tuning parameter allowing the designer to select the damping of the closed-loop responses (Gorez 2003). A damping factor was tested in milking machine vacuum control (Reinemann 2005). Author argued that the damping factor is influenced by the system design as well as the amount of damping in the regulation device itself. Reinemann found the best value of damping factor = 1. Damping factor < 1 was causing oscillations. On the other hand, damping factor > 1 led to under-shoot. In another paper (Haugen 2005) was damping factor adjusted to the value 0,6 instead 1 for obtaining faster dynamics using Skogestad's method.

This paper presents results of experiments obtained by simulations and control of the semi-batch process using PID controller without online identification, adaptive PID controller and pole placement 2 degree-of-freedom (2DOF) controller with compensator for second order processes. The paper is organised as follows. In section 2, the semi-batch reactor and 2DOF controller are described; section 3 presents simulation results and section 4 concludes the current work and suggests new areas for investigation.

2. METHODS SECTION

2.1 The semi-batch reactor model

To simulate tanning salts from chromium sludge regeneration process a mathematical model is used. The chemical reactor scheme is shown in Fig 1. We can see there a vessel with a double wall used for a cooling liquid circulation. It has two inputs and one output. Inputs are for the cooling liquid and for the chemical compound dosing, output only for the cooling liquid.

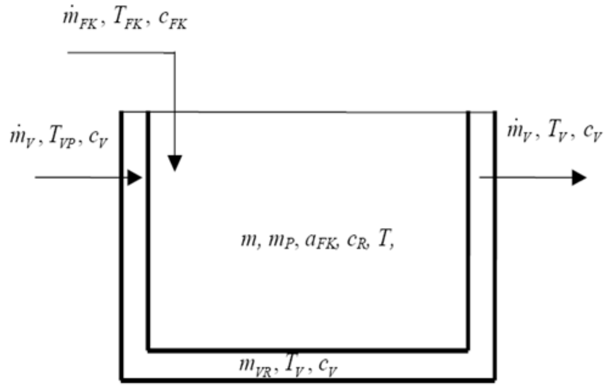


Fig. 1. Chemical reactor scheme

The mathematical model of the fed-batch reactor is defined by differential equations 1-4.

$$\dot m_{FK} = \frac{d}{dt} m(t) \tag{1}$$

$$\dot m_{FK} = k\, m(t)\, a_{FK}(t) + \frac{d}{dt}\left[m(t) a_{FK}(t)\right] \tag{2}$$

$$\begin{aligned} &\dot m_{FK} c_{FK} T_{FK} + \Delta H_r k\, m(t)\, a_{FK}(t) = K\, S \\ &\cdot\left[T(t) - T_V(t)\right] + \frac{d}{dt}\left[m(t)\, c_R\, T(t)\right] \end{aligned} \tag{3}$$

$$\dot m_V c_V T_{VP} + K\, S\left[T(t) - T_V(t)\right] = \dot m_V c_V T_V(t) + m_{VR} c_V T'_V(t) \tag{4}$$

The reactor model comprises the total mass balance (1), chromium sludge mass balance, where $\dot m_{FK}$ is used as control signal in this process (2), the enthalpy balance (3) and coolant heat balance (4). Further variables and the parameters of the reactor model are listed in Table 1. In Eq. (2), k [s^{-1}] is the reaction rate constant expressed by the Arrhenius equation:

$$k = Ae^{-\frac{E}{RT(t)}} \tag{5}$$

Detailed description of this model is given in Macků (2003).

2.2 Pole placement 2 degree-of-freedom controller with compensator for second order processes

In this work, 2DOF controller was applied to calculate the optimal temperature trajectory to reach desired properties in minimum time.

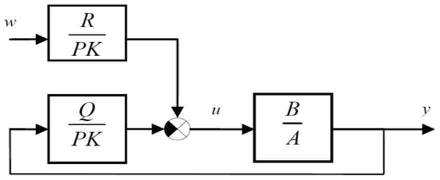


Fig. 2. 2DOF control loop

Table 1 - Variables and parameters of the reactor model

Fig. 1	m_p [kg]	Initial amount of reaction solution
(1)	$\dot m_{FK}$ [kg.s^{-1}]	Mass flow of the entering chromium sludge
	$m(t)$ [kg.s^{-1}]	Accumulation of the in-reactor content
(2)	$a_{FK}(t)$ [-]	Mass concentration of the chromium sludge
	$m(t)$ [kg]	Weight of the reaction components in the system
	k [s^{-1}]	The reaction rate constant
(3)	c_{FK} [J.kg^{-1}.K^{-1}]	Chromium sludge specific heat capacity (c_{FK} = 4400 J.kg^{-1}.K^{-1})
	c_R [J.kg^{-1}.K^{-1}]	Reactor content specific heat capacity (c_{FK} = 4500 J.kg^{-1}.K^{-1})
	T_{FK} [K]	Chromium sludge temperature
	ΔH_r [J.kg^{-1}]	Reaction heat (ΔH_r = 1392350 J.kg^{-1})
	K [kg^{-3}.K^{-1}]	Conduction coefficient (K = 200 kg^{-3}.K^{-1})
	S [m^2]	Heat transfer surface (S = 7,36 m^2)
	$T(t)$ [K]	Temperature of reaction components in the reactor
	$T_v(t)$ [K]	Temperature of coolant in the reactor double wall
(4)	$\dot m_v$ [kg.s^{-1}]	Coolant mass flow
	c_v [J.kg^{-1}.K^{-1}]	Coolant specific heat capacity (c_v = 4118 J.kg^{-1}.K^{-1})
	T_{vp} [K]	Input coolant temperature
	m_{vR} [kg]	Coolant mass weight in the reactor double wall (m_{vR} = 220 kg)
(5)	A [s^{-1}]	Pre-exponential factor (A = 219,588 s^{-1})
	E [J.mol^{-1}]	Activation energy (E = 29967,5087 J.mol^{-1})
	R [J.mol^{-1}.K^{-1}]	Gas constant (R = 8,314 J.mol^{-1}.K^{-1})

Feedback controller:

$$G_R = \frac{\mathrm{Q}(z^{-1})}{\mathrm{P}(z^{-1})K(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{(1 + p_1 z^{-1})(1 - z^{-1})} \tag{6}$$

Feedforward controller for a step reference signal:

$$G_F = \frac{\mathrm{R}(z^{-1})}{\mathrm{P}(z^{-1})K(z^{-1})} = \frac{r_0}{(1 + p_1 z^{-1})(1 - z^{-1})} \tag{7}$$

Characteristic polynomial of closed loop:

$$A(z^{-1})P(z^{-1})K(z^{-1}) + B(z^{-1})Q(z^{-1}) = D(z^{-1}) \tag{8}$$

Where polynomials are as follows:

$$A(z^{-1}) = 1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} \tag{9}$$

$$P(z^{-1}) = 1 + \hat{p}_1 z^{-1} \tag{10}$$

$$K(z^{-1}) = 1 - z^{-1} \tag{11}$$

$$B(z^{-1}) = \hat{b}_1 z^{-1} + \hat{b}_2 z^{-2} \tag{12}$$

$$Q(z^{-1}) = q_0 + q_1 z^{-1} + q_2 z^{-2} \tag{13}$$

$$D(z^{-1}) = 1 + d_1 z^{-1} + \ldots + d_4 z^{-4} \tag{14}$$

$$\mathrm{d}_1 = \text{-}2\exp(\text{-}\ \xi\omega\mathrm{T}_0)\cos\left(\omega\mathrm{T}_0\sqrt{1\text{-}\ \xi^2}\right) \tag{15}$$

$$\mathrm{d}_1 = \text{-}2\exp(\text{-}\ \xi\omega\mathrm{T}_0)\cosh\left(\omega\mathrm{T}_0\sqrt{\xi^2 - 1}\right) \tag{16}$$

$$\mathrm{d}_2 = \exp(\text{-}\ 2\xi\omega\mathrm{T}_0) \tag{17}$$

$$\mathrm{d}_3 = \mathrm{d}_4 = 0 \tag{18}$$

where ξ is damping factor and ω is natural frequency. Both parameters specifying dynamic behaviour of closed loop. The dynamic behaviour of the closed-loop is similar to second order continuous system with characteristic polynomial $s^2 + 2 \cdot \xi \cdot \omega \cdot s + \omega^2$.

Matrix equation:

$$\begin{bmatrix} \hat{b}_1 & 0 & 0 & 1 \\ \hat{b}_2 & \hat{b}_1 & 0 & \hat{a}_1 - 1 \\ 0 & \hat{b}_2 & \hat{b}_1 & \hat{a}_2 - \hat{a}_1 \\ 0 & 0 & \hat{b}_2 & -\hat{a}_2 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} d_1 + 1 - \hat{a}_1 \\ d_2 + \hat{a}_1 - \hat{a}_2 \\ d_3 + \hat{a}_2 \\ d_4 \end{bmatrix} \tag{19}$$

Control law:

$$P(z^{-1})K(z^{-1})u_k = R(z^{-1})w_k - Q(z^{-1})y_k \tag{20}$$

$$u_k = r_0 w_k - q_0 y_k - q_1 y_{k-1} - q_2 y_{k-2} + (1 - p_1)u_{k-1} + p_1 u_{k-2} \tag{21}$$

$$r_0 = \frac{1 + d_1 + d_2 + d_3 + d_4}{\hat{b}_1 + \hat{b}_2} \tag{22}$$

2.2 Online identification method

Proportional-integral-derivate (PID) controllers have been the most commonly used feedback controllers in the past years. The popularity and widespread use of PID controllers attributed to their simplicity and robustness but it cannot effectively control some complicated or fast running systems since the response of a plant depends on only the gain P, I and D. Most of the PID tuning rules developed in the past years use the conventional methods. For example, the Ziegler-Nichols approach often leads to a rather oscillatory response to set-point changes because of system non-linearities and various uncertainties such as modelling error and external disturbances. These methods provide simple tuning formulae to determine the PID controller parameters. However, since only a small amount of information on the dynamic behaviour of the process is used, in many situations they do not provide good enough tuning or produce a satisfactory closed-loop response.

This was the reason to improve classical PID controller with parameters tuned according to Ziegler-Nichols from previous study (Novosad 2007). Controller was equipped by recursive least squares identification based on ARX model which can be used for the discrete on-line identification of processes described by the following transfer function:

$$G(z) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}} z^{-d} \tag{23}$$

The estimated output of the process in the step is computed on the basis of the previous process inputs and outputs according to the equation:

$$\begin{aligned} \hat{y}_k &= -\hat{a}_1 y_{k-1} - \ldots - \hat{a}_n y_{k-n} \\ &\quad + \hat{b}_1 u_{k-d-1} + \ldots + \hat{b}_m u_{k-d-m} \end{aligned} \tag{24}$$

where $\hat{a}_1,..\hat{a}_n, \hat{b}_1,...,\hat{b}_m$ are the current estimations of the process parameters. This equation can be also written in vector form, which is more suitable for further work:

$$\begin{aligned} \hat{y}_k &= \Theta_{k-1}^T \cdot \Phi_k \\ \Phi_{k-1} &= \left[\hat{a}_1,..\hat{a}_n, \hat{b}_1,..\hat{b}_m\right]^T \\ \Phi_k &= \left[-y_{k-1},..-y_{k-n}, u_{k-d-1},..u_{k-d-m}\right]^T \end{aligned} \tag{25}$$

The vector Θ_{k-1} contains the process parameter estimations computed in the previous step and the vector Φ_k includes output and input values for computation of current output y_k. This on-line identification is also used for 2DOF controller.

3. RESULTS SECTION

Identification of suitable models which accurately describe a batch reactor process is essential to successful optimization and control. In this study, on a semi-batch reactor by means of a simulation, 2DOF controller was tested and the effect of changes of the various parameters for a quality of the regulation process was monitored.

First, ξ (damping factor) coefficient was adjusted. In this aspect it must be said that some authors (Haugen 2005) introduced the recommended values of parsliameter ξ around 1; however, these values were not suitable in our case. It is necessary to use a far higher value. In Fig. 3, several temperature profiles with different ξ coefficients are plotted. As can be seen, the performance of the 2DOF is the best for parameter $\xi = 45$. In cases of lower ξ, the setpoint is overshot (Fig 3. and Fig. 4.).

Cooling stage occurs when the reactor is full and the feeding is stopped.

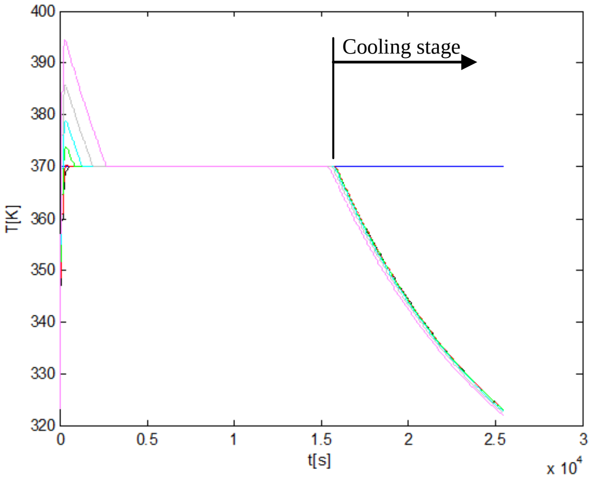


Fig. 3. Comparison of the temperature profiles for different damping factor

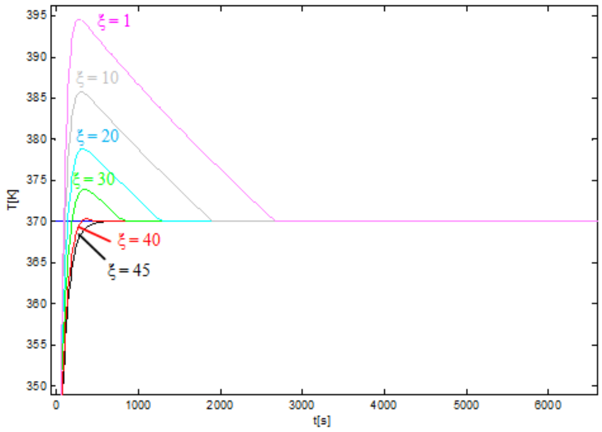


Fig. 4. Comparison of the temperature profiles for different damping factor – detailed view

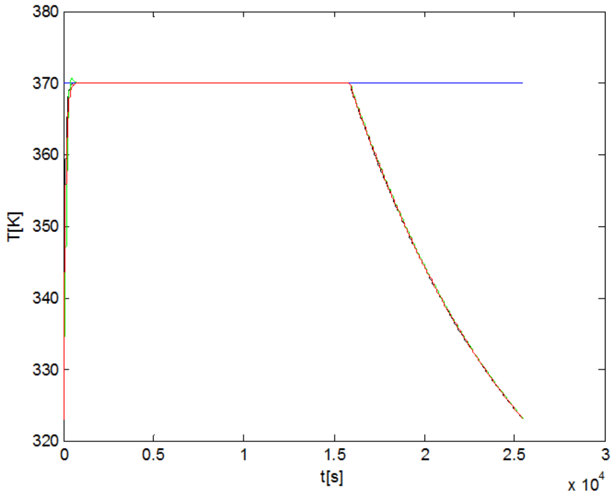


Fig. 5. Comparison of the temperature profiles

Figure 5 and 6 shows comparison of different methods of controlling semi-batch process. It can be seen that behaviour of PID controller without online identification and 2DOF controller are almost similar, both without oscillating and overshoots. In case of 2DOF, setpoint is reached faster for about 250 s. On the other hand, the performances in case of PID controller is slightly worse with overshoot at the beginning of the process and followed by undershoot.

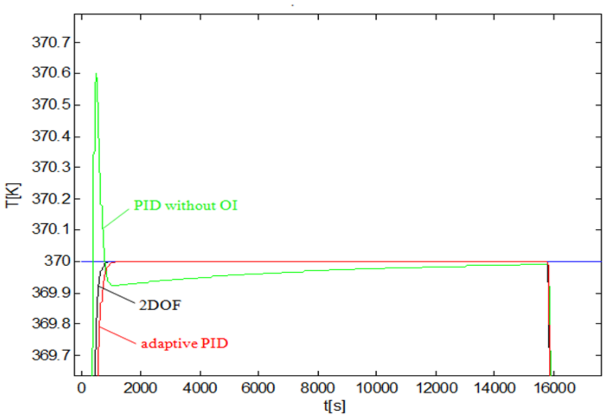


Fig. 6. Comparison of the temperature profiles – detailed view

Some differences in feeding can be also showed (Fig. 7). Maximum feeding (3 kg.s^{-1}) is reached in cases of controllers with online identification (adaptive PID and 2DOF). 2DOF controller provides the highest rates of the feeding at the beginning of the process and then feeding fall until the zero - feeding is stopped for about 150 s.

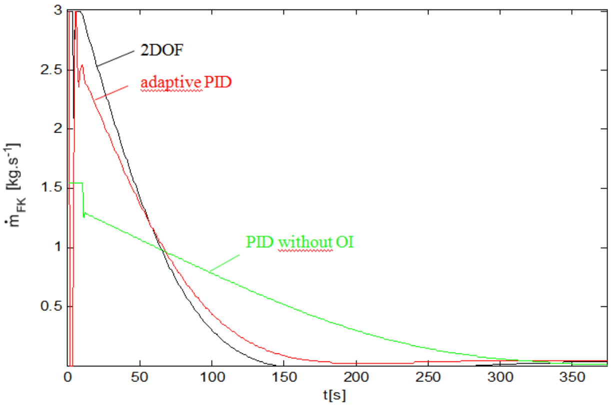


Fig. 7. Comparison of the feeding profiles at the beginning of the process

4. DISCUSSION SECTION

In this study, the 2DOF controller for the temperature control in a semi-batch reactor was demonstrated by simulation means. The process control sensitivity is influenced by damping factor parameter. In general it can be said that increasing of damping factor leads to reducing the overshoot and the response becomes slower. The implemented control strategy was also compared with two control strategies using

PID controllers applied on the same process in the previous works (Novosad 2007, Novosad & Macků 2010). Based on presented results it can be concluded that proposed 2DOF controller can effectively overcomes problems with oscillating around the desired value in comparison with PID controller without online identification. The quality of the regulation process in cases of controllers with implemented online identification (adaptive PID and 2DOF) shows satisfactory results.

There are still some other methods, which could possibly improve this process. In the future work, some other approaches will be applied to the batch process to find out other possible ways.

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Education, Youth and Sports of the Czech Republic under grant MSM 7088352102 and by the Tomas Bata University under grant IGA/40/FAI/10/D. These supports are gratefully acknowledged.

Authors are also grateful to Dr. Chalupa, whose self-tuning simulink library for real-time control was used for simulations mentioned in this paper.

REFERENCES

BEYER, M.A.; GROTE, W.; REINIG, G. 2008. Adaptive exact linearization control of batch polymerization reactors using a Sigma-Point Kalman Filter. Journal of Process Control. 2008, Issue 18, p. 663-675. ISSN 0959-1524.

BOUHENCHIR, H.; CABASSUD, M.; LE LANN, M.V. 2006. Predictive functional control for the temperature control of a chemical batch reactor. Computers and Chemical Engineering. 2006, Issue 30, p. 1141-1154.

CHO, W.; EDGAR, T.F.; LEE, J. 2008. Iterative learning dual-mode control of exothermic batch reactors. Control Engineering Practice. 2008, 16, p. 1244-1249. ISSN 0967-0661.

GOREZ, R. New design relations for 2-DOF PID-like control systems. Automatica. 2003, 39, s. 901-908.

HAUGEN, F. PID Control. [s.l.] : Tapir academic press, 2005. 286 s. ISBN 82-519-1945-2.

MACKŮ, L.: Control design for the preparation of regenerate for tanning. PhD. Thesis, UTB in Zlin, 2003.

NOVOSAD, D. 2007. Modely a principy řízení vsádkových reaktorů, Graduation Theses, UTB Zlín, 2007.

NOVOSAD, D. & MACKŮ, L. (2010). Ziegler-Nichols Controller with Online Identification Versus PID Controller Comparison (2010). 1017-1019, Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, ISBN 978-3-901509-73-5, ISSN 1726-9679, pp 0509, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria 2010.

REINEMANN D.J., The History of Vacuum Regulation Technology. Proceedings of the 44th Annual meeting of the National Mastitis Council. 2005.

ŠKRJANC, I. 2007. Self-adaptive supervisory predictive functional control of a hybrid semi-batch reactor with constraints. University of Ljubljana, Slovenia 2007.

WU, X.; ZHANG, J.; ZHU, Q. 2010. A generalized procedure in designing recurrent neural network identification and control of time-varying-delayed nonlinear dynamic systems. Neurocomputing. 2010, Issue 73, p. 1376-1383. ISSN 0925-2312.

Real-Time Model Predictive Control of a Fan Heater via PLC

Ivan Rauová [*], **Richard Valo** [*], **Michal Kvasnica** [*], **Miroslav Fikar** [*]

[*] *Institute of Automation, Information Engineering and Mathematics, Slovak University of Technology in Bratislava, 812 37 Bratislava, Slovakia, http://www.kirp.chtf.stuba.sk*
{ivana.rauova,richard.valo,michal.kvasnica,miroslav.fikar}@stuba.sk

Abstract: This paper deals with real-time implementation of Model Predictive Control (MPC) of a fan heater system using Programmable Logic Controller (PLC) platform. The MPC problem is solved using parametric programming techniques, which encode the optimal control moves as a lookup table. The challenge then becomes how to implement such a table on a memory-restricted device. The proposed design procedure is illustrated on real-time control of a laboratory heat exchange plant.

Keywords: model predictive control, programmable logic controllers

1. INTRODUCTION

Model predictive control (MPC) is an attractive approach widely used in industry to control a broad range of the systems due to its ability to provide optimal performance while taking process constraints into account (Maciejowski, 2002). In MPC the control objectives are translated into an optimization problem, which is formulated over a finite prediction horizon. The result of the optimization is a sequence of optimal control moves which drives system states towards a given reference point while respecting system constraints (such as upper and lower limits on the inputs and states) and optimizing a selected performance criterion. Traditionally, MPC is implemented in a so-called Receding Horizon fashion where the optimal control move is achieved by solving optimization problem in each time instance for a newly measured state. This approach induces lot of computation load at each sampling time, which might be prohibitive if not enough computation power is available or if sampling time is too short.

If less powerful control platforms are employed, additional care has to be taken to respect real-time constraints. One approach to decrease computational load involved in obtaining of optimal control action u^* for a particular value of x is to “pre-compute” the optimal solution to a given optimization problem for all possible initial conditions of x using parametric programming techniques (Bemporad et al., 2002). The optimal control can be then found as an explicit function $u^*(x)$ mapping the states to the control inputs. The function is computed off-line and takes a form of lookup table. Implementation of such a table can be done very efficiently on-line, as the evaluation of the feedback law involves only matrix multiplications, additions and logic comparisons. As a consequence, real-time implementation of such an *explicit* MPC can be done much faster compared to traditional on-line MPC fashion.

In this work we aim at implementing explicit MPC on a Programmable Logic Controller (PLC) restricted to 1024 bytes of memory. Three factors determine whether the design procedure will be successful:

(i) whether it is possible to construct the explicit MPC controller off-line in an automated fashion;
(ii) whether the controller is reasonably small as not to exceed the memory capabilities of the PLC;
(iii) whether the controller can be implemented using programming instructions which the control device understands.

In the paper we illustrate how to synthesize the parametric solution to MPC optimization problem using the Multi-Parametric (MPT) Toolbox (Kvasnica et al., 2004) and how to implement it on a PLC. First, we introduce the controlled plant and derive its mathematical model. Then we show which commands have to be used to set up the MPC optimization problem and how to synthesize the explicit MPC controller using MPT. Having a lookup table we introduce an algorithm capable to transform it into a binary search tree (BST), which can be downloaded directly to the PLC. At the end we show results of the laboratory plant control using explicit MPC controller provided on the PLC.

2. PHYSICAL SETUP

2.1 Controlled Plant

The laboratory Air-stream and Temperature Control Plant LTR 700 (Svetíková et al., 2003) is produced by the German company Amira. It consists of a fan, a heating coil, a differential pressure sensor, a temperature sensor, and an actuator box.

This plant is designed for heating the entering medium. Commonly, air is the medium which is intake to the plant thanks to the fan. The entering air is further heated by heating coil. In order to obtain hot air of desired amount and temperature, we can manipulate the fan speed or the amount of heat generated by heating coil. In this work, the heating coils output is set to constant value (50%). We aim at controlling only the air-flow rate (manipulated variable) by fan speed (control variable).

The plant is schematically illustrated in Fig. 1. Actual temperature and air-flow can be measured by sensors (TI, FI).

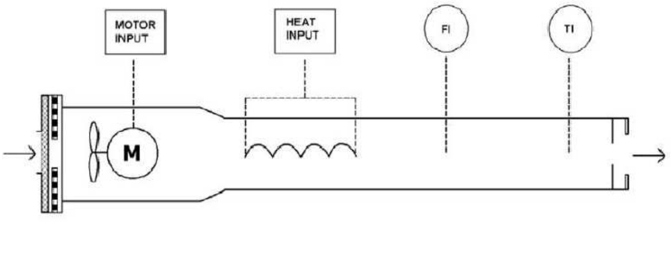


Fig. 1. Sensors in the air heater.

It is possible to implement several control configurations ranging from simple feedback loops, through cascade loops, up to multivariable control with two inputs and two outputs.

Airflow The plant converts air-flow value to a current signal in a range of 10.4-20 mA. This output signal is connected to the 3rd input of analog I/O module which is shown in Fig. 2. For our control program, the connected signal is converted into an integer number with physical address $AIW4$ (A – analog, I – input of the PLC, W - memory size of 16 bits, 4 – 3rd input to PLC). This integer is converted into a corresponding quantity expressed in mA units by the following relation:

$$x_{[\mathrm{mA}]} = \frac{AIW4 - 0.596}{1583.026}, \tag{1}$$

Corresponding value of the state in a percentage range is achieved by

$$x_{[\%]} = \frac{x_{[\mathrm{mA}]} - 10.4}{0.096}, \tag{2}$$

Fan speed Formula which relates actual control action $u_{[\%]}$, expressed in percentage rage, and a current signal output to the fan engine in mA units is

$$u_{[\mathrm{mA}]} = 0.2u_{[\%]}, \tag{3}$$

Therefore, corresponding integer representation of the output is

$$AQW0 = 1585.917u_{[\mathrm{mA}]} + 51.23, \tag{4}$$

where $AQW0$ is a physical address (A – analog, Q – output of the PLC, W – memory size of 16 bits, 0 – 1st output from PLC). This input signal is connected to the output of analog I/O module which is shown in Fig. 2.

The mathematical model of the fan airflow can be captured by one differential equation of the following form

$$\frac{\mathrm{d}V}{\mathrm{d}t} = kD_M^3 f - V \tag{5}$$

Here, V represents volume of air flow, k denotes a proportionality coefficient as a function of the Reynolds number of blender, D_M is blender diameter and f represents blender rotation frequency. By linearizing (5) around the steady state f^s and V^s, the following transfer function model can be derived

$$G = \frac{kD_M^3}{s+1} = \frac{Z}{T_v s + 1}, \tag{6}$$

where Z denotes gain of the system and T_v represents time constant.

Corresponding state-space representation of the fan is

$$\dot{x} = Ax + Bu \tag{7a}$$

$$y = Cx + Dy \tag{7b}$$

where $x = V - V^s$ is the state and $u = f - f^s$ is the input. Based on the steady state values of the variables mentioned above, the matrices are defined by

$$A = -\frac{1}{T_v},\ B = \frac{1}{Z},\ C = 1,\ D = 0.$$

This linear state-space representation can then be used to find a closed-form representation of the MPC feedback law by using techniques of parametric programming as described in Section 3.

2.2 PLC

A programmable logic controller (PLC) is a special digital computer often used in process automation such as for control of machinery on factory assembly lines, amusement rides, or lighting fixtures. Unlike general-purpose computers, PLCs are designed for multiple input and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are performed in constant length cycles (Siemens, 2008).

In this work, we have used the SIMATIC S7-200 micro PLC, which is exceptionally compact, remarkably capable, fast and comes with easy to operate hardware and software. It has a modular design, still open-ended enough to be expanded. The main components of the selected PLC are briefly described next.

CPU The S7-222 CPU can be seen on the left side in Fig. 2. Important to notice is that the CPU only provides 1024 bytes of memory for program data. This limit is both restrictive and challenging from the control synthesis point of view. Another limitation is that control algorithms have to be developed using so-called *ladder logic*, a visual programming language which only requires the algorithm to be composed of most basic operations (e.g. sums, products, comparisons, etc.).

Real-time data measurements can be stored on a memory cartridge (Siemens, 2008), marked by “MC” in Fig. 2. In our case it provides a 256 kB storage for measured signals, which could be captured at a 0.04 second sampling rate. The captured data can subsequently be opened in MS-Excel.

Power Source Power source (LOGOPower 6EP 1332-1SH42) (Siemens, 2010, 2008) is a standard transformer used to supply PLC from public power network (see Fig. 2), where the “Power Network” is 1-phase AC line supply with voltage rate of 100-240 V (50/60 Hz) to isolated output voltage 24 V DC.

Analog I/O Module Module EM 235 is necessary for the PLC to communicate with the controlled plant by means of analog signals. The module is shown in Fig. 2, where it is situated in between the central processor unit and the power source. The module allows 4 analog inputs and one analog output to be connected. Inputs have to be connected and configured corresponding way. The outputs in addition have to have correct HW configuration. For setup the outputs to the 0-20 mA we have to set a set of 6 switches. The set is placed right bottom corner of the EM 235 in Fig. 2. For our purpose we have set them as in Fig. 3. The PLC communicates with the master PC by a 9 pin RS 486 port, located under the memory cartridge.

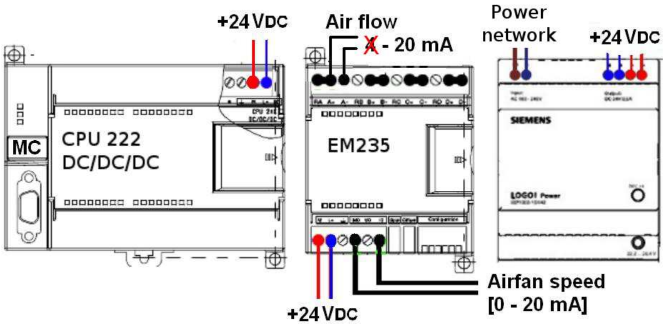


Fig. 2. PLC connections to the plant.

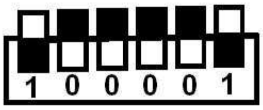


Fig. 3. Setting of the EM 235 inputs.

3. EXPLICIT MODEL PREDICTIVE CONTROL

In model predictive control (MPC), the optimal control actions are found by optimizing for plant behavior while taking process constraints into account. This is always achieved by formulating and solving an optimization problem where a given objective function is minimized subject to the constraints. A model of the plant is employed as an additional constraint to capture the predicted evolution of the plant:

We consider the discrete-time, stabilizable linear time-invariant model given by the state-space representation

$$x(k+1) = Ax(k) + Bu(k), \tag{8a}$$
$$y(k) = Cx(k). \tag{8b}$$

Here, $x(k)$ denotes state at time instance k, $x(k+1)$ is a successor state, $u(k)$ is control input, and $y(k)$ is a system output. It is assumed that variables are constrained by upper and lower limits

$$\underline{x} \le x(k) \le \overline{x}, \qquad \underline{u} \le u(k) \le \overline{u}, \qquad \underline{y} \le y(k) \le \overline{y} \tag{9}$$

For the system (10) consider now the constrained finite-time optimal control problem

$$\min_{\Delta u_0,\ldots,\Delta u_{N-1}} \sum_{k=0}^{N-1} \| R\Delta u_k \|_p + \| Q(y - y_{\text{ref}}) \|_p \tag{10a}$$
$$\text{s.t. } x_0 = x(t), \tag{10b}$$
$$x_{k+1} = Ax_k + Bu_k, \tag{10c}$$
$$y_k = Cx_k, \tag{10d}$$
$$x_k \in \mathcal{X}, \tag{10e}$$
$$y_k \in \mathcal{Y}, \tag{10f}$$
$$u_k \in \mathcal{U}. \tag{10g}$$

Here p denotes a matrix norm (either $p = 1$, $p = 2$ or $p = \infty$), the integer N represents the finite prediction horizon, and R, Q are weighting matrices used to tune performance of the MPC controller. The linear model in (10c) serves to predict the future states based on the knowledge of the initial state $x(t)$, which is assumed to be available at each time instance. The optimization is performed over the increments Δu_k to provide offset-free tracking of the reference trajectory y_{ref}. The state, output and input constraints represented, respectively, by the polyhedral sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{U}$.

MPC is usually implemented in so-called receding horizon fashion. Here the optimal solution to the problem (10) is found for particular value of $x(t)$, which results into the optimal sequence $[\Delta u_0^*, \ldots, \Delta u_{N-1}^*]$. Here $\Delta u_k = u_k - u_{k-1}$ is the control action increment, used to introduce integral action. Out of this sequence, only the first element (i.e. Δu_0^*) is actually implemented to the plant and the rest is discarded. At the next time instance, a new initial state measurement $x(t)$ is obtained and the whole procedure is repeated. This repetitive optimization is performed in order to introduce feedback control into the procedure and to deal with possible disturbances and plant model mismatches.

One can write new control action as $u_k = u_{k-1} + \Delta u_k$ which function of previous known control action and control action increment. The optimization problem (10) than can be rewritten into the state-space formulation with extended state feedback $\widetilde{x}_k = [x_k, u_{k-1}]^T$. Therefore, state-space representation reduces number of optimization variables from two (u_k and Δu_k) to one (Δu_k). The time-varying reference is constrained to be $y_{\text{ref},k+1} = y_{\text{ref},k}$, hence, state vector can be extended into three states $\widetilde{x}_k = [x_k, u_{k-1}, y_{\text{ref},k}]^T$ and state-space representation is defined in the form:

$$\widetilde{x}_{k+1} = \begin{bmatrix} A & B & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \widetilde{x}_k + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} \Delta u_k = \widetilde{A}\widetilde{x}_k + \widetilde{B}\Delta u_k \tag{11a}$$
$$y_k = [C \;\; D \;\; 0]\, \widetilde{x}_k + D\Delta u_k = \widetilde{C}\widetilde{x}_k + \widetilde{D}u_k \tag{11b}$$

with all the matrices of appropriate dimensions.

If the initial state $x(t)$ and the value of the reference signal y_{ref} are known, the problem (10) can be solved as a quadratic problem (QP) for $p = 2$ and for $p = 1$ or $p = \infty$ as a linear program (LP). Even though efficient polynomial-time algorithms exist to solve both type of problems, the time needed to perform the optimization can be prohibitive of the sampling time is too short, or if the implementation hardware is very simple and thus less capable. To address this issue, in their seminal work (Bemporad et al. (2002)) have shown (for a quadratic type of performance induces) how to solve the problem (10) parametrically for all admissible initial conditions $x(t)$ by employing techniques of parametric programming. In this approach the optimal solution to (10) is found as an explicit state feedback law parametrized in the initial condition $x(t)$. The advantage of the parametric solutions is that the optimal control input can be obtained in real-time by simply evaluating a look-up table. The main result of the parametric approach is summarized by the following theorem.

Theorem 3.1. (Bemporad et al. (2002)). The optimal solution to the problem (10) is a piecewise affine function of the initial state x_0

$$\Delta u_0^* = F_r x_0 + g_r \qquad \text{if} \qquad x_0 \in \mathcal{R}_r \tag{12}$$

where $\mathcal{R}_r = \{x_0 \mid H_r x_0 \leq K_r\}$ are polytopic regions of the state space, and F_r and g_r are the matrices of the affine state-feedback law active in the r-th region.

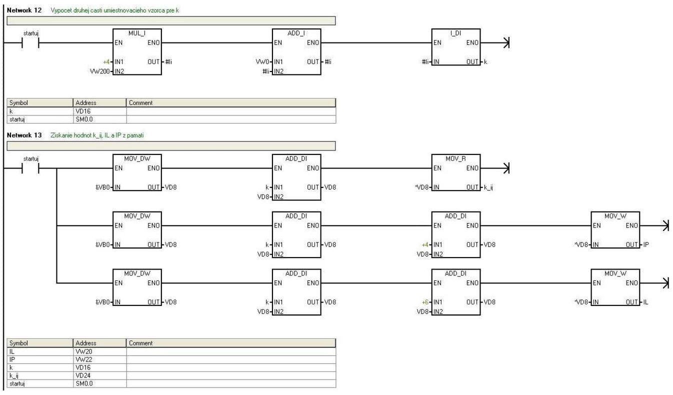
Theorem 3.1 shows that the optimal solution for the problem (10) can be found as a look-up table consisting of r components. Therefore, once the table is calculated, MPC can be implemented in a real time by simply evaluating the table for the actual measurements of $x_0 := x(t)$. The table can be calculated efficiently using, e.g. the Multi-Parametric Toolbox (Kvasnica et al., 2004). Performance of the MPC scheme can be tuned by appropriately adjusting the weighting matrices Q and R, and by a suitable choice of prediction horizon N.

4. IMPLEMENTATION ON PLC

As already mentioned, typical PLCs have severe memory limitations. Our PLC, in particular, only allows 1024 bytes of memory storage. A special care has thus to be taken when evaluating the explicit MPC feedback (encoded as a lookup table composed of feedbacks F_r, g_r, and regions $\mathcal{R}_r$) on such a device. To perform this task efficiently, we employ the *binary search tree* (BST) algorithm.

The basic idea of BST algorithm is to hierarchically organize the controller regions into a tree structure where, at level of the tree, the number of regions to consider is decreased by a factory of two. Therefore the table traversal can be performed in time logarithmic in the number of regions. The tree is constructed in an iterative fashion. At each iteration an optimal separating hyperplane $h_i x(t) \leq k_i$ is selected such that the set of all regions processed at the i-th iteration is divided into two smaller subsets: regions $\mathcal{R}_i^+$ residing on one side of the hyperplane and $\mathcal{R}_i^-$ on the other side. A new node in the tree is then created which contains information about the hyperplane and two pointers to child nodes. The left child is created by recursively calling the algorithm for regions $\mathcal{R}_i^+$ and the right child for the regions $\mathcal{R}_i^-$. The exploration of a given tree branch stops when no further subdivision can be achieved. In such a case a leaf is created which points to the region which contains $x(t)$. The resulting tree is then composed of the set of separating hyperplanes linked to the actual regions through a set of pointers.

To be able to use a BST-encoded tree on a PLC, the tree is transformed into a so-called "data-block". In this data table, first M entries represent one hyperplane and pointers to next line which should be explored. Obtaining the optimal control action for a particular value of x then reduces to traversing the binary search tree using Algorithm 1.



Network 12 Vypocet druhej casti umiestnovacieho vzorca pre k

Symbol	Address	Comment
k	VD16	
startuj	SM0.0	

Network 13 Ziskanie hodnot k_ij, IL a IP z pamati

Symbol	Address	Comment
IL	VW20	
IP	VW22	
k	VD16	
k_ij	VD24	
startuj	SM0.0	

Fig. 4. A short excerpt of the LAD implementation of Algorithm 1

Algorithm 1 Table traversal via binary search tree

```
INPUT: BST tree composed of separating hyperplanes $h_i x \leq k_i, i = 1, \ldots, M$ and linked nodes, state measurements $x(t)$
OUTPUT: Optimal control input $u^*(x)$
$r \leftarrow 1$
repeat
    if $h_r x \leq k_r$ then
        $r \leftarrow$ index of the left child node (negative index)
    else
        $r \leftarrow$ index of the right child node (positive index)
    end if
until $r$ is a leaf node (positive index).
$u_0^*(x(t)) = F_r x(t) + g_r$
```

The PLC version of Algorithm 1, implemented using the Ladder Logic (LAD) programming language, is universal and can process any kind of lookup tables described by binary search trees. The LAD diagram consists of several routines and subroutines, a short excerpt of which is shown in Fig. 4. The program allocates 74 bytes of global memory in main routine and at most 34 bytes of temporary memory in subroutines. The total amount of memory allocated for controller is 874 bytes, the rest (150 bytes) remains to user.

5. EXPERIMENTAL RESULTS

In this section we show how MPC could be used for control of the fan heater described in the Section 2 using PLC. The control objective is to drive the volume of air flow to a time varying reference y_{ref} while respecting motor capacity $0\% \leq f \leq 100\%$ and volume of the air flow $0\% \leq V \leq 100\%$. The following mathematical model of the fan heater was obtained using identification methods:

$$G = \frac{5.12}{0.4726s + 1} \mathrm{e}^{-0.3s}. \tag{13}$$

MPC synthesis using the Multi-Parametric Toolbox begins with a definition of the prediction model:

```
>> A=-0.1953; B=0.0923; C=1; D=0;
>> fan=ss(A,B,C,D)
>> Ts=0.25
>> model=mpt_sys(fan,Ts)
```

where the model is converted into the discrete-time domain using sampling time T_s. Time delay of the model can be omitted as it is less than the time constant and the real time verification proved such model to be satisfactory. Next, constraints on state, input and output are defined:

```
>> model.umax=100-fs; model.umin=0-fs;
>> model.xmax=100-Vs; model.xmin=0-Vs;
>> model.ymax=100-Vs; model.ymin=0-Vs;
```

Notice that the constraints are imposed on the deviation variables with linearization points $f^s = 15\%$ and $V^s = 30\%$.

Once the model is complete, parameters of the MPC problem to be solved could be defined by

```
>> problem.R=1; %penalty on u_k
>> problem.Q=1; %penalty on x_k
>> problem.Qy=1000; %penalty on (y_k-y_ref)
>> problem.N=5; %prediction horizon
>> problem.norm=2; %use quadratic cost
>> problem.tracking=1; %use time-varying
    reference
```

Values of the penalty matrices R and Q_y were chosen with respect to allowed number of regions (which reflect the memory footprint of the controller). The number of regions can be reduced by lowering R and increasing the value of Q_y. The upper bound on the number of regions is 26 for a controller with 1 state and 1 input, otherwise the controller footprint would exceed 1 kB.

Finally, the parametric solution to problem (10) can be calculated as a lookup table using the command

```
>> ctrl=mpt_control(model,problem)
```

Result of the composition is, in this case, a lookup table consisting of 25 regions in a 3D state-space. State-space representation used in the controller consist of the following matrices

$$\widetilde{A} = \begin{bmatrix} 0.8226 & 0.3199 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \widetilde{B} = \begin{bmatrix} 0.3199 \\ 0 \\ 0 \end{bmatrix} \tag{14a}$$

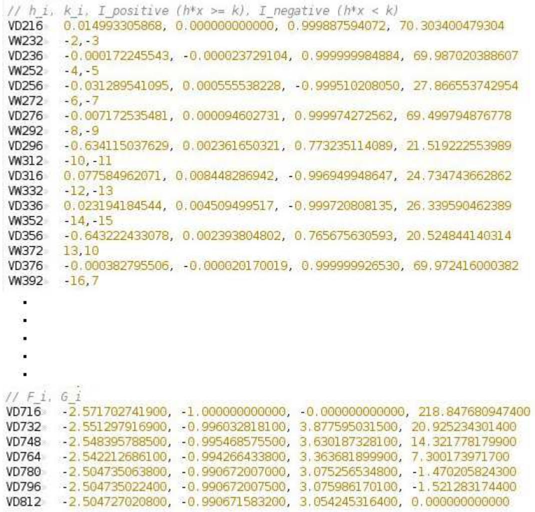
$$\widetilde{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \widetilde{D} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{14b}$$

BST tree is constructed from the lookup table using the MPT command

```
>> tree=mpt_searchTree(ctrl)
```

In our case, tree consists of 25 nodes in 7 levels, which corresponds to 724 bytes of memory. Selected parts of the data-block are depicted in Fig. 5. The number of values in one line corresponds with number of the state variables and constant.

The data are subsequently downloaded to the PLC, which then executes the table traversal at each sampling instance based on the measurements of the states. When the region for the actually measured state is found, Algorithm 1 is executed and the corresponding control input to the system is calculated as



```
// h_i, k_i, I_positive (h*x >= k), I_negative (h*x < k)
VD216   0.014993305868, 0.000000000000, 0.999887594072, 70.303400479304
VW232   -2,-3
VD236   -0.000172245543, -0.000023729104, 0.999999984884, 69.987020388607
VW252   -4,-5
VD256   -0.031289541095, 0.000555538228, -0.999510208050, 27.866553742954
VW272   -6,-7
VD276   -0.007172535481, 0.000094602731, 0.999974272562, 69.499794876778
VW292   -8,-9
VD296   -0.634115037629, 0.002361650321, 0.773235114089, 21.519222553989
VW312   -10,-11
VD316   0.077584962071, 0.008448286942, -0.996949948647, 24.734743662862
VW332   -12,-13
VD336   0.023194184544, 0.004509499517, -0.99720808135, 26.339590462389
VW352   -14,-15
VD356   -0.643222433078, 0.002393804802, 0.765675630593, 20.524844140314
VW372   13,10
VD376   -0.000382795506, -0.000020170019, 0.999999926530, 69.972416000382
VW392   -16,7
  .
  .
  .
  .
  .
// F_i, G_i
VD716   -2.571702741900, -1.000000000000, -0.000000000000, 218.847680947400
VD732   -2.551297916900, -0.996032818100, 3.877595031500, 20.925234301400
VD748   -2.548395788500, -0.995468575500, 3.630187328100, 14.321778179900
VD764   -2.542212686100, -0.994266433800, 3.363681899900, 7.300173971700
VD780   -2.504735063800, -0.990672007000, 3.075256534800, -1.470205824300
VD796   -2.504735022400, -0.990672007500, 3.075986170100, -1.521283174400
VD812   -2.504727020800, -0.990671583200, 3.054245316400, 0.000000000000
```

Fig. 5. A short excerpt of the data-block provided on the PLC.

$$u_k = u_{k-1} + \Delta u^* \tag{15}$$

with state-feedback law $\Delta u^* = F_i\widetilde{x}_k + g_i$, where $\widetilde{x}_k$ is state variable of the system (11).

The data-block representing the controller was downloaded to the PLC to perform real-time experiments. First, ability of the controller to follow a time-varying reference, where user can change a setpoint at any time, is documented in Figs. 6–7.

System response near the and upper bound is without oscillations within a reasonable settling time, while response around the middle and near the lower bound has longer settling time. Such response can be caused by different behavior of the plant throughout the state ranges. That means, several models are necessary to describe plant behavior sufficiently. Therefore, possibility how to eliminate oscillations, is to control the plant as a hybrid system, which is not possible due to restricted amount of the memory.

To reduce long settling time, one can approximate time delay in model (13) by Taylor series or Padé approximation. This approach results in the better but more complex model with more regions, thus impossible to apply on PLC. The maximum amount of regions is function of the number of state variables. Therefore, if one wants more regions, either a simpler model is required or region reduction techniques have to be employed (Kvasnica et al., 2011; Kvasnica and Fikar, 2010).

Real data presented in Figs. 6–7 show that the MPC controller utilizes the predictions to change the value of the input signal in the same period as the reference was changed, such that output signal is steered towards this reference. Experiment also proved that MPC controller with time-varying reference can be implemented on the PLC in a real time.

6. CONCLUSIONS

In this paper we have shown how MPC can be implemented on a programmable logic controller with severe limitations on allowed memory storage. The approach was

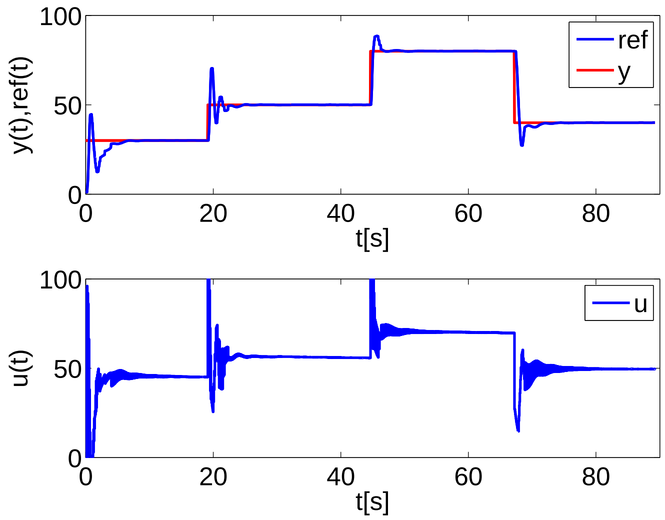


Fig. 6. Control of the fan heater tracking time-varying reference.

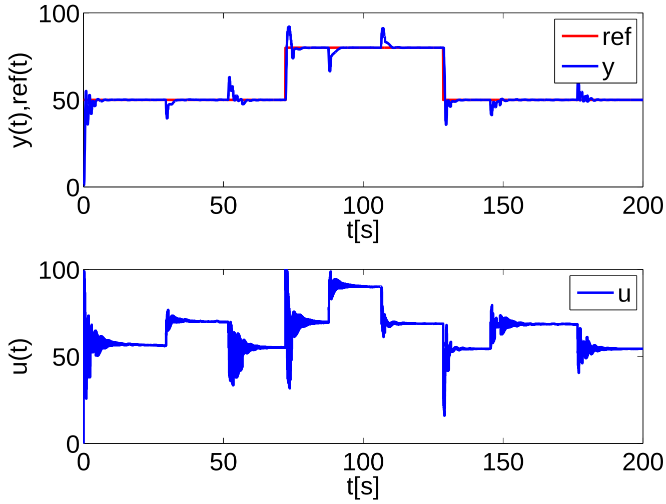


Fig. 7. Control of the fan heater with disturbance during the tracking of time-varying reference.

based on the pre-calculating the solution to the MPC optimization problem just once, for all possible initial conditions. The result is then given in the form of a lookup table. Such a table was subsequently encoded as a binary search tree for its efficient evaluation in real time. Experimental results confirm that the controller provides satisfactory performance while respecting design constraints.

ACKNOWLEDGMENT

Authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11, and the Slovak Research and Development Agency under the project APVV-0029-07. This work was also financed by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Kvasnica, M. and Fikar, M. (2010). Performance-lossless complexity reduction in explicit MPC. In *Proceedings of the 49th IEEE Conference on Decision and Control*, 5270–5275. Atlanta, USA.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/˜mpt/.

Kvasnica, M., Löfberg, J., and Fikar, M. (2011). Stabilizing polynomial approximation of explicit MPC. *Automatica*. (accepted).

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall.

Siemens (2008). *S7-200 Programable Controller System Manual*.

Siemens (2010). *Betriebsanleitung Nr.:C98130-A7560-A2-6419*.

Svetíková, M., Annus, J., Čirka, Ľ., and Fikar, M. (2003). Real time control of a laboratory fan heater using dspace tools. In *Proceedings of the 14th International Conference Process Control '03*. Štrbské Pleso, High Tatras (Slovakia).

H∞ Controler Design for Active Suspension System

Monika Zuščíková, Cyril Belavý

*Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering,
Slovak University of Technology, Nám. Slobody 17, 812 31 Bratislava , Slovak Republic,
Tel/Fax: ++421/2/52495315
e-mail: monika.zuscikova@stuba.sk*

Abstract: This paper presents the *H*∞ synthesis of control for an active suspension design based on an extended quarter-car model. The usage of automobile active suspension has two main reasons, to increase ride comfort and to improve handling performance. Both this requirements are contradictory. To obtain the model performances and solve the *H*∞ synthesis the Matlab software with the Robust Control Toolbox has been used. The benefits of controlled active suspensions compared to passive ones are here emphasized.

1. INTRODUCTION

Vehicle suspension has been a hot research topic for many years due to its important role in ride comfort, vehicle safety, road damage minimization and the overall vehicle performances. To meet these requirements, many types of suspension systems, ranging from passive, semi/active, to active suspensions, are currently being employed and studied. It has been well recognized that active suspension has a great potential to meet the tight performance requirements demanded by users. Therefore, in recent years more and more attention has been devoted to the development of active suspensions and various approaches have been proposed to solve the crucial problem of designing a suitable control law for these active suspension systems. In many control applications, it is expected that the behaviour of the designed system will be insensitive (robust) to external disturbance and parameter variations. It is known that feedback in conventional control system has the inherent ability of reducing the effects of external disturbances and parameter variations. In this paper, the H∞ control design problem is converted into a convex optimization problem described by linear matrix inequalities LMI, Zhou (1998).

The *H*∞ method addresses a wide range of the control problems, combining the frequency and time-domain approaches. The design is an optimal one in the sense of minimization of the *H*∞ norm of the closed-loop transfer function. The *H*∞ model includes coloured measurement and process noise. It also addresses the issues of robustness due to model uncertainties, and is applicable to the SISO system as well as to the MIMO system, Gawrovski (2004). In this paper is present the *H*∞ control design for quarter-car active suspension system.

2. THE SUSPENSION MODEL

The usually used quarter-car model has two degrees-of-freedom see Fig.1. It includes the vertical motion of the sprung mass m_2 which represents the car body with passengers and the unsprung mass m_1 which corresponds to the mass of the wheel and suspension. The disturbance input *w* is the road profile. x_1 represent the positions of the sprung mass and x_2 the positions of the unsprung mass.

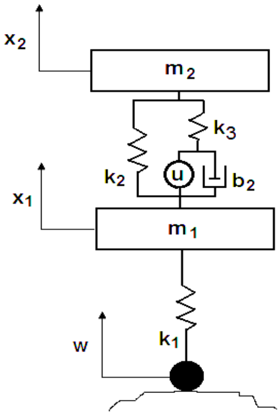


Fig. 1. Extended quarter car with active suspension

Table 1: The values of Parameters in quarter-car

Description	Units	Values
Body (sprung) Mass	m_2 (kg)	350
Axle (unsprung) Mass	m_1 (kg)	35
Suspension Stiffness	k_1 (N/m)	200 000
Suspension Stiffness	k_2 (N/m)	14 000
Tire Damping	b_1 (Ns/m)	500
Tire Damping	b_2 (Ns/m)	1600
Damper Stiffness	k_3 (N/m)	250 000

2.1 Rheological damper model

Usually the suspension is modelled by means of a linear damper and a spring. However also the real spring has basically a linear characteristic, the real damper has a nonlinear and a considerable hysteresis caused primarily by the oil compressibility (bulk modulus – β=0,8 (Pa)). These properties can by well modelled by means of the Maxwell element Fig.2., Guglielmino (2004).

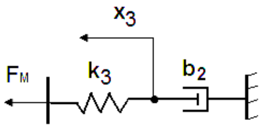


Fig. 2. Rheological damper model (Maxwell element)

The spring k_3 represents the mentioned stiffness of the dampers hydraulics circuit and can by calculated as

$$k_3 = \frac{\beta \cdot S_p^2}{V} = \frac{\beta \cdot \pi^2 d_p^4}{16\ V} \tag{1}$$

where d_p = 0,022 (m) is the diameter damper rod and $V \approx$ 0,0003 (m^3) is the mean volume of the damper pressure and expanse chambers.

The rheological damper properties for different damper values b_2 are shown in Fig 3.

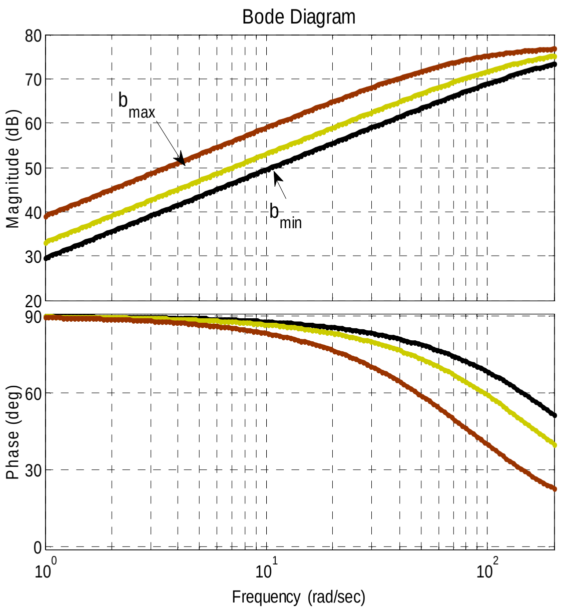


Fig. 3. Characteristic of a rheological damper model

2.2 State space modeling

The state space representation of the controlled system of an extended quarter car model Fig.3 can be formalized as following:

$$\begin{aligned} \dot{x} &= Ax \ + B_1 w + B_2 u \\ y &= C_1 x + D_{11} w + D_{12} u \\ z &= C_2 x + D_{21} w + D_{22} u \end{aligned} \tag{2}$$

where the state vector x, output vector y and vector of measurement z are defined as following:

$$x = [x_1 - w_1 \quad x_2 - x_1 \quad \dot{x}_1 \quad \dot{x}_2 \quad \dot{x}_3]^T \tag{3}$$

$$y = [\ddot{x}_2 \quad x_2 - x_1 \quad F_{dyn}]^T \tag{4}$$

$$z = [\dot{x}_1 \quad \dot{x}_2 \quad x_2 - x_1]^T \tag{5}$$

The state space matrices are defined following:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ -\frac{k_1}{m_1} & \frac{k_2}{m_1} & 0 & 0 & -\frac{k_3}{m_1} \\ 0 & -\frac{k_2}{m_2} & 0 & 0 & \frac{k_3}{m_2} \\ 0 & 0 & 1 & -1 & -\frac{k_3}{b_2} \end{bmatrix}, \tag{6}$$

$$B_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \ B_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{1}{b_2} \end{bmatrix}, \tag{7}$$

$$C_1 = \begin{bmatrix} 0 & -\frac{k_2}{m_2} & 0 & 0 & \frac{k_3}{m_2} \\ 0 & 1 & 0 & 0 & 0 \\ \frac{k_1}{m_1} & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{8}$$

$$D_{11} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ D_{12} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{9}$$

$$C_2 = \begin{bmatrix} -\frac{k_1}{m_1} & \frac{k_2}{m_1} & 0 & 0 & -\frac{k_3}{m_1} \\ 0 & -\frac{k_2}{m_2} & 0 & 0 & \frac{k_3}{m_2} \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{10}$$

$$D_{21} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \ D_{22} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{11}$$

2.3 Suspension performances and weighting filters

In this paper, the following performance aspects of quarter-car suspension system are taken into account:

1. Ride comfort – can be quantified by the car body acceleration $\ddot{x}_2$
2. Suspension deflection limitation – the travel space does not need to be minimal but its peak value need to be constrained.

$$\left| x_2(t) - x_1(t) \right| \le x_c \qquad (12)$$

3. Road holding ability – in order to ensure a firm uninterrupted contract of wheels to road, the dynamic tyre.

The feedback structure is shown in Fig. 4. It includes the input W_1 and output W_2 weighting functions, the extended quarter car model *P(s)* and the controller model *K(s).*

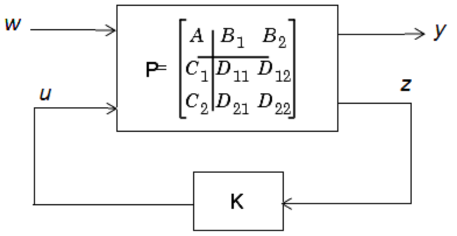

w
y
u
z
P=
$\begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}$
K

Fig. 4. The active suspension control scheme

The input weight (13) includes the road disturbance filter and the weight for the actuator force.

$$W_1 = \begin{bmatrix} W_{road}(s) & 0 \\ 0 & 1/u_c \end{bmatrix}, \qquad (13)$$

$$W_{road}(s) = A_w\, W_{Butter\,0{,}5Hz}(s) \qquad (14)$$

Where the constants A_w represents the power of chosen road type and the $W_{Butter0{,}5Hz}$ represents a classics high pass Butterworth analogue filter with a cut-off frequency 0,5(Hz). u_c represents the value of the critical force produced by the controlled actuator.

The output weights (14) for the optimized values *y* and for the measured values z are: the matrix of weighting functions is chosen as:

$$W_2 = \begin{bmatrix} W_y & 0 \\ 0 & W_z \end{bmatrix}, \qquad (15)$$

$$W_y = \begin{bmatrix} W_{acc}/a_c & 0 & 0 \\ 0 & 1/x_c & 0 \\ 0 & 0 & 1/F_{Dc} \end{bmatrix}, \quad W_z = I \qquad (16)$$

where W_{acc} is the weighting filter of acceleration, definite in norm ISO 2631.

$$W_{acc}(s) = \frac{num_w}{den_w} \qquad (17)$$

$$num_w = [87{,}72 \quad 1138 \quad 11336 \quad 5453 \quad 5509]$$
$$den_w = [1 \quad 92{,}6854 \quad 2549{,}83 \quad 25969 \quad 81057 \quad 79783]$$

Where the values represent: a_c – critical weighted acceleration acting on the human body chosen from the ISO 2631, x_c – critical suspension deflection given by the suspension design and F_{Dc} – critical dynamic tyre force gravity of the static weight which is acting on the tyre. Dividing each of the optimized parameters with his critical value, we are used normalization and so the weighted and normalized optimized parameters will have no units.

The magnitude frequency characteristics of the road and sprung mass acceleration filters are shown in Fig.5.

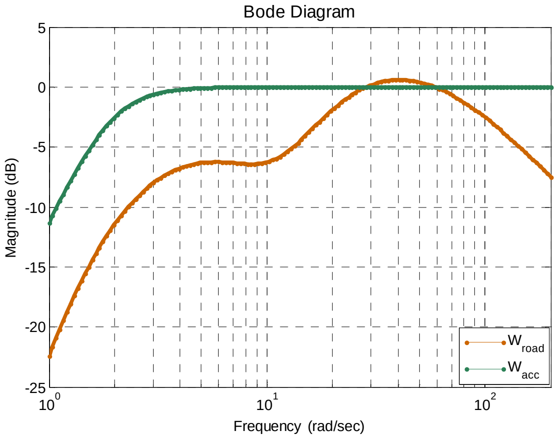


Fig. 5. Bode plot of W_{road} and W_{acc} filters

3. *H*∞ CONTROLLER DESING

When open loop is denoted $T_{yw,}$, then a standard optimal *H*∞ controller problem is to find admissible controller *K* such that $\left\| T_{yu} \right\|_\infty$ is minimal. The problem of finding a suboptimal *H*∞ controller can be formulated: for given γ > 0 find all admissible controllers *K*, they exits, such that

$$\left\| T_{yu} \right\|_\infty < \gamma \qquad (18)$$

3.1 Solution

The solution of this problem requires the solving of two Ricatti equations, one for controller and one for the observer, Gawrovski (2004).

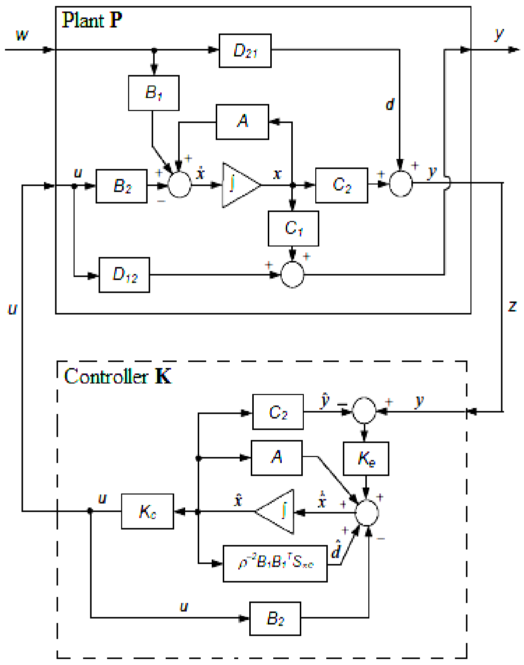


Fig. 6. The central *H*∞ closed-loop system

The control law is given by

$$u = -K_c\hat{x} \tag{19}$$

and the state estimator equation by

$$\dot{\hat{x}} = Ax + B_2u + B_1\hat{w} + K_e(y - \hat{y}) \tag{20}$$

where

$$\hat{w} = \gamma^{-2}B_1^T X_\infty\hat{x} \tag{21}$$

$$\hat{y} = C_2\hat{x} \tag{22}$$

The controller gain is K_c as for the LQG case, and the estimator gain is $Z_\infty K_e$ instead of K_e as for the LQG case, with

$$K_c = B_2^T X_\infty \tag{23}$$

$$K_e = Z_\infty Y_\infty C_2^T \tag{24}$$

$$Z_\infty = (I - \gamma^{-2}Y_\infty X_\infty)^{-1} \tag{25}$$

The terms X_∞ and Y_∞ are solutions to controller and estimator Ricatti equations

$$X_\infty = Ric\begin{bmatrix} A & \gamma^{-2}B_1B_1^T - B_2B_2^T \\ -C_1^TC_1 & -A^T \end{bmatrix} \tag{26}$$

$$Y_\infty = Ric\begin{bmatrix} A^T & \gamma^{-2}C_1^TC_1 - C_2^TC_2 \\ -B_1B_1^T & -A \end{bmatrix} \tag{27}$$

We do not carry out these calculations by hand – the tools supplied by Matlab *Robust Control Toolbox* just do that.

4. SIMULATION RESULTS

In the next chapter are results in frequency and time domain compared. The results have bean solved using the model shown in the Fig. 1 and its parameters are stated in the Tab. 1.

4.1 Frequency Response Simulations

In the next figures the performance magnitudes of the considered active and passive vehicle suspensions are compared. In Fig.7 the weighted acceleration of the car body is shown. We can read that the active suspension system has better comfort performances from all around the first system eigenfrequency. After the second eigenfrequency the performance of the passive system is better but that is not so important region of frequencies for the comfort criterion an also in real model it is very difficult to control vibrations at so high fervencies. So the active system would acting anyway like a passive one.

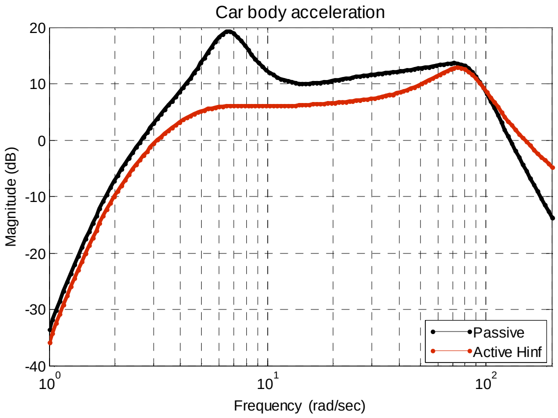


Fig. 7. Bode plot of the weighted and normalized car body vertical acceleration – Comfort criterion

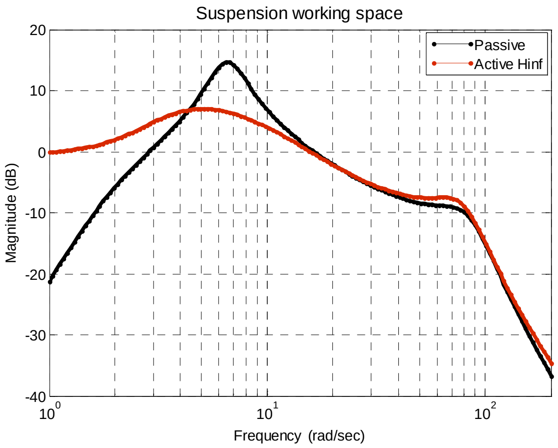


Fig. 8. Bode plot of the normalized suspension deflection – Reliability criterion

In the Fig. 8 the frequency response of the suspension deflection is shown. However there is an increase of the deflection magnitude on the active suspension according to the magnitude of the passive suspension, but at this criterion the most important thing is the maximal value in the whole region of the frequencies - H_∞ norm and this criterion is significantly better achieved with the active suspension.

At the last frequency response Fig. 9 is magnitude of the normalized dynamic tyre force. Here we can see that a significantly improvement by means of the active suspension was achieved and that from all at the first system eigenfrequency.

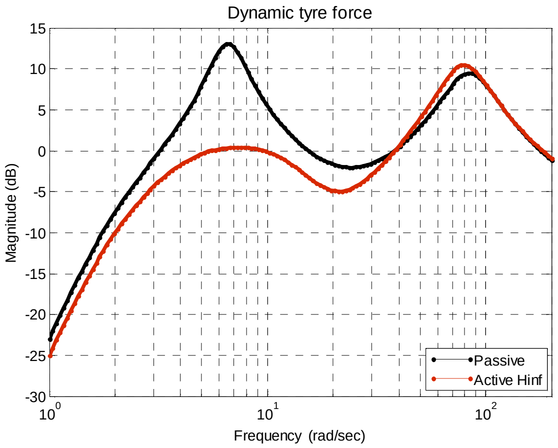


Fig. 9. Bode plot of the normalized dynamics tyre force – Road holding criterion

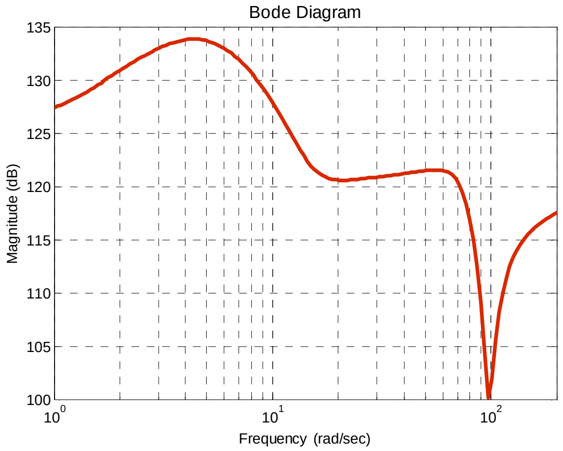


Fig. 10. Bode plot of the actuator

4.2 Time response simulations

Also a time response has been calculated which shows how the passive and active suspension systems are responding by crossing a road bump disturbance see Fig. 11 – 14.

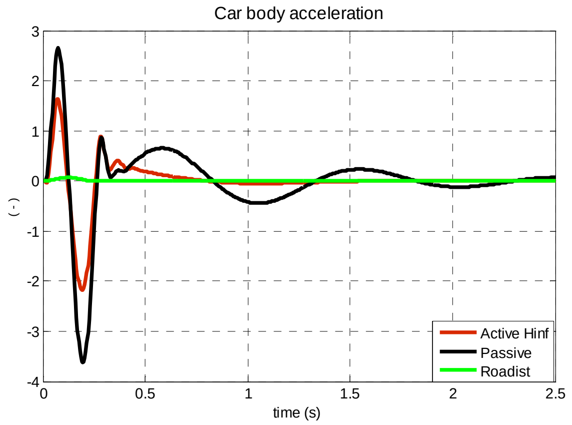


Fig. 11. Time response of the vertical acceleration (active, passive suspension and road disturbance)

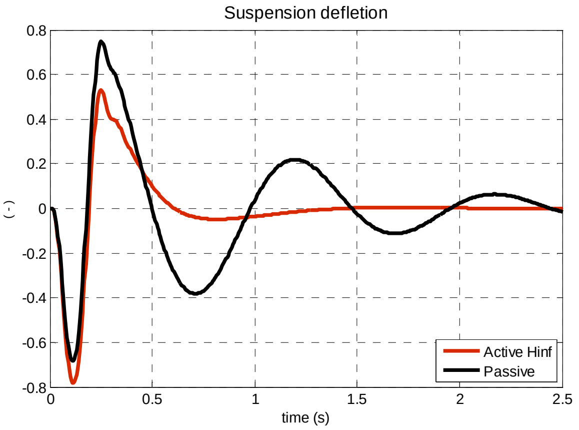


Fig. 12. Time response of the suspension deflection

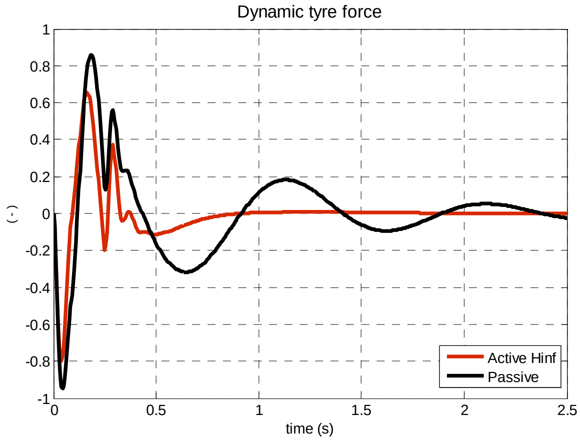


Fig. 13. Time response of dynamic tyre force

In the next table the suspensions performance values calculated via the H_2 and $H\infty$ norms from the previous time responses are shown.

Table 2: Performance values of passive and active suspension

Suspension performances (-)	passive H_2	active H_2	passive $H\infty$	active $H\infty$
Car body acceleration	0.7476	0.4342	3.6275	2.1830
Suspension deflection	0.2509	0.1824	0.7866	0.7434
Dynamic tire force	0.2420	0.1517	0.9494	0.7974

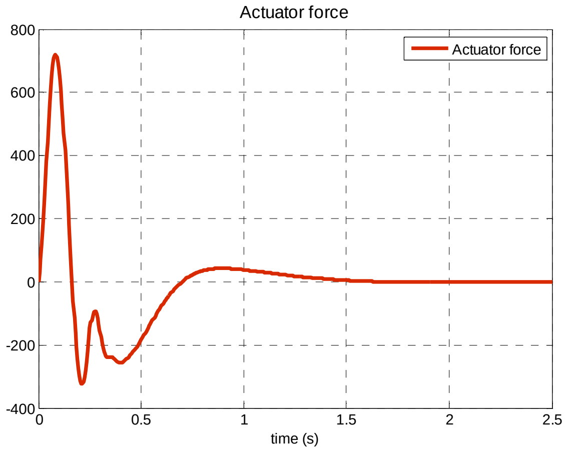


Fig. 14. Time response of actuator force

CONCLUSION

From the simulations results we can clearly see that the active controlled suspension with $H\infty$ controller offers a much better suspension performances as the classics passive suspension model. These results have been confirmed also even if we have extended the simple quarter car model with the damper stiffness which has brought one more degree of freedom into the system and also mead the simulation model more realistic.

ACKNOWLEDGMENTS

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0160-07 for project “*Advanced Methods for Modeling, Control and Design of Mechatronical Systems as Lumped-input and Distributed -output System*”.

REFERENCES

Wodek K. Gawrovski (2004). *Advanced Structural Dynamics and Aktive Control of Structurec* Springer-Verlag New York, Inc.

Yamashita, M., Fujimori, K., Hayakawa, K., and Kimura, H. (1994*). Application of H-infinity Control to Active Suspension Systems. Automatica*, 30(11): 1717-1792.

K. Zhou, J.C.Doyle, and K. Glover (1996). *Robust and Optimal Control*. Prentice-Hall, Upper Saddle River

Zhou, K. and Doyle J.C. (1998). *Essentials of Robust Control* Prentice Hall.

Bruce A. Francis (1987), *Lecture Notes in Control and Information Sciences*, Springer – Verlag Berlin Heidelberg,

P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali(1995) *LMI Control Toolbox for Use with MATLAB*. The MatWorks Inc.

Rajamani, R., (2006) *Vehicle Dynamics and Control*, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

Emanuele Guglielmino, Tudor Sireteanu, Charles W. Stammers, Chorghe Ghita, Marius Giuclea, (2004) *Semi-active Suspension Control*, London

DECOUPLING OPTIMAL CONTROLLERS

V. Kučera*

** Czech Technical University in Prague, Faculty of Electrical Engineering,*
Technická 2, 16627 Prague 6, Czech Republic
fax : +420-224 916 648 and e-mail : kucera@fel.cvut.cz.

Abstract: The problem of decoupling a linear system by dynamic compensation into multi-input multi-output subsystems is studied by applying proper and stable fractional representations of transfer matrices. A necessary and sufficient condition is given for a decoupling and stabilizing controller to exist. The set of all controllers that decouple and stabilize the system is determined in parametric form. Decoupling optimal controllers are then obtained by an appropriate selection of the parameter.

Keywords: Linear systems, fractional representations, decoupling controllers, stabilizing controllers, optimal controllers.

1 INTRODUCTION

Decoupling is a way to decompose a complex system into non-interacting subsystems. In fact, certain applications necessitate controlling independently different parts of the system. Even if this is not required, the absence of interaction can significantly simplify the synthesis of the desired control laws.

The decoupling problem has received much attention in the literature. For linear systems, different approaches have been used and control laws of various structure and complexity applied.

The basic form of decoupling into single-input single-output subsystems is often referred to as the diagonal decoupling. This problem was posed by Voznesenskij (1936) and studied by Kavanagh (1957), Strejc (1960), Mejerov (1965), and Wolovich (1974). The studies were related to the inversion problem of rational matrices. Attention was paid to the existence of proper rational transfer matrices. The issue of stability, however, was not properly addressed.

A deeper insight was provided by the state-space approach. The pioneering work is due to Morgan (1964), who posed the problem of decoupling by static state feedback. Falb and Wolovich (1967) established a solvability condition while Gilbert (1969) related this condition to state feedback invariants of the system. Descusse and Dion (1982) then interpreted this condition in terms of system’s structure at infinity.

The use of restricted static state feedback, namely the static output feedback, in decoupling was studied by Howze and Pearson (1970), Howze (1973), Denham (1973), Hazlerigg and Sinha (1978), Filev (1982b), Descusse and Malabre (1982), and Descusse, Lafay and Kučera (1984). This is a very restricted problem, whose solution is hard to obtain, but it is very useful in applications.

A more general form of decoupling into multi-input multi-output subsystems is referred to as the block decoupling. This problem was introduced by Wonham and Morse (1970) and Basile and Marro (1970). Using a geometric approach, they determined the solvability of the problem by static state feedback in several special cases. An alternative algebraic approach based on the structure algorithm was presented by Silverman and Payne (1971). Relationships between the two approaches were studied by Filev (1982a).

The decoupling by dynamic state feedback was studied via the geometric approach by Morse and Wonham (1970), who obtained a deep insight into the internal structure of the decoupled system. By this time, the problem of decoupling by dynamic state feedback was solved, including stability or pole distributions that may be achieved while preserving a decoupled structure. The status of noninteracting control was reviewed by Morse and Wonhan (1971).

A comeback of the transfer function methods in the study of block decoupling is witnessed through the works of Koussiouris (1979), Hautus and Heymann (1983), and Kučera (1983). A dynamic state feedback was shown to be equivalent with combined dynamic output feedback and feedforward reference compensation, often referred to as a two-degree-of-freedom controller. To address stability issues, the Youla-Kučera parameterization of all stabilizing controllers was invoked. The basic results are reported by Kučera (1983), Hautus and Heymann (1983), and Gómez and Goodwin (2000). The class of all decoupled transfer matrices that can be achieved by a stabilizing controller was parameterized by Desoer and Gündeş (1986) and Lee and Bongiorno (1993). This result has made it possible to derive the H_2-optimal decoupling controller, which minimizes the performance deterioration due to decoupling.

The two-degree-of-freedom controller structure is ideally suited to decoupling since only one of the degrees of freedom is affected by the decoupling requirement. This is not true for a pure feedback, or a one-degree-of-freedom controller. This case is considerably more difficult to solve, as shown by Hammer and Khargonekar (1984), Lin (1997), Youla and Bongiorno (2000), Bongiorno and Youla (2001), and Park (2008a).

Finally, the decoupling in the generalized plant model, which covers a broad range of control problems in a unified setting, was considered by Park (2008b). Such a plant model can accommodate non-square plant and non-unity feedback cases with one-degree-of-freedom or two-degree-of-freedom controller configuration. The benefits of such a general problem formulation consist in a unified treatment rather than in simplicity of the solution. Indeed, matrix operations need to be converted to vector operations with vectors of a much larger dimension, which result from the Kronecker and Khatri-Rao products of matrices.

This paper adopts the most general setting that is meaningful for decoupling: a system in which the measurement output may be different from the output to be decoupled and a dynamic controller that features both feedback and feedforward parts. The class of all such controllers that decouple and stabilize the system is determined in parametric form and the parameter is used to obtain the H_2-optimal controller. The solution is simple and direct. The controller configuration implies that decoupling and stability are two independent issues.

2 PROBLEM FORMULATION

Consider a linear, time-invariant, differential system governed by the input-output relation

$$y = S_y u, \qquad (1)$$

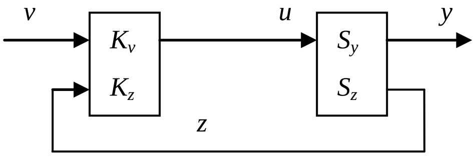


Fig. 1. Control system

where u is the q-vector input, y is the p-vector output and S_y is the transfer matrix of the system. We assume that S_y is a proper rational matrix over R(s), the field of rational functions.

Let p_1, ..., p_k be a given set of positive integers that satisfy

$$\sum_{i=1}^{k} p_i = p\ .$$

System (1) is said to be *decoupled*, or more specifically (p_1, ..., p_k)-decoupled, if there exist positive integers q_1, ..., q_k satisfying

$$\sum_{i=1}^{k} q_i = q$$

such that S_y has the block diagonal form

$$S_y := \begin{bmatrix} S_1 & & \\ & \ddots & \\ & & S_k \end{bmatrix},$$

where S_i is $p_i \times q_i$.

This is not a generic property of the system, but it can be achieved by a suitable compensation. To this effect, let z denote the m-vector output of the system that is available for measurement and let it be related with the input by the equation

$$z = S_z u\ , \qquad (2)$$

where S_z is a proper rational matrix over R(s).

The most suitable linear, time-invariant, differential controller can then be described by the equation

$$u = K_v v + K_z z\ , \qquad (3)$$

where v is an external reference input of appropriate dimension, say r. As it is seen in Fig. 1, the transfer matrices K_v and K_z represent the feedforward and the feedback parts of the controller, respectively. We assume that both K_v and K_z are proper rational matrices over R(s).

The *decoupling problem* is then to find matrices K_v and K_z such that the transfer matrix

$$T = S_y (I - K_z S_z)^{-1} K_v \qquad (4)$$

from v to y be suitably block diagonal.

Obviously, unless additional provisions are made, the decoupling problem is trivial as it could be solved by $K_v = 0$. Thus it is necessary to impose certain admissibility condition on the decoupling controller to make the problem meaningful, for example

$$\operatorname{rank} T = \operatorname{rank} S_y \tag{5}$$

over R(s). This condition is equivalent to the preservation of the class of controlled output trajectories. We thus require that no essential loss of control occurs through the decoupling process.

Another requirement, frequently imposed on the decoupled system in practice, is that of stability. This requirement means that the states of the system go to zero from any initial values.

3 PRELIMINARIES

A stable system gives rise to a proper and stable transfer function. In order to study stability of the decoupled system it is convenient to express the transfer matrices of the given system and those of the controller in the following factorized form

$$\begin{bmatrix} S_z \\ S_y \end{bmatrix} := \begin{bmatrix} B \\ C \end{bmatrix} A^{-1}$$

$$\begin{bmatrix} K_z & K_v \end{bmatrix} := P^{-1} \begin{bmatrix} -Q & R \end{bmatrix},$$

where

$$A, \begin{bmatrix} B \\ C \end{bmatrix}$$

are proper and stable rational matrices that are right coprime and

$$P, \begin{bmatrix} -Q & R \end{bmatrix}$$

are proper and stable rational matrices that are left coprime.

These proper and stable fractional representations exist and are unique up to right and left multiplication, respectively, by a unimodular matrix. Recall that a proper and stable rational matrix is said to be unimodular if its inverse exists and is proper and stable.

The system equations (1) and (2) and the controller equation (3) then take the form

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} B \\ C \end{bmatrix} A^{-1} u \,, \tag{6}$$

$$u = P^{-1} \begin{bmatrix} -Q & R \end{bmatrix} \begin{bmatrix} z \\ v \end{bmatrix}. \tag{7}$$

The overall system transfer function reads

$$T = C(PA + QB)^{-1} R \,. \tag{8}$$

The fundamental assumption we make here is that the part of the given system that is not controllable from u is stable and the part of the given system that is not jointly observable from y, z is stable. Similarly, we assume that the controller is realized in such a manner that its part that is not jointly controllable from v, z is stable and its part that is not observable from u is stable.

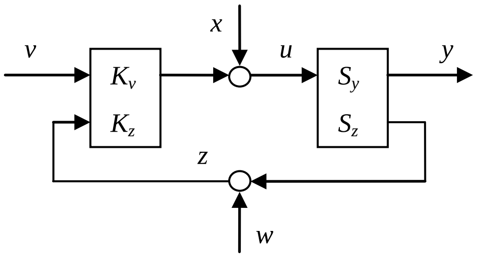


Fig. 2. Control system with the complete set of independent inputs and outputs

The issue of stability of the overall system is then solved as follows.

Lemma1. The overall system described by (6) and (7) is stable if and only if the matrix $PA + QB$ is unimodular.

Proof. In the overall system, inject inputs x and w as shown in Fig. 2. Then the overall system is stable if and only if the nine transfer matrices between the inputs v, w, x and the outputs u, y, z given by

$$\begin{bmatrix} u \\ z \\ y \end{bmatrix} = \begin{bmatrix} A \\ B \\ C \end{bmatrix} (PA + QB)^{-1} \begin{bmatrix} P & -Q & R \end{bmatrix} \begin{bmatrix} x \\ w \\ v \end{bmatrix}$$

are all well defined and proper and stable rational. This statement follows from the assumption of stability of the uncontrollable and unobservable parts of the system.

Now, in view of the coprimeness assumptions on A, B, C and P, Q, R these transfer matrices are well defined and stable if and only if $PA + QB$ is a unimodular matrix. □

4 PROBLEM SOLVABILITY

A simple necessary and sufficient condition will now be established for a system to be decoupled and stable.

Based on the partition (p_1, ..., p_k), write

$$C := \begin{bmatrix} C_1 \\ \vdots \\ C_k \end{bmatrix}, \tag{9}$$

where C_i is a $p_i \times q$ submatrix.

Theorem 1. Given system (1), (2) in fractional form (6) and partition (9), there exists an admissible controller (3) such that the overall system is

(i) stable if and only if

$$A \text{ and } B \text{ are right coprime,} \tag{10}$$

(ii) decoupled if and only if

$$\sum\nolimits_{i=1}^{k} \operatorname{rank} C_i = \operatorname{rank} C\,. \tag{11}$$

Proof. (i) Let the overall system be stable. By Lemma 1, the matrix $PA + QB$ is unimodular whence A and B must be right coprime.

Conversely, let the matrices A and B of (6) be right coprime. Then there exist proper and stable rational matrices P and Q such that

$$PA + QB = I \tag{12}$$

with P invertible and the inverse of P proper.

Then controller (3) in fractional form (7) that is defined by the matrices P and Q from (12) and by an arbitrary proper and stable rational matrix R satisfying rank CR = rank C is admissible since, by (8),

$$\operatorname{rank} T = \operatorname{rank} CR = \operatorname{rank} C = \operatorname{rank} S_y\,.$$

The resulting system (1), (2) and (3) is stable in view of Lemma 1 and identity (12).

(ii) Let (7) be an admissible decoupling controller for system (6). Denote

$$K := (PA + QB)^{-1} R\,.$$

The block diagonal property of the matrix T then implies

$$\operatorname{rank} CK = \sum\nolimits_{i=1}^{k} \operatorname{rank} C_i K$$

and the admissibility of the controller gives

$$\operatorname{rank} C_i K = \operatorname{rank} C_i\,,\ \ i = 1, \ldots, k.$$

Therefore (11) holds.

The sufficiency will be proved by constructing a suitable R. Denote

$$r_i := \operatorname{rank} C_i\,, \quad i = 1, \ldots, k\,.$$

Then there exists a $p_i \times p_i$ unimodular proper and stable rational matrix U_i such that

$$C_i = U_i \begin{bmatrix} C_i' \\ 0 \end{bmatrix}, \tag{13}$$

where the rows of C_i' are linearly independent over R(s) and where the zero matrix has $p_i - r_i$ rows and may be empty. If (11) holds, then

$$C' := \begin{bmatrix} C_1' \\ \vdots \\ C_k' \end{bmatrix}$$

has linearly independent rows over R(s). Hence there exists a $q \times q$ unimodular proper and stable rational matrix U' such that

$$C'U' := \begin{bmatrix} D_1 & & & 0 \\ & \ddots & & \vdots \\ & & D_k & 0 \end{bmatrix}, \tag{14}$$

where D_i is an $r_i \times r_i$ diagonal proper and stable rational matrix and where the zero matrices have $q - r$ columns with r defined by

$$r := \sum\nolimits_{i=1}^{k} r_i\,.$$

Define an admissible controller (7) by the matrices P and Q from (12) and by the matrix R formed by the first r columns of U'. The transfer matrix (8)

$$T = CR = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_k \end{bmatrix} \begin{bmatrix} \begin{bmatrix} D_1 \\ 0 \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} D_k \\ 0 \end{bmatrix} \end{bmatrix} \tag{15}$$

is block diagonal. The resulting system is therefore decoupled and the external reference input v has dimension r. □

The interpretation of these solvability conditions is as follows. Condition (10) corresponds to the stability of the subsystem of the given system that is not observable at the measured output z. Condition (11) calls for the linear independence of any two outputs of the given system that belong to different blocks. The solvability of the decoupling problem thus strongly depends on the partition (p_1, ..., p_k), that is to say, upon the allocation of the outputs into the blocks.

5 CONTROLLER PARAMETERIZATION

When a decoupling and stabilizing controller exists, we shall parameterize the class of all such controllers.

The control system (6), (7) is stable if and only if $PA + QB$ is a unimodular matrix by Lemma 1. Thus stabilization involves only the feedback part K_z of the controller, which surrounds the measurement subsystem S_z. As a result, the parameterization of K_z amounts to the well-known Youla-Kučera parameterization of feedback stabilizing controllers For details, see Kučera (1975), Youla, Jabr and Bongiorno (1976), Kučera (1979), Desoer *et al.* (1980), and Vidyasagar (1985).

Let $\overline{P}$, $\overline{Q}$ be any solution pair of equation (12). Then the solution class of (12) is given by

$$P = \overline{P} + W\overline{B},\ \ Q = \overline{Q} - W\overline{A}\,, \tag{16}$$

where $\overline{A}$ and $\overline{B}$ are left coprime, proper and stable rational matrices such that

$$\overline{A}^{-1}\overline{B} = BA^{-1} \tag{17}$$

and W is an arbitrary proper and stable rational matrix parameter.

The class of all stabilizing proper rational K_z is then obtained in the form

$$K_z = -P^{-1}Q = -(\overline{P} + W\overline{B})^{-1}(\overline{Q} - W\overline{A}), \tag{18}$$

where the parameter W is constrained so that the inverse of $\overline{P} + W\overline{B}$ exists and is proper rational.

Once the control system (6) and (7) is stabilized, it is decoupled if and only if $T = CR$ by (8). Thus decoupling involves only the feedforward part K_v of the controller.

Partition the $q \times q$ unimodular matrix U' defined in (14) as

$$U' = \begin{bmatrix} U'_r & U'_{q-r} \end{bmatrix},$$

where U'_r has r columns and U'_{q-r} has $q - r$ columns and may be empty. The class of all decoupling proper rational K_v is then given by $K_v = P^{-1}R$ with P determined in (16) and

$$R = U'_r \begin{bmatrix} V_1 & & \\ & \ddots & \\ & & V_k \end{bmatrix}, \tag{19}$$

where V_i is an arbitrary $r_i \times r_i$ proper and stable rational matrix parameter. The matrices V_1, ..., V_k in turn parameterize the class of achievable block-diagonal transfer matrices (8) as follows

$$T = \begin{bmatrix} U_1 & & \\ & \ddots & \\ & & U_k \end{bmatrix} \begin{bmatrix} \begin{bmatrix} D_1 \\ 0 \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} D_k \\ 0 \end{bmatrix} \end{bmatrix} \begin{bmatrix} V_1 & & \\ & \ddots & \\ & & V_k \end{bmatrix}. \tag{20}$$

The parameterization of decoupling stabilizing controllers reveals that decoupling and stabilization are two independent issues. That is why the controller described by (3) is called the two-degree-of-freedom controller. However, this is no longer true for one-degree-of-freedom controllers, e.g., for the error-actuated controllers described by $u = -P^{-1}Q(v - w)$ in place of (7).

6 OPTIMAL CONTROLLERS

The decoupling constraint can deteriorate system's performance. The bonus of having a parameterized solution set is that the lost performance can easily be optimized. Optimal decoupling controllers can be obtained by an appropriate choice of the parameters V_1, ..., V_k and W.

Suppose that the control objective is for each block of outputs y_i to track the corresponding block of reference inputs v_i. Thus we suppose that $p_i = r_i$ for i = 1, ..., k, i.e., there are as many reference inputs as controlled outputs in each block. The tracking error for each block is given by

$$e_i := v_i - y_i = H_i v_i \,.$$

In view of (20), H_i has the generic form

$$H_i = I - F_i V_i \,, \tag{21}$$

where $F_i := U_i D_i$ and V_i are proper and stable rational matrices with F_i fixed and V_i an arbitrary parameter to be specified.

The benefits of controller parameterization will now be demonstrated in the case of H_2 control design. It turns out that only the parameters V_1, ..., V_k are subject to selection whereas W is free and can be independently selected to accommodate additional design specifications.

Suppose that for each block, the reference-to-error transfer function H_i is to have least H_2 norm defined by

$$\|H_i\|_2 := \left(\text{trace} \tfrac{1}{2\pi} \int_{-\infty}^{\infty} H_i^*(j\omega) H_i(j\omega) d\omega \right)^{1/2},$$

where the asterisk denotes the conjugate transpose. Thus, $H_i^*(s) := H_i^T(-s)$ for any complex argument s.

To achieve this task, determine the inner-outer factorization of F_i,

$$F_i = F_I F_O \,,$$

where F_I is inner and F_O is outer. Since F_i is square and nonsingular, F_I satisfies $F_I^* F_I = I$ and F_O is free of zeros in Re$s > 0$.

Since F_I is inner, left multiplication by F_I^* preserves the H_2 norm,

$$\|H_i\|_2 = \|F_I^* H_i\|_2 = \|F_I^* - F_O V_i\|_2 \,.$$

Observe that $F_I^*(\infty) = I$. Separate the strictly proper part, F_{Isp}^* , of F_I^* as follows

$$F_I^* = I + F_{Isp}^*$$

and note that, by definition, F_{Isp}^* has poles only in Re$s > 0$. Then

$$\begin{aligned} \|H_i\|_2^2 &= \|F_{Isp}^* + (I - F_O V_i)\|_2^2 \\ &= \|F_{Isp}^*\|_2^2 + \| I - F_O V_i\|_2^2 \end{aligned}$$

because the cross terms contribute nothing to the norm. This is a complete square in which only the second term depends on V_i . Therefore, a unique V_i

that attains the minimum of the norm for subsystem i is

$$V_i = F_O^{-1}. \tag{22}$$

However, only a proper and stable V_i is admissible. It follows that the H_2 control problem for subsystem i has a solution if and only if F_O is unimodular. The minimum norm is then given by

$$\min_{V_i} \|H_i\|_2 = \|F_{Isp}\|_2 .$$

7 AN EXAMPLE

Consider a system defined by (1), (2) with the transfer matrices

$$S_y = \begin{bmatrix} 1 & \dfrac{s+2}{s-1} \\ \dfrac{s-1}{s+2} & 2 \end{bmatrix}, \quad S_z = \begin{bmatrix} \dfrac{2s+1}{s+2} & \dfrac{3s}{s-1} \\ \dfrac{s-1}{s+2} & 2 \end{bmatrix}.$$

Thus the measurement output z is different from the output y to be decoupled in that it involves a non-unity feedback sensor.

The task is to determine a two-degree-of-freedom controller (3) that (1, 1)-decouples and stabilizes the system.

The first step is to obtain a proper and stable fractional representation (6) for the system. Standard calculations yield

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \dfrac{s-1}{s+2} \end{bmatrix},$$

$$B = \begin{bmatrix} \dfrac{2s+1}{s+2} & \dfrac{3s}{s+2} \\ \dfrac{s-1}{s+2} & 2\dfrac{s-1}{s+2} \end{bmatrix}, \quad C = \begin{bmatrix} \dfrac{1}{\dfrac{s-1}{s+2}} & \dfrac{1}{2\dfrac{s-1}{s+2}} \end{bmatrix}.$$

Now apply Theorem 1. Since A is right coprime to B, a stabilizing controller exists. Since the rank of C equals the sum of the ranks of the rows of C, an admissible decoupling controller exists as well.

All stabilizing and decoupling controllers will be parameterized using the fractional representation (7). To obtain the feedback part of the controller, we consider any particular solution of equation (12), for example

$$\overline{P} = \begin{bmatrix} 1 & 0 \\ -\dfrac{2s+1}{s+2} & -2 \end{bmatrix}, \quad \overline{Q} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

A left coprime fractional representation that satisfies (17) is given by

$$\overline{A} = \begin{bmatrix} 0 & 1 \\ -\dfrac{s-1}{s+2} & 0 \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} \dfrac{s-1}{s+2} & 2 \\ -\dfrac{(s-1)(2s+1)}{(s+2)^2} & -\dfrac{3s}{s+2} \end{bmatrix}.$$

Thus the solution class (16) of equation (12) is

$$P = \begin{bmatrix} 1 & 0 \\ -\dfrac{2s+1}{s+2} & -2 \end{bmatrix} + W \begin{bmatrix} \dfrac{s-1}{s+2} & 2 \\ -\dfrac{(s-1)(2s+1)}{(s+2)^2} & -\dfrac{3s}{s+2} \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} - W \begin{bmatrix} 0 & 1 \\ -\dfrac{s-1}{s+2} & 0 \end{bmatrix}. \tag{23}$$

To obtain the feedforward part of the controller, note that $U_1 = U_2 = 1$ and the unimodular matrix defined in (14) equals

$$U' = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}.$$

Thus (19) yields

$$R = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}. \tag{24}$$

The matrices P, Q in (23) and R in (24) define the class of all controllers that solve the given problem. The parameters V_1, V_2 are free proper and stable rational functions and W is permitted to range over proper and stable rational 2 × 2 matrices so that the inverse of P exists and is proper. Obviously, this means that $P(\infty)$ is to be a nonsingular matrix.

The decoupled transfer matrices that can be achieved in this example are given by (20) as

$$T = \begin{bmatrix} 1 & 0 \\ 0 & \dfrac{s-1}{s+2} \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}.$$

The optimal controller that minimizes the H_2 norm of the reference-to-error transfer matrix is determined from (22), channel by channel. Clearly $V_1 = 1$. To optimize V_2, the inner-outer factorization of

$$F_2 = \frac{s-1}{s+2}$$

is seen to be

$$F_I = \frac{s-1}{s+1}, \quad F_O = \frac{s+1}{s+2}$$

and the strictly proper part of

$$F_I^* = \frac{s+1}{s-1}$$

equals

$$F_{Isp}^* = \frac{2}{s-1} .$$

Thus, from (22),

$$V_2 = \frac{s+2}{s+1}\ .$$

It follows from (24) that the unique optimal R is

$$R = \begin{bmatrix} 2 & -\dfrac{s+2}{s+1} \\ -1 & \dfrac{s+2}{s+1} \end{bmatrix}$$

and the overall system has the transfer function

$$T = \begin{bmatrix} 1 & 0 \\ 0 & \dfrac{s-1}{s+1} \end{bmatrix}.$$

8 CONCLUSION

An optimal H_2 decoupling control problem has been studied in the most general setting, for systems in which the measurement output may be different from the output to be decoupled and for dynamic controllers that feature both feedback and feedforward parts. The class of all such controllers that decouple and stabilize the system has been determined in parametric form and the parameter has been used to obtain the H_2-optimal controller.

The main contribution of the present paper is in a streamlined and transparent exposition and a simple and direct solution. This is primarily because of the following facts. The adopted controller configuration is ideally suited to decoupling since stability and noninteraction can be treated as two independent constraints. The problem is formulated and solved using an algebraic approach, namely the notion of proper and stable fractional representations for system's transfer matrices. The parameterization of the decoupling controllers is achieved via the Youla-Kučera parameterization of all stabilizing controllers. Finally, the H_2 norm involved in the optimization is minimized using the completion of the squares, which is a simple algebraic technique.

A large body of literature exists on decoupling and related topics. In technical details, the present paper draws inspiration from the work of Hautus and Heymann (1983) for the formulation of the problem, from Kučera (1983) for the algebraic treatment of stability, from Desoer and Gündeş (1986) for the parameterization of the decoupled system, and from Lee and Bongiorno (1993) for the optimal control.

ACKNOWLEDGMENTS

Supported by the Ministry of Education of the Czech Republic under project 1M0567 Centre for Applied Cybernetics.

REFERENCES

Basile, G. and G. Marro (1970). A state space approach to non interacting controls. *Ric. Autom.* **1**, 68-77.

Bongiorno, J.J. and D.C. Youla (2001). Wiener-Hopf design of optimal decoupling one-degree-of-freedom controllers for plants with rectangular transfer matrices. *Int. J. Control* **74**, 1393-1411.

Denham, M.J. (1973). A necessary and sufficient condition for decoupling by output feedback. *IEEE Trans. Automatic Control* **18**, 537.

Descusse, J. and M. Malabre (1982). Solvability of the decoupling problem for linear constant (*A*, *B*, *C*, *D*)-quadruples with regular output feedback. *IEEE Trans. Automatic Control* **27**, 456-458.

Descusse, J. and J.M. Dion (1982). On the structure at infinity of linear square decoupled systems. *IEEE Trans. Automatic Control* **27**, 971-974.

Descusse, J., J.F. Lafay, and V. Kučera (1984). Decoupling by restricted static state feedback: The general case. *IEEE Trans. Automatic Control* **29**, 79-81.

Desoer, C.A., R.W. Liu, J. Murray, and R. Saex (1980). Feedback system design: The fractional representation approach to analysis and synthesis. *IEEE Trans. Automatic Control* **25**, 399-412.

Desoer, C.A. and A.N. Gündeş (1986). Decoupling linear multiinput-multioutput plant by dynamic output feedback: An algebraic theory. *IEEE Trans. Automatic Control* **31**, 744-750.

Falb, P.L. and W.A. Wolovich (1967). Decoupling in the design and synthesis of multivariable control systems. *IEEE Trans. Automatic Control* **12**, 651-659.

Filev, D.P. (1982a). Some new results in state space decoupling of multivariable systems. I. A link between geometric and matrix methods. *Kybernetika* **18**, 215-233.

Filev, D.P. (1982b). Some new results in state space decoupling of multivariable systems. II. Extensions to decoupling of systems with $D \neq 0$ and output feedback decoupling. *Kybernetika* **18**, 330-344.

Gilbert, E.G. (1969). The decoupling of multivariable systems by state feedback. *SIAM J. Control* **7**, 50-63.

Gómez, G.I. and G.C. Goodwin (2000). An algebraic approach to decoupling in linear multivariable systems. *Int. J. Control* **73**, 582-599.

Hautus, M.L.J. and M. Heymann (1983). Linear feedback decoupling – Transfer function analysis. *IEEE Trans. Automatic Control* **28**, 823-832.

Hazlerigg, A.D.G. and P.K. Sinha (1978). A noninteracting control by output feedback and dynamic compensation. *IEEE Trans. Automatic Control* **23**, 76-79.

Howze, J.W. (1973). Necessary and sufficient conditions for decoupling using output feedback. *IEEE Trans. Automatic Control* **18**, 44-46.

Howze, J.W. and J.B. Pearson (1970). Decoupling and arbitrary pole placement in linear systems using output feedback. *IEEE Trans. Automatic Control* **15**, 660-663.

Kavanagh, R.J. (1957). Noninteracting controls in linear multivariable systems. *AIEE Trans. Applications and Industry* **76**, 95-100.

Koussiouris, T.G. (1979). A frequency domain approach to the block decoupling problem. *Int. J. Control* **29**, 991-1010.

Kučera, V. (1975). Stability of discrete linear feedback systems. In: *Preprints 6th IFAC Congress*. Vol.1, paper 44.1.

Kučera, V. (1979). *Discrete Linear Control: The Polynomial Equation Approach*. Wiley. Chichester.

Kučera, V. (1983). Block decoupling by dynamic compensation with internal properness and stability. *Probl. Control Info. Theory* **12**, 379-389.

Lee, H.P. and J.J. Bongiorno (1993). Wiener-Hopf design of optimal decoupling controllers for plants with non-square transfer matrices. *Int. J. Control* **58**, 1227-1246.

Lin, C.A. (1997). Necessary and sufficient conditions for existence of decoupling controllers. *IEEE Trans. Automatic Control* **42**, 1157-1161.

Mejerov, M.V. (1965). *Multivariable Control Systems* (in Russian). Nauka. Moscow.

Morgan, B.S. (1964). The synthesis of linear multivariable systems by state feedback. In: *Proc. Joint Automatic Control Conference*, pp. 468-472.

Morse, A.S. and W.M. Wonham (1970). Decoupling and pole placement by dynamic compensation. *SIAM J. Control* **8**, 317-337.

Morse, A.S. and W.M. Wonham (1971). Status of noninteracting control. *IEEE Trans. Automatic Control* **16**, 568-581.

Park, K.H. (2008a). H_2 design of one-degree-of-freedom decoupling controllers for square plants. *Int. J. Control* **81**, 1343-1351.

Park, K.H. (2008b). Existence conditions of decoupling controllers in the generalized plant model. In: *Proc. 47th IEEE Conf. Decision and Control*, pp. 5158-5163.

Silverman, L.M. and H.J. Payne (1971). Input-output structure of linear systems with application to the decoupling problem. *SIAM J. Control* **9**, 199-233.

Strejc, V. (1960). The general theory of autonomity and invariance of linear systems of control. *Acta Technica* **5**, 235-258.

Voznesenskij, I.N. (1936). A control system with many outputs (in Russian). *Automat. i Telemech*. **4**, 7-38.

Vidyasagar, M. (1985). *Control System Synthesis: A Factorization Approach*. MIT Press. Boston.

Wolovich, W.A. (1974*). Linear Multivariable Systems*. Springer. New York.

Wonham, W.M. (1974). *Linear Multivariable Control*. Springer. New York.

Wonham, W.M. and A.S. Morse (1970). Decoupling and pole assignment in linear multivariable systems: A geometric approach. *SIAM J. Control* **8**, 1-18.

Youla, D.C. and J.J. Bongiorno (2000). Wiener-Hopf design of optimal decoupling one-degree-of-freedom controllers. *Int. J. Control* **73**, 1657-1670.

Youla, D.C., H. Jabr, and J.J. Bongiorno (1976). Modern Wiener-Hopf design of optimal controllers – Part II: The multivariable case. *IEEE Trans. Automatic Control* **21**, 319-338.

Two-state bilinear predictive control for hot-water storage tank

Zbigniew Ogonowski *

* *Institute of Automatic Control, Silesian University of Technology,*
Akademicka 16, 44-100 Gliwice, Poland
(Tel: +48 32 2371084; e-mail: Zbigniew.Ogonowski@polsl.pl)

Abstract: The paper presents original predictive algorithm for use in two-state (or binary) input control of nonlinear systems which are described with state-constrained bilinear models. It is shown in the paper, that instead of non-linear continuous-time model, non-stationary linear discrete-time model can be used to predict the system response. On the other hand, state constrains can be attached to the criterion index to be minimized in the predictive control law. This inclusion assures the closed-loop stability of the control system and simplifies minimization problem. The proposed algorithm is particularly valuable for applications in heating systems where bilinearity follows from the heat exchange due to flow of liquid medium and constrains concern temperature regime. Application of the algorithm to control a hot water tank is presented in the paper. The tank is modeled with stratified model.

Keywords: bilinear models, predictive control, non-linear state observer, stratified models, hot-water tank.

1. INTRODUCTION

High hopes that were associated with non-linear predictive control to be a general control methodology proved to be futile at the turn of the century. Great ferment that the works of Michalska and Mayne (e.g. Michalska and Mayne (1993)) raised in the middle of nineties collapsed after confrontation with requirements of a real-world applications. The biggest problem posed optimization task which needs to be solved in every sampling period thus the applications have been restricted to slow processes Kwon and Han (2005). This problem was clearly stated at the end of nineties (Allgower and Zheng (2000), "Nonlinear Model Predictive Control: Challenges and Opportunities" by D. Mayne, pp. 23–44), and still remains unsolved. On the other hand special cases of nonlinearities has been studied meanwhile Rossiter (2004). One of the most deeply explored is the case of linear dynamics and input/state constrains Maciejowski (2002).

In the same spirit this paper explores bilinear systems with state constrains and two-state input signals as yet another special case of nonlinear system to be controlled. There are number of processes being modeled with bilinear models. The most important group of such systems form heat transfer processes where the energy is transported with liquid fluids ASHRAE (2009). The bilinear model of heat exchange arises due to states (temperatures) are multiplied by the control signal (liquid flow). Additionally, states are constrained by technological requirements and inputs are constrained because heat sources and pumps can be only switched on or off. Usually control systems apply simple relay controllers where so called cut-off method allows for compliance with constrains. Quality of the relay-based control can be hardly improved. There are only few knobs to be used as hysteresis or dynamical corrections Skoczowski (1981).

The paper presents original predictive algorithm for use in two-state input control of nonlinear systems which are described with state-constrained bilinear models.

Bilinear models are described in sec. 2. Based on these models the predictive control algorithm with two-state input and state constrains is derived in sec. 3. Discrete in time nature of predictive control needs discretization of the bilinear model. It is shown in sec. 3, that instead of non-linear continuous-time model, non-stationary linear discrete-time model is obtained. This technique is similar to so called successive linearization (e.g. in L. Magni and Allgower (2009), M. Cannon et.all. 'Successive Linearization NMPC for a Class of Stochastic Nonlinear Systems', pp. 249-262). However, proposed method uses exact model and does not impose linearization errors. State prediction on the assumed horizon is made on the basis of obtained model. It follows form the general theory of nonlinear predictive control Rossiter (2004) that closed-loop stability is assured by augmenting the criterion function to be minimized with final state weighting. This is done here by inclusion of the constrains into the criterion function as a penalty term.

Sec. 4 presents state observer for bilinear systems. This is the case of application example namely heating systems where bilinearity follows from the heat exchange due to flow of liquid medium and constrains concern temperature regime. Application of the algorithm to control a hot water tank is presented in sec. 5. The tank is modeled with so-called stratified model. Usually it is not possible to measure all states which follow from the stratification thus observer is necessary.

2. BILINEAR PROCESSES

Bilinear systems are the special nonlinear systems where linearity concerns separately state and control variables but not jointly. The general form of the bilinear model can be represented by the following:

$$\dot{X}(t) = AX(t) + B_0U(t) + \sum_{k=1}^{m} B_kX(t)u_k(t) + E(t) \quad (1)$$

where $X \in R^n$ and $U \in R^p$ (u_k represents k-th element of U). It is clear that $A \in R^{n\times n}$, $B_0 \in R^{n\times p}$ and $B_k \in R^{n\times n}$, $k = 1, 2, \dots, m$. Operating point is assumed zero (the model (1) represents deviations from the operating point). Term $E(t)$ represents disturbances. It is assumed, that after sampled, disturbances can be modeled by white noise. This assumption allows for optimal in mean-square sense prediction of the state by using the model (1) with disturbances term omitted (optimal prediction of white noise is equal to zero).

Modern system theory made possible and stimulated expansion and deepening of research so that the intrinsic limits of linear models appear more and more evident. There are number of disciplines where bilinear models found applications e.g. industrial processes, biology, economics, ecology agriculture etc. This type of nonlinear dynamical models have been rigorously explored in the last three decades. It has been shown Mohler (1991) that bilinear systems are better controllable in general then linear systems. They offer better possibility in control performance. Still interest in these systems is very high. The structural theory is fairly well established and in particular there are several satisfactory contributions on controllability, mainly for homogeneous in the state bilinear systems. Also, with respect to mathematical modeling problems, the available results are quite definite.

3. PREDICTIVE TWO-STATE CONTROL OF BILINEAR SYSTEM WITH CONSTRAINS

In two-state control it is assumed that elements u_k of the control vector variable U can achieve only 0 or 1 value. Inequality state constrains are also involved, and can be expressed in general form as

$$\Omega X(t) \le X_{con}. \quad (2)$$

Matrix Ω allows to easily limit on the maximum value of state variable (e.g. temperature cut-off) as well as the value of the acceptable range of states (e.g. output temperature of the heating system). Control predictive algorithm is formulated as discrete in time and zero-order holder of the control signal is assumed. The essence of the predictive control algorithm synthesis is solving of the optimization task in every sampling period. The objective function of the optimization task is formulated as a difference between predicted state trajectory and reference trajectory (usually equal to assumed set-points in the future) according to assumed prediction horizon. The optimizing criterion is the function of future controls, however, after optimization task is solved, only first element of the solution (nearest control) is applied and the whole procedure is repeated in the next sampling period (receding horizon technique).

Let the current moment in time is denoted by t_i, and sampling period T_s. Then: $t_{i+j} = t_i + j \cdot T_p$. Usually objective function is defined in the following quadratic form:

$$J(U(t_{i+j})|_{j=0,1,\dots,N-1}) = \sum_{j=1}^{N} e_x^T(t_{i+j})Qe_x(t_{i+j}) + \\ + U^T(t_{i+j-1})RU(t_{i+j-1}) \quad (3)$$

where

$$e_x(t_{i+j}) = X(t_{i+j} - X_{sp}) \quad (4)$$

is j-step prediction of the difference between states X and their set-points X_{sp}. Positive (semi)definite matrices Q and R as well as the prediction horizon N form the algorithm's parameters. Constrains (2) of the optimization task should be fulfilled in every sampling period t_{i+j}, $j = 1, 2, \dots, N$. Obviously, in general it is not possible to assure the existence of such control sequence $U(t_{i+j})|_{j=0,1,\dots,N-1}$, that the constrains are fulfilled because initial conditions can be out of the constrains. Thus it is much simpler to include the constrains into the criterion function and allows penalty method for searching the optimal solution. This also simplifies the searching algorithm because the optimization task is now constrains-free. Finally, the criterion function takes the form

$$J(U(t_{i+j})|_{j=0,1,\dots,N-1}) = \sum_{j=1}^{N} e_x^T(t_{i+j})Qe_x(t_{i+j}) + \\ + U^T(t_{i+j-1})RU(t_{i+j-1}) + \varphi(\Omega X(t_{i+j}) - X_{con}) \quad (5)$$

where φ is scalar penalizing function with the vector argument equal to exceeding the limits.

The above formulation of the predictive control algorithm allows for simple inclusion of requirements to keep the states within the proper range. There are two ways to do that:

- Determine set-points for the certain state and choose the proper weighting matrix Q in (5) depending on the role of the state in the system.
- Form the constrains (2) in such a way, that the range of certain state is properly narrowed.

The predicted states in objective function (5) should be determined from the model (1). Zero-order holder allows for the following representation of (1):

$$\dot{X}(t) = \left(A + \sum_{k=1}^{m} u_k(t_{i+j-1})B_k\right) X(t) + B_0U(t), \\ t \in [t_{i+j-1}, t_{i+j}]. \quad (6)$$

Equation (6) is linear and its solution at the end of the sampling period is as follows:

$$X(t_{i+j}) = \Phi_{i+j-1}X(t_{i+j-1}) + \Gamma_{i+j-1}U(t_{i+j-1}) \quad (7)$$

where

$$\Phi_{i+j-1} = e^{\left(A+\sum_{k=1}^{m} u_k(t_{i+j-1})B_k\right)T_p} \quad (8)$$

$$\Gamma_{i+j-1} = \int_0^{T_p} e^{\left(A+\sum_{k=1}^{m} u_k(t_{i+j-1})B_k\right)\tau} d\tau B_0. \quad (9)$$

Starting with initial state $X(t_i)$ the succeeding iterations are performed according to (7 − 9) for $j = 1, 2, \ldots, N$ to determine the whole trajectory of the state on the horizon N.

Number of possible control vector values on the horizon N is equal to $m \cdot 2^N$. If the sampling period is not to short then the prediction horizon need not to be large and the optimization task can be solved by bruteforce method. Similar approach was used in Ogonowski (2011b)

It should be emphasized that in every sampling period matrices Φ_{i+j-1} and Γ_{i+j-1} have to be determined. These matrices depend on input signal and change in every step. Thus the model (7) is nonstationary. Calculation of Φ_{i+j-1} and Γ_{i+j-1} needs application of special algorithms e.g. squaring and scaling Higham (2005). If sampling period is short and complex calculations are not possible then simplified model can be applied by using Euler method of integration:

$$X(t_{i+j}) = A'_{i+j-1}X(t_{i+j-1}) + B'_0U(t_{i+j-1}) \tag{10}$$

where

$$A'_{i+j-1} = \left(A + \sum_{k=1}^{m} u_k(t_{i+j-1})B_k\right)T_p \tag{11}$$

$$B'_0 = B_0T_p. \tag{12}$$

4. STATE OBSERVER FOR BILINEAR SYSTEM

To calculate state prediction it is necessary to start iteration of the model (7) or (10) with current measurement of the state $X(t_i)$. Often the only part of X is measured. Then the state observer is necessary. The theory of bilinear model state observer is well established (e.g. Hara and Furuta (1976)). Assume that s elements of X vector is measured. The state vector can be ordered to keep them on the top, to simplify the notation:

$$Y(t) = (I_s\ 0)\,X(t) \tag{13}$$

where I_s is s-dimensional unity matrix. Y represents then vector of measured states. Equation (1) can be factorized as follows

$$\begin{aligned}\dot{X}(t) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} X(t) + \begin{bmatrix} B_{0,1} \\ B_{0,2} \end{bmatrix} U(t) + \\ + \sum_{k=1}^{m} \begin{bmatrix} B_{k,11} & B_{k,12} \\ B_{k,21} & B_{k,22} \end{bmatrix} X(t)u_k(t)\end{aligned} \tag{14}$$

where $A_{11}, B_{k,11} \in R^{s\times s}$, $A_{12}, B_{k,12} \in R^{s\times(n-s)}$, $A_{21}, B_{k,21} \in R^{(n-s)\times s}$, $A_{22}, B_{k,22} \in R^{(n-s)\times(n-s)}$, $B_{0,1} \in R^{s\times m}$, $B_{0,2} \in R^{(n-s)\times m}$. In Hara and Furuta (1976) it was proven, that if the following two conditions are kept

$$\mathrm{Re}\left[\mathrm{eig}\left(A_{22} + HA_{12}\right)\right] < 0 \tag{15}$$

$$B_{k,22} + HB_{k,12} = 0 \quad k = 1, 2, \ldots, m \tag{16}$$

then there exists state observer of minimal order which is realized with the following dynamical system:

$$\dot{Z}(t) = \hat{A}Z(t) + \hat{B}_0Y(t) + \sum_{k=1}^{m} \hat{B}_kY(t)u_k(t) + \hat{G}U(t) \tag{17}$$

$$\hat{X}(t) = \hat{C}Z(t) + \hat{D}Y(t) \tag{18}$$

where

$$\hat{A} = A_{22} + HA_{12} \tag{19}$$

$$\hat{B}_0 = A_{21} + HA_{11} - (A_{22} + HA_{12})H \tag{20}$$

$$\hat{B}_k = B_{k,21} + HB_{k,11} \tag{21}$$

$$\hat{G} = B_{0,2} + HB_{0,1} \tag{22}$$

$$\hat{C} = \begin{bmatrix} 0 \\ I_{n-s} \end{bmatrix} \tag{23}$$

$$\hat{D} = \begin{bmatrix} I_s \\ -H \end{bmatrix}. \tag{24}$$

It was proven that the error $\hat{X}(t) - X(t)$ and all its derivatives tends to zero independently on U and initial conditions $X(t_0)$ and $Z(t_0)$.

5. APPLICATION

The proposed algorithm is particularly valuable for applications in heating systems where bilinearity follows from the heat exchange due to flow of liquid medium. Constrains concern temperature requirements. Application of the presented predictive algorithm will be now shown on the example of 300 liters hot water tank. The tank is equipped with one heating coil pipe placed in the upper part of the tank. The coil is fed with on-off controlled boiler (16200 W) throughout water as a heating medium. The tank has been equipped with measurement system Ogonowski (2010) containing termo-elements and hot water flow meter. The tank is modeled with stratified model.

5.1 Model of the hot water tank

Hot water tank is a vertically standing cylinder equipped with M heating pipe coils distributed in different parts along the vertical axis. Cold water water enters the tank bottom and is charged on top. Thus temperature gradient occurs. After division of the cylinder onto S layers the basic heat balance can be written as follows:

$$Q_{wn} = \sum_{m=1}^{M} Q_{m(n)} - Q_{un} - Q_{sn}, \tag{25}$$

where Q_{wn} is the heat accumulated in the n-th layer, $Q_{m(n)}$ is the heat transmitted by the m-th source to the n-th layer, Q_{un} is the heat applied from n-th layer and Q_{sn} is the heat loses of the n-th layer to the surroundings. Let consider single layer which is driven with a heat source transmitting Q_p through the heating medium of the flow F_z with enter temperature T_{zi} and exit temperature T_{zo}. Could water has got the temperature on the input equal to T_{wi} and on the output T_{wo}. The tank is surrounded by the environment of the temperature T_{sur}. Heat exchange describes the following two differential state equations:

$$\begin{aligned}\rho C_w V\frac{dT_{wo}}{dt} = \rho C_w F_z(T_{zi} - T_{zo}) - \rho C_w F_w(T_{wo} - T_{wi}) - \\ - \frac{\lambda A}{d}(T_{wo} - T_{sur}),\end{aligned} \tag{26}$$

$$\rho C_w V_w\frac{dT_{zi}}{dt} = Q_p - \rho C_w F_z(T_{zi} - T_{zo}) - \frac{\lambda_w A_w}{d_w}(T_{zi} - T_{sur}),$$

where ρ, C_w represents density and specific heat of the water respectively, V, V_w are the volumes of the layer and pipe coil, λ, λ_w are heat permeability coefficient of the tank wall and pipe coil respectively. In general, by division of the both sizes of (26) by $\rho C_w V$ and $\rho C_w V_w$, and taking into account heat exchange between layers, one can derive the following:

$$\begin{aligned}\frac{dT^n_{wo}}{dt} &= b^n_1 F^{n,m}_z (T^n_{zi} - T^n_{wo}) - b^n_2 F_w (T^n_{wo} - T^{n-1}_{wo}) - \\ & b^n_3 (T^n_{wo} - T_{sur}) - b^n_4 (T^n_{wo} - T^{n-1}_{wo}) + b^n_5 (T^{n+1}_{wo} - T^n_{wo}), \quad (27)\\ \frac{dT^m_{zi}}{dt} &= p^n_1 Q^m_g - p^n_2 F^m_z (T^m_{zi} - T^n_{wo}) - p^n_3 (T^m_{zi} - T_{sur}),\end{aligned}$$

where superscript n denotes number of layer, $n+1$ is the number of upper layer and $n-1$ is the number of lower layer. m is the number of heat source which is directly coupled with the n-th layer. In equation (27) physical coefficients has been exchanged with constants b and p. First evaluation of b and p can follow from physical meaning. The final ones, however, have to be identified because stratified model is simplification of the real plant which has got a distributed parameter nature. Additionally, dependent variable T_{zo} has been excluded from the above equations which is possible under assumption that heat transfer driving force is the average temperature T_{zi} and T_{zo} Marlin (1995).

5.2 Parameter identification

Simple method for identification of (27) model bases on distinguishing the periods of time where some parts of the model remain zero. This follows from specific of the model (27) e.g. if heat source or pump does not work then respective signals F_z or F_w are zero. The model (27) can be identified part by part with properly chosen data.

It is assumed three-layered structure of the model thus the model takes form:

$$\begin{aligned}\frac{dT^3_{wo}}{dt} &= b^3_1 F_z (T_{zi} - T^3_{wo}) - b^3_2 F_w (T^3_{wo} - T^2_{wo}) - \\ & -b^3_3 (T^3_{wo} - T_{sur}) - b^3_4 (T^3_{wo} - T^2_{wo}), \quad (28)\\ \frac{dT^2_{wo}}{dt} &= b^2_1 F_z (T_{zi} - T^2_{wo}) - b^2_2 F_w (T^2_{wo} - T^1_{wo}) - \\ & -b^2_3 (T^2_{wo} - T_{sur}) - b^2_4 (T^2_{wo} - T^1_{wo}) + b^2_5 (T^3_{wo} - T^2_{wo}), \quad (29)\\ \frac{dT^1_{wo}}{dt} &= -b^1_2 F_w (T^1_{wo} - T_{wi}) - b^1_3 (T^1_{wo} - T_{sur}) + \\ & +b^1_5 (T^2_{wo} - T^1_{wo}), \quad (30)\\ \frac{dT_{zi}}{dt} &= p_1 Q_g - p_2 F_z (T_{zi} - T^3_{wo}) - p_3 (T_{zi} - T_{sur}). \quad (31)\end{aligned}$$

Formulation (28)-(31) can be transformed to (1) with the following:

$$X = \begin{bmatrix} T^1_{wo} \\ T^2_{wo} \\ T^3_{wo} \\ T_{zi} \end{bmatrix}, \quad U = \begin{bmatrix} F_z \\ Q_g \end{bmatrix},$$

$$A = \begin{bmatrix} -(b^1_3 + b^1_5) & b^1_5 & 0 & 0 \\ b^2_4 & -(b^2_3 + b^2_4 + b^2_5) & b^2_5 & 0 \\ 0 & b^3_4 & -(b^3_3 + b^3_4) & 0 \\ 0 & 0 & 0 & -p_3 \end{bmatrix},$$

$$\qquad (32)$$

$$B_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & p_1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -b^2_1 & 0 & b^2_1 \\ 0 & 0 & -b^3_1 & b^3_1 \\ 0 & 0 & p_2 & -p_2 \end{bmatrix}, \quad B_2 = 0_{4\times 4}$$

$$E(t) = \begin{bmatrix} b^1_3 T_{sur} + b^1_2 F_w (T_{wi} - T^1_{wo}) \\ b^2_3 T_{sur} + b^2_2 F_w (T^1_{wo} - T^2_{wo}) \\ b^3_3 T_{sur} + b^3_2 F_w (T^2_{wo} - T^3_{wo}) \\ p_3 T_{sur} \end{bmatrix}.$$

Note, that disturbance vector $E(t)$ is added to the right hand side of (1). E contains two components. The first depends on T_{sur} and changes sufficiently slow to be accepted as constant. The second depends on F_w and changes much faster (see Figure 1). Analysis of F_w shows its white character. This justifies assumption, that disturbances term can be omitted in prediction.

It can be easily verified that if the heating is off ($Q_p, F_z = 0$) and no hot water is use ($F_w = 0$) then all state are equal to T_{sur}. Assuming $T_{sur} = const$ one can use deviation model where X means deviations from T_{sur}.

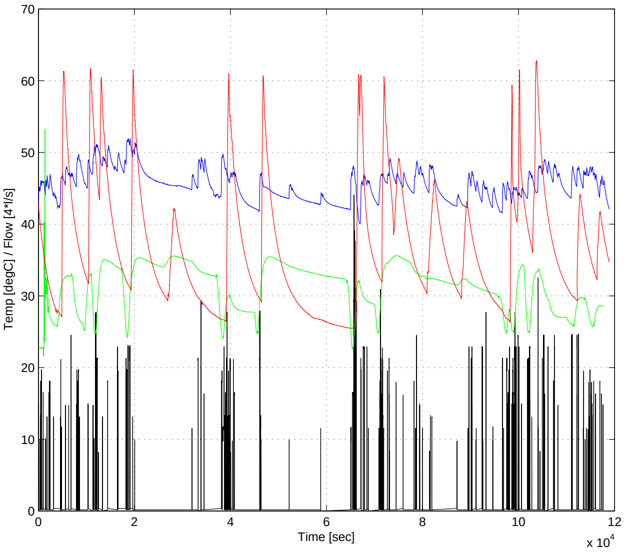


Fig. 1. Example of data for identification. T^3_{wo} - blue, T^2_{wo} - green, T_{zi} - red, F_w - black.

Figure 1 presents example of data. There are three temperatures measured: hot water (output of the tank) T^3_{wo} – blue line, temperature at half height of the tank T^2_{wo} (green) and temperature of the heating medium T_{zi} (red). Lower part of the tank keeps constant temperature $T^1_{wo} = 10^o$C which need not to be measured. $T_{sur} = 25.5^o$C was assumed constant as well. On-off control signal were boiler power $Q_g = 0$ or 16200 and heating medium flow $F_z = 0$ or 0.5. It is interesting to notice the behavior of T^3_{wo} temperature: if the heating is off (T_{zi} decreases) and hot water use appears (black line) then T^3_{wo} increases for some time while T^2_{w0} decreases. This phenomenon follows form placement of the measurement element – close, but outside of the tank. If the pump is off the pipe gets colder despite of the high inner temperature. After disturbance occurs pipe gets warmer despite temperature of cold water at the input of the tank is much lower then the inner temperature.

After carefully chosen periods it became possible to determine parameters of three-layer model Ogonowski (2011a) as presented in Table 1.

Table 1. Parameters of three-layer model.

Layer/ Parameter	$n = 1$	$n = 2$	$n = 3$
b_1	0	0.019	0.025
b_2	0.73	0.071	0.067
b_3	0.00005	0.00093	0.0058
b_4	0	0.00076	0.0049
b_5	0.00001	0	0
p_1	0.13		
p_2	0.015		
p_3	0.005		

5.3 Standard rely control

In practice, standard control system of hot water tank uses two relays. The first (with hysteresis) stabilizes T^2_{wo} on the prespecified set-point $T^2_{wo,sp}$. The second realizes so called cut-off algorithm: if T_{zi} exceeds $T_{zi,cut}$ then the boiler is switched off, however pump is still on until the first rely is on. This very simple algorithm is robust and ensures the maintenance of hot water volume on some level due to middle temperature is stabilized instead of the output one. The only drawback seems indirect stabilization of the output temperature. Thus, $T^2_{wo,sp}$ has to be properly chosen (usually by trial and error method).

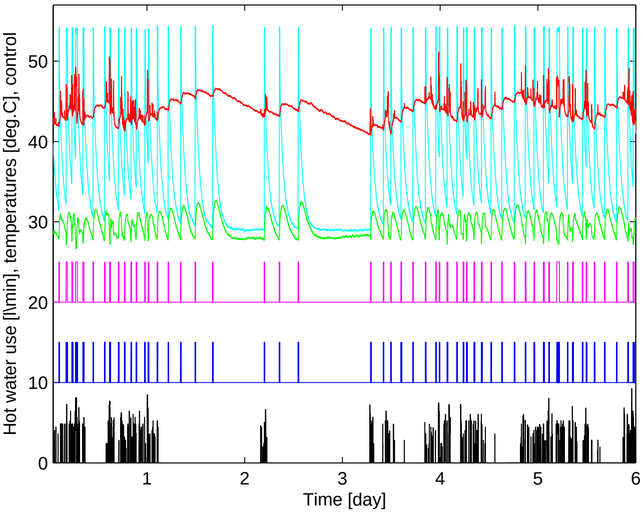


Fig. 2. Results of the standard on-off control performance. T^3_{wo} - red, T^2_{wo} - green, T_{zi} - cyjan, F_w - black, Q_p - blue (scaled to 10-15), F_z - magenta (scaled to 20-25).

Figure 2 demonstrates the results of the standard control under real-world operations. This means not only real-world experimentation in the environment sense, but also that the control system was tested during normal using of the tank. Disturbances (hot water use) caused a decrease in output temperature and control system reaction. Set point for middle temperature is $T^2_{wo,sp} = 28^o$C and is kept properly (green). Mean value of the output temperature T^3_{wo} is equal to 46^{o}C (red). Variance if T^3_{wo} is relatively large, however, one should remember that T^3_{wo} is controlled indirectly.

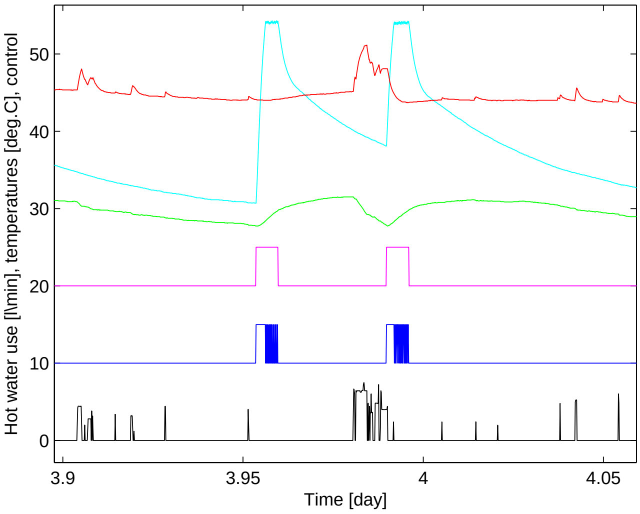


Fig. 3. Detail of the figure 2.

Detail of Figure 2 is presented in Figure 3 and explains on-off algorithm performance. After T^2_{wo} reached 28^{o}C (green), pomp F_z (magenta) and boiler Q_p (blue) gets on (time 3.98). T_{zi} (cyjan) increases fast and after reached cut-off temperature $T_{zi,cut} = 54^o$C signal Q_p starts switching. In this time F_z remains on because $T^2_{wo} < T^2_{wo,sp} = 28^o$C. After T^2_{wo} reached 30^{0}C (28+2^{o}C of hysteresis) both control signals are off. Then, due to succeeding use of hot water (black disturbances) T^2_{wo} decreases and the next reaction of the controller takes place (time about 4.03). Note the phenomenon of temporary increase of T^3_{wo} after hot water use.

5.4 Predictive control

Observer. Standard relay control does not need T^3_{wo} measurement (only T^2_{wo} and T_{zi} are necessary). Thus the tanks are not equipped with inner (i.e. placed in the probe) measuring thermoelement. Even if the outer measurement is possible (i.e. using clip-on temperature sensor) the phenomenon described above disturbs the result significantly thus the measurement can be hardly used for control. To conclude, observer of T^3_{wo} is necessary. In fact, there is also T^1_{wo} to be observed because it is not measured. However, bottom part of the tank has constant temperature, or it changes in significantly small range, thus the result of the observation has little influence on the control system.

According to (13) the states are rearranged to the following form:

$$X = \begin{bmatrix} T^2_{wo} \\ T_{zi} \\ T^3_{wo} \\ T^1_{wo} \end{bmatrix} \qquad (33)$$

and vector of measurement Y can be written as:

$$Y(t) = (I_2\ 0)X(t) \tag{34}$$

where I_2 is unity matrix of 2×2 size. The task of the observer is to determine

$$\hat{X} = \begin{bmatrix} T_{wo}^2 \\ T_{zi} \\ \hat{T}_{wo}^3 \\ \hat{T}_{wo}^1 \end{bmatrix} \tag{35}$$

which elements $\hat{T}_{wo}^3$ and $\hat{T}_{wo}^1$ tends sufficiently fast to T_{wo}^3 and T_{wo}^1 independently on initial conditions and disturbances. The matrices of the model (14) are as follows

$$A_{11} = \begin{bmatrix} -(b_3^2+b_4^2+b_5^2) & 0 \\ 0 & -p_3 \end{bmatrix}, \quad A_{12} = \begin{bmatrix} b_5^2 & b_4^2 \\ 0 & 0 \end{bmatrix}$$
$$A_{21} = \begin{bmatrix} b_4^3 & 0 \\ b_5^1 & 0 \end{bmatrix}, \quad A_{22} = \begin{bmatrix} -(b_3^3+b_4^3) & 0 \\ 0 & -(b_3^1+b_5^1) \end{bmatrix} \tag{36}$$

$$B_{0,1} = \begin{bmatrix} 0 & 0 \\ 0 & p_1 \end{bmatrix}, \quad B_{0,2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{37}$$

$$B_{1,11} = \begin{bmatrix} -b_1^2 & b_1^2 \\ 0 & -p_2 \end{bmatrix}, \quad B_{1,12} = \begin{bmatrix} 0 & 0 \\ p_2 & 0 \end{bmatrix}$$
$$B_{1,21} = \begin{bmatrix} 0 & b_1^3 \\ 0 & 0 \end{bmatrix}, \quad B_{1,22} = \begin{bmatrix} -b_1^3 & 0 \\ 0 & 0 \end{bmatrix} \tag{38}$$

and $B_{2,11} = B_{2,12} = B_{2,21} = B_{2,22} = 0_2$, where 0_2 is zero matrix of 2×2 size.

It can be easily checked that the condition (16) takes the form:

$$H = \begin{bmatrix} h_{11} & \dfrac{b_1^3}{p_2} \\ h_{21} & 0 \end{bmatrix} \tag{39}$$

Characteristic equation of a matrix

$$\hat{A} = A_{22} + HA_{12} = \begin{bmatrix} h_{11}b_5^2 - b_3^3 - b_4^3 & h_{11}b_4^2 \\ h_{21}b_5^2 & h_{21}b_4^2 - b_3^1 + b_5^1 \end{bmatrix} \tag{40}$$

has got the following form

$$\lambda^2 + \alpha\lambda + \beta = 0 \tag{41}$$

where:

$$\alpha = -h_{11}b_5^2 + (b_3^3+b_4^3) - h_{21}b_4^2 + (b_3^1+b_4^3), \tag{42}$$

$$\beta = h_{11}b_5^2(b_3^1+b_5^1) + h_{21}b_4^2(b_3^3+b_4^3) + (b_3^1+b_5^1)(b_3^3+b_4^3) \tag{43}$$

It follows from the Hurwitz criterion that the condition (15) is fulfilled if

$$\alpha > 0 \quad \text{i} \quad \beta > 0 \tag{44}$$

This can be transformed into two cases:

$$\text{If} \quad b_5^2 = 0 \quad \begin{cases} h_{11} & -\text{arbitral} \\ h_{21} < \dfrac{b_3^1+b_5^1}{b_4^1} \end{cases} \tag{45}$$

$$\text{If} \quad b_5^2 \neq 0 \quad \begin{cases} h_{11} < -\dfrac{b_4^2}{b_5^2}h_{21} + \dfrac{b_3^3+b_4^3+b_3^1+b_5^1}{b_5^2} \\ h_{11} < -\dfrac{b_4^2(b_3^3+b_4^3)}{b_5^2(b_3^1+b_5^1)}h_{21} + \dfrac{b_3^3+b_4^3}{b_5^2} \end{cases} \tag{46}$$

Derivation of (45) and (46) used fact, that $b_i^k \geq 0$. Using parameters of the model given in Table 1 one obtains: $h_{21} < 0.0789$ and h_{11} to be freely chosen. The choice influences convergence of the observer. Speed of the convergence follows from eigenvalues of $\hat{A}$. In the case discussed ($b_5^2 = 0$ – see Table 1), the eigenvalues are equal to:

$$\lambda_1 = -b_3^3 - b_4^3$$
$$\lambda_2 = h_{21} - b_3^1 - b_5^1 \tag{47}$$

It is clear form (47) that λ_1 does not depend on H, thus the speed of convergence can be shaped to a small extent by changing only λ_2. Finally the following values has been chosen:

$$H = \begin{bmatrix} 0 & 1.667 \\ -10 & 0 \end{bmatrix}. \tag{48}$$

Example of application. Set-points have been determined for the states as follows:

$$X_{sp} = \begin{bmatrix} T_{wo,sp}^2 \\ T_{zi,sp} \\ T_{wo,sp}^3 \\ T_{wo,sp}^1 \end{bmatrix} = \begin{bmatrix} 28 \\ 54 \\ 46 \\ 10 \end{bmatrix}. \tag{49}$$

Sampling time has been chosen $T_s = 1$ min. The model (1) has been discretized with simple Euler method (10). Constrains can be summarized as

$$\Omega\hat{X} = [0\ \ 1\ \ 0\ \ 0] \begin{bmatrix} T_{wo}^2 \\ T_{zi} \\ \hat{T}_{wo}^3 \\ \hat{T}_{wo}^1 \end{bmatrix} \leq 54^o\text{C} \tag{50}$$

One can notice that the control system influences of T_{zi} in two ways (set-point and constrains). This problem does not disappear after inclusion of the constrains into the criterion function (5). However, proper choice of the weighting matrix Q transfres the responsibility of T_{wzi} control on the penalizing function:

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & q_2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad q_1 \geq 0, \quad q_2 \geq 0. \tag{51}$$

Note, that the second row and column is zero. Control weighting matrix is assumed to be diagonal as well

$$R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix}, \quad r_1 \geq 0, \quad r_2 \geq 0. \tag{52}$$

Penalizing function is assumed to the Heaviside one

$$\varphi\left(\Omega\hat{X}(\cdot) - X_{con}\right) = \alpha 1(T_{zi}(\cdot) - 54^o\text{C}), \quad \alpha > 0. \tag{53}$$

The above formulated algorithm has bee tuned by trial and error method using simulations which has been conducted with the disturbances that had been measured during relay control experiment (see Figure 2). The results of tuning are as follows: $N = 4$, $\alpha = 150$, $q_1 = q_2 = 1.36$, $r_1 = r_2 = 0.12$. Figure 4 presents the results of the predictive control algorithm performance.

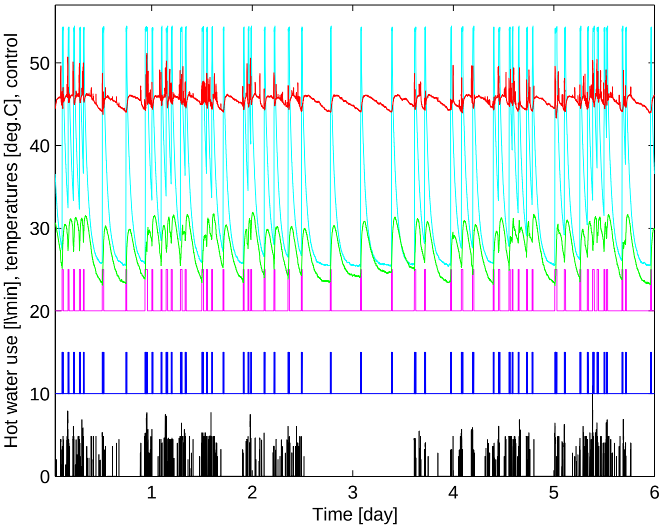


Fig. 4. Results of the predictive control algorithm performance. T^3_{wo} - red, T^2_{wo} - green, T_{zi} - cyjan, F_w - black, Q_p - blue (scaled to 10-15), F_z - magenta (scaled to 20-25).

Obviously, there is no possibility to use the same disturbance signal as in relay case, because experiment in real-life environment can not be repeated. However, it can be noted significantly better stabilization of the output T^3_{wo} temperature of the hot water (red line). On the other hand, stabilization of the middle T^2_{wo} temperature is worse (green) but yet this state is not important from the user needs view point. This can be seen e.g. between 2.5 and 3.5 [day]. Even in the absence of hot water outlets, reaction of the control algorithm takes place. This is due to the existence of feedback from observed T^3_{wo} which decreases because the tank cools down.

One would expect increase of the fuel consumption due to more frequent reaction of the control system when compare with the standard relay controller. This is not true. After much longer tests it became clear that predictive control is significantly economical. Long term observations proved about 9.5% fuel save when compare with relay control. The reason probably follows from the fact that predictive control takes into account energy price while minimizing objective function due to the term $U^T RU$. Standard controller does not take into account energy consumption at all.

6. CONCLUSION

Predictive control algorithm with state constrains allows for much better control performance then standard relay controllers. However, it is paid for with difficulty of tuning. There are number of parameters that should be properly chosen. What is more, quality of control depends on the quality of the model, because the prediction depends directly on model accuracy and indirectly on precision of observer which in turns depends on the model. Two further directions of research seems necessary to be undertaken. The first is multilayering of the control system structure which allows for application of upper-layer optimization of the controller parameters and operating point. The optimization can directly take into account fuel consumption. The second direction is adaptation of the model (or model self-tuning). It is possible due to structure of the model is known (stratification). Additional problem could be robustness of the control system on the model inaccuracy.

The paper presents application of the proposed algorithm to three layered model. The volume of the tank being tested allows for such stratification. Large industrial tanks need more precise stratified model which are build with greater number of layers. The theory, however remains the same and can be directly used.

ACKNOWLEDGMENTS

The work has been supported by the grant of Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland. This support is very gratefully acknowledged.

REFERENCES

Allgower, F. and Zheng, A. (2000). *Nonlinear Model Predictive Control*. Birkhauser Verlag.

ASHRAE (2009). *ASHRAE Handbook - Fundamentals*. ASHRAE, Atlanta, GA 30329.

Hara, S. and Furuta, K. (1976). Minimal order state observer for bilinar systems. *International Journal of Control*, 24, 705–718.

Higham, N. (2005). The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26, 11791193.

Kwon, W. and Han, S. (2005). *Receiding Horizon Control*. Spronger.

L. Magni, D.R. and Allgower, F. (2009). *Nonlinear Model Predictive Control. Towards New Challengigng Applications*. Springer.

Maciejowski, J.M. (2002). *Predictive Control*. Prentice Hall, Harlow.

Marlin, T. (1995). *Process Control*. McGraw-Hill.

Michalska, H. and Mayne, D. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38, 1623–1633.

Mohler, R. (1991). *Nonlinear systems – Application to bilinear control*. Prentice Hall.

Ogonowski, S. (2010). Modeling of the heating system in small building for control. *Energy and Buildings*, 42, 15101516.

Ogonowski, S. (2011a). Identification of nonlinear stratified model of the hot water tank. *Applied Energy*, (to appear).

Ogonowski, Z. (2011b). Drying control system for spray booth with optimization of fuel consumption. *Applied Energy*, 88, 1586–1595.

Rossiter, J. (2004). *Model-Based Predictive Control. A practical approach*. CRC Press LLC.

Skoczowski, S. (1981). *Dwunastawna regulacja temperatury (in Polish)*. PWN, Warszawa.

NMPC for stiff, distributed parameter system: Semi-Automatic Code Generation and optimality condition evaluation

R. Noga [*,**] **T. Ohtsuka** [***]

[*] *European Organization for Nuclear Research, CERN CH-1211, Genève 23, Switzerland*
[**] *University of Valladolid, c/ Real de Burgos s/n., 47011 Valladolid, Spain*
[***] *Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan (e-mail: mail@rafal-noga.com).*

Abstract: AutoGenU is a Mathematica program to automatically generate simulation programs for Nonlinear Model Predictive Control (NMPC). It analytically evaluates the Jacobians necessary to calculate the optimality condition in the NMPC realized using Continuation/Generalized Minimum Residual (C/GMRES) method. However, in the case of the LHC Superfluid Helium Cryogenic System, which is distributed parameter system, these Jacobians, expressed directly in terms of inputs, states and co-states become complex expressions due to cascading relations between internal variables of the circuit's model. A semi-automatic code generation procedure based on AutoGenU is presented, where intermediate variables are introduced and the chain rule is applied to evaluate the Jacobians, thereby avoiding complex expressions. In addition, the ODE set describing the system state dynamics is stiff, thus the dynamics time integration step is small. The intermediate variables are available at each step and are used to evaluate the optimality condition more precisely at low additional computing cost. The observed computational cost of the semi-automatically generated code is slightly lower than that of automatically generated and the controller performance is similar in both cases. However, the generation of semi-automatic code requires significantly less memory, and is much faster, widening the applicability of code generation for complex systems.

Keywords: Nonlinear Model Predictive Control, Nonlinear Receding Horizon Control, Automatic Code Generation System, Distributed Parameters System, Stiff system

1. INTRODUCTION

AutoGenU is a Mathematica program to automatically generate simulation programs for Nonlinear Model Predictive Control (NMPC) also known as Receding Horizon Control (RHC). Once the state equation, the performance index and some other simulation conditions are specified by a user as an input file in Mathematica® Format, then AutoGenU.nb loads the input file, executes such necessary operations as partial differentiation, and generates a C source file. The generated source file is ready for compilation and execution. The simulation program employs a fast optimization algorithm, Continuation/Generalized Minimum Residual (C/GMRES) (Ohtsuka (2000)). AutoGenU has been applied to generate the NMPC for the Superfluid Helium Cryogenic Circuit (SHCC) at the Large Hadron Collider (LHC) [1]. More precisely, a simulation independent implementation of the NMPC based on the C/GMRES optimization (Ohtsuka (2004)) has been separated from the simulation program. Then, this C code has been used in Matlab® simulations, accessed via MEX functions, (Noga et al. (2010)) and also it has been integrated into the PVSS II® SCADA of the LHC cryogenic system as a prototype implementation of NMPC for the SHCC.

A 106.9 m long Standard Cell of the SHCC is composed of eight, main superconducting magnets of the LHC, submerged in a bath of superfluid helium (Brüning et al. (2004)). The magnets are cooled via an over 100 m long Bayonet Heat Exchanger integrated into the magnets (Lebrun et al. (1997)). Our system corresponds to a Sub-Sector of the SHCC that is composed of two Standard Cells that share common helium bath (Gubello et al. (2006)). The system state $x(l,t)$ is the magnet temperatures as a function of time t and is spatially distributed over the Sub-Sector length $0 \leq l \leq 2 \times 106.9$ m. Its dynamics has been modeled as a function of a distributed value of cooling power of two heat exchangers that is a function of helium saturation temperature $a(l,t)$ and mass flow rate $b(l,t)$ in each heat exchanger (Noga (2007), Noga et al. (2010)). After spatial discretization of x, a and b using a Finite Volume approach with $N = 10$ intervals, the dynamics of discretized state

[1] The LHC is the newest particle accelerator and collider at the European Organization for Nuclear Research (CERN)

$$
\begin{aligned}
\mathrm{d}x_1/\mathrm{d}t &= f_1(x_1, x_2, a_1, b_1)\\
\mathrm{d}x_i/\mathrm{d}t &= f_i(x_{i-1}, x_i, x_{i+1}, a_i, b_i) \quad (i = 2, ..., N-1)\\
\mathrm{d}x_N/\mathrm{d}t &= f_N(x_{N-1}, x_N, a_N, b_N),
\end{aligned} \tag{1}
$$

with saturation temperature

$$
\begin{array}{l}
a_1 = a_{I,1}\\
a_{i+1}(a_i, b_i) \quad (i = 1, ..., N/2-1)\\
a_{O,1}(a_{N/2}, b_{N/2})\\
a_{N/2+1} = a_{I,2}\\
a_{i+1}(a_i, b_i) \quad (i = N/2+1, ..., N-1)\\
a_{O,2}(a_N, b_N)
\end{array} \tag{2}
$$

and He II mass flow rate

$$
\begin{array}{l}
b_1 = b_{I,1}\\
b_{i+1}(a_i, b_i, x_i) \quad (i = 1, ..., N/2-1)\\
b_{O,1}(a_{N/2}, b_{N/2}, x_{N/2})\\
b_{N/2+1} = b_{I,2}\\
b_{i+1}(a_i, b_i, x_i) \quad (i = N/2+1, ..., N-1)\\
b_{O,2}(a_N, b_N, x_N).
\end{array} \tag{3}
$$

Please note the presence of a Two-Point Boundary Value problem, since the boundaries are the He II mass flow rates at heat exchanger inlets that are the model manipulated variables

$$b_{I,k} = u_k \ (k = 1, 2) \tag{4}$$

and the saturation temperature at the outlets $a_{O,k}$. The saturation temperatures are equal in a Sub-Sector $a_{O,k} = a_{SS}$. Due to the spatial discretization scheme chosen, the saturation temperature at the inlets $a_{I,k}$ is solved to satisfy the boundary conditions at the outlets $a_{O,k}$ using the Newton method with n-th iteration

$$(a_{I,k})_{n+1} = (a_{I,k})_n - (\mathrm{d}a_{O,k}/\mathrm{d}a_{I,k})_n^{-1} \, [(a_{O,k})_n - a_{SS}]. \tag{5}$$

In order to evaluate the necessary optimality condition in the NMPC (Ohtsuka (2004)), AutoGenU analytically calculates the Jacobians H_u and H_x of the Hamiltonian

$$H = L(x, u) + \lambda^T \ f(x, u, t) + \mu^T \ C(x, u, t) \tag{6}$$

with respect to vectors of system inputs $u(t)$ and states $x(t)$. Here, t is time, L is a performance index that appears in the cost functional to be minimized during the optimization,

$$\dot{x} = f(x, u), \ C(x, u) = 0 \tag{7}$$

represent the system state dynamics and constraints respectively and $\lambda(t)$ and $\mu(t)$ are the Lagrange multipliers. In case of the SHCC , the Jacobians H_u and H_x expressed directly in terms of inputs and states become very complex and the automatic code generation fails due to excessive operational memory needed. However, the automatic generation of each component of the Jacobians separately has been successful. One method to significantly reduce the complexity of the H_u and H_x expressions is to introduce intermediate variables and use the chain rule to evaluate the Jacobians. A number of intermediate variables corresponding to internal model variables has been chosen to exploit the model structure. Since the choice of the intermediate variables and the implementation of the chain rule are done by hand, the resulting code generation procedure is semi-automatic.

The set of ODEs describing the SHCC state dynamics is stiff. In the C/GMRES version for systems with stiff dynamics, the residuum of the optimality condition is calculated using integrals of the Jacobian H_u over the prediction horizon grid (Noga et al. (2010)). The grid corresponds to intervals with constant control signal. The automatically generated H_u is expressed directly in terms of system inputs and states and thus has high computational cost, thus the integrals are calculated using simple quadratures such as one-point rectangular or two-points trapezoidal, where H_u is evaluated exclusively at the horizon grid. However, once the intermediate variables are introduced and then available at each state integration step, which in case of stiff system is much shorter than the grid interval, the H_u may be evaluated at each step at low additional computational cost, resulting in a more precise calculation of the optimality condition.

This paper presents the semi-automatic procedure of NMPC code generation for the SHCC based on AutoGenU. This section introduced AutoGenU and the motivation for the use of intermediate variables in case of the SHCC. Next, the choice of intermediate variables and the implementation of the chain rule are described in detail. Then the evaluation of the optimality condition using the intermediate variables is presented. Finally the performance of the semi-automatically generated NMPC code is compared against that generated automatically. The low memory required by the semi-automatic generation process is highlighted.

2. SEMI-AUTOMATIC CODE GENERATION

During automatic code generation using symbolic mathematics, the cascading relations between i-th and $(i-1)$-th variable a and b in Eqs. (2) and (3) enable propagation of complex expressions. This is especially visible in case of the Hamiltonian, Eq. (6), that involves the system dynamics f, Eq. (1), and, trough the index L, helium mass flow rates at the heat exchanger outlets calculated from Eq. (3). Its Jacobians H_u and H_x expressed directly in terms of inputs and states become very complex and the automatic code generation fails due to excessive operational memory needed. However, automatic generation of C code for each component of the Jacobians separately requires less memory and has been successful.

A method to avoid the propagation that significantly reduces the complexity of the expressions for H_u and H_x is to use a_i and b_i as intermediate variables. The components of H_u and H_x, which are the Jacobians f_x, f_u, C_x, C_u, L_x and L_u, are calculated using the chain rule

$$
\begin{aligned}
\frac{\mathrm{d}f_i}{\mathrm{d}x_j} &= \frac{\partial f_i}{\partial x_j} + \frac{\partial f_i}{\partial a_i}\frac{\mathrm{d}a_i}{\mathrm{d}x_j} + \frac{\partial f_i}{\partial b_i}\frac{\mathrm{d}b_i}{\mathrm{d}x_j} && (8)\\
\frac{\mathrm{d}f_i}{\mathrm{d}u_k} &= \frac{\partial f_i}{\partial a_i}\frac{\mathrm{d}a_i}{\mathrm{d}u_k} + \frac{\partial f_i}{\partial b_i}\frac{\mathrm{d}b_i}{\mathrm{d}u_k}, && (9)
\end{aligned}
$$

recalling that $b_1 = u_1$ and $b_{N/2+1} = u_2$, see Eq. (4). The Jacobians of intermediate variables with respect to the states and inputs are calculated as follows. At the discretization points along the heat exchanger $i = 1, ..., N/2-1, N/2+1, ..., N-1$

$$
\begin{bmatrix} \dfrac{\mathrm{d}a_{i+1}}{\mathrm{d}x_j} \\ \dfrac{\mathrm{d}b_{i+1}}{\mathrm{d}x_j} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial a_{i+1}}{\partial a_i} & \dfrac{\partial a_{i+1}}{\partial b_i} \\ \dfrac{\partial b_{i+1}}{\partial a_i} & \dfrac{\partial b_{i+1}}{\partial b_i} \end{bmatrix} \begin{bmatrix} \dfrac{\mathrm{d}a_i}{\mathrm{d}x_j} \\ \dfrac{\mathrm{d}b_i}{\mathrm{d}x_j} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{\partial b_{i+1}}{\partial x_i} \ \delta_{i,j} \end{bmatrix} \tag{10}
$$

$$\begin{bmatrix} \frac{\mathrm{d}a_{i+1}}{\mathrm{d}u_k} \\ \frac{\mathrm{d}b_{i+1}}{\mathrm{d}u_k} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{i+1}}{\partial a_i} & \frac{\partial a_{i+1}}{\partial b_i} \\ \frac{\partial b_{i+1}}{\partial a_i} & \frac{\partial b_{i+1}}{\partial b_i} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}a_i}{\mathrm{d}u_k} \\ \frac{\mathrm{d}b_i}{\mathrm{d}u_k} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial b_{i+1}}{\partial u_k} \end{bmatrix} \quad (11)$$

with Kronecker delta $\delta_{i,j}$. At the outlets of the k-th heat exchanger $(i = N/2, N)$

$$\begin{bmatrix} \frac{\mathrm{d}a_{O,k}}{\mathrm{d}x_j} \\ \frac{\mathrm{d}b_{O,k}}{\mathrm{d}x_j} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{O,k}}{\partial a_i} & \frac{\partial a_{O,k}}{\partial b_i} \\ \frac{\partial b_{O,k}}{\partial a_i} & \frac{\partial b_{O,k}}{\partial b_i} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}a_i}{\mathrm{d}x_j} \\ \frac{\mathrm{d}b_i}{\mathrm{d}x_j} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial b_{O,k}}{\partial x_i} \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} \frac{\mathrm{d}a_{O,k}}{\mathrm{d}u_k} \\ \frac{\mathrm{d}b_{O,k}}{\mathrm{d}u_k} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_{O,k}}{\partial a_i} & \frac{\partial a_{O,k}}{\partial b_i} \\ \frac{\partial b_{O,k}}{\partial a_i} & \frac{\partial b_{O,k}}{\partial b_i} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}a_i}{\mathrm{d}u_k} \\ \frac{\mathrm{d}b_i}{\mathrm{d}u_k} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial b_{O,k}}{\partial u_k} \end{bmatrix} \quad (13)$$

At the inlets of k-th heat exchanger, the manipulated inputs are the boundary conditions, see Eqs. (3) and (4), thus

$$\begin{aligned} \mathrm{d}b_i/\mathrm{d}x_j &= 0 \ (i = 1, N/2+1) \\ \mathrm{d}b_i/\mathrm{d}u_k &= 1 \ (i = 1, k = 1 \text{ and } i = N/2+1, k = 2) \\ \mathrm{d}b_i/\mathrm{d}u_k &= 0 \ (i = 1, k = 2 \text{ and } i = N/2+1, k = 1), \end{aligned} \quad (14)$$

however the saturation pressure in Eq. (2) is fixed at the outlet, thus

$$\begin{aligned} \mathrm{d}a_{O,k}/\mathrm{d}x_j &= 0 \\ \mathrm{d}a_{O,k}/\mathrm{d}u_k &= 0 \end{aligned} \quad (15)$$

and the partial derivatives at the inlet is calculated as

$$\begin{aligned} \mathrm{d}a_i/\mathrm{d}x_j &= -(\partial a_{O,k}/\partial a_i)^{-1}\ \partial a_{O,k}/\partial x_j \\ \mathrm{d}a_i/\mathrm{d}u_k &= -(\partial a_{O,k}/\partial a_i)^{-1}\ \partial a_{O,k}/\partial u_k. \end{aligned} \quad (16)$$

Here the Jacobians $\partial a_{O,k}/\partial x_j$ and $\partial a_{O,k}/\partial u_k$ are found as $\mathrm{d}a_{O,k}/\mathrm{d}x_j$ and $\mathrm{d}a_{O,k}/\mathrm{d}u_k$ using $\mathrm{d}a_1/\mathrm{d}x_j = 0$ and $\mathrm{d}a_{N/2+1}/\mathrm{d}u_k = 0$, see Eqs. (10) and (11). The partial derivatives $\partial a_{O,k}/\partial a_i$ are calculated in an iterative manner similar to Eqs. (10)–(13).

All other partial derivatives in Eqs. (8)–(13), are generated automatically as in AutoGenU. However, as demonstrated, the choice of the intermediate variables and the implementation of the chain rule are done by hand, thus the resulting code generation procedure is semi-automatic. Regarding other components of H_u and H_x, the cost function L is a sum of terms among which some are independent of the intermediate variables and the corresponding parts of the gradients L_x, L_u can be generated automatically. However, the helium mass flow rate at the heat exchanger outlets is minimized, thus enters the performance index and corresponding parts of the gradients have been generated similarly to and using some sub-expressions of the f_x, f_u. In the system, the Jacobians of the constraints, C_x and C_u, do not involve the intermediate variables and can be generated automatically.

The semi-automatic code generation procedure is much faster and requires much less operational memory than the automatic used in AutoGenU. The observed generation time was seconds vs. 30 min. and the maximum memory used to store all the data for the Mathematica session was 7.9 MB vs. 1034 MB. This makes it possible to apply this type of the semi-automatic code generation process to more complicated systems.

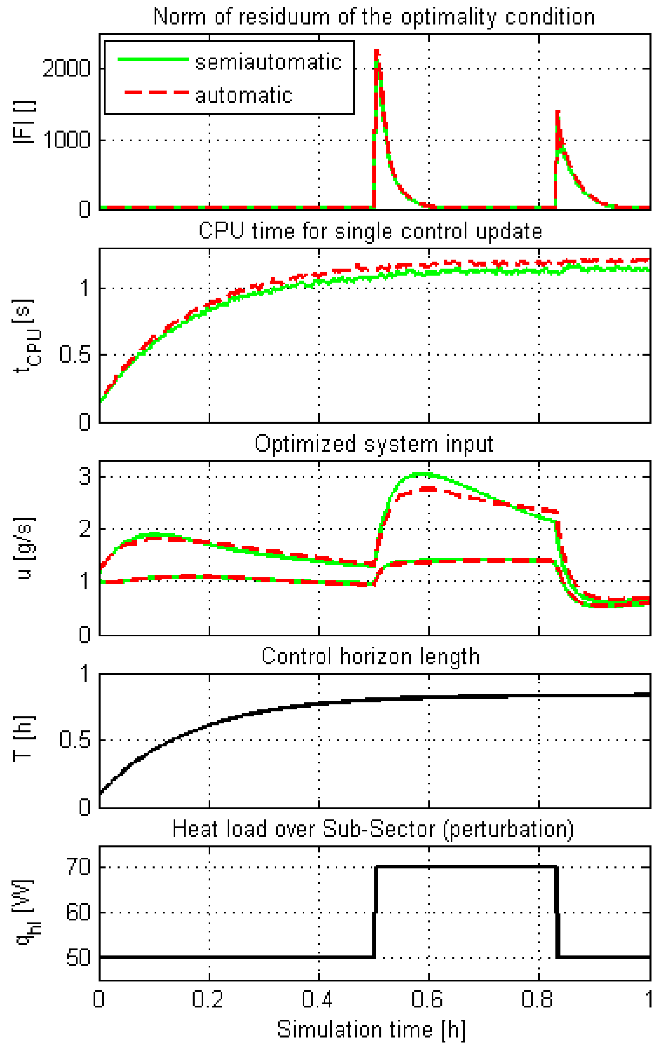


Fig. 1. Comparison of the performance of the automatically and semi-automatically generated controllers.

3. C/GMRES OPTIMALITY CONDITION

In NMPC an open loop optimal control problem is solved over the future prediction time t' horizon taken from the current time t to $t+T$ (Ohtsuka (2004)):

$$\text{minimize } \bar{J} = \phi + \int_t^{t+T} L + \lambda^T \ (f - \dot{x}) + \mu^T \ C \ \mathrm{d}t', \quad (17)$$

with the predicted state trajectory starting at current state: $x'(0,t) = x(t)$. Based on the optimized, predicted future input trajectory $u'(t,t')$, the feedback control is realized by applying only its initial part $u(t) = u'(0,t)$ and continuously repeating the optimization using current measurements and receding the time horizon as the time passes.

In case of stiff dynamics, the state/costate integration step must be very short and in order to separate its length from the control horizon discretization grid t'_i, $i = 1..N_{t'}$, $u'(t',t)$ and $\mu(t',t)$ are parameterized using $N_{t'}$ discrete inputs $u^*_i(t)$ and Lagrange multipliers $\mu^*_i(t)$ (Noga et al. (2010))

$$u'(t',t) = \sum_{i=0}^{N_{t'}-1} \sigma_i(t')\ u_i^*(t) \tag{18}$$

$$\mu(t',t) = \sum_{i=0}^{N_{t'}-1} \sigma_i(t')\ \mu_i^*(t), \tag{19}$$

with basis window functions:

$$\sigma_i(t') = \begin{cases} 1 & \text{if } t'_i \le t' < t'_{i+1} \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

The necessary condition for an extremum of $\bar{J}$, Eq. (17), are: the constraints (7), the costate dynamics

$$\mathrm{d}\lambda/\mathrm{d}t' = -H_x^T(x', u', \lambda, \mu) \tag{21}$$

$$\lambda(t+T,t) = \phi_x^T(x'(t+T,t)), \tag{22}$$

with the Hamiltonian H as in Eq. (6), and a nonlinear equation (Bryson and Ho (1975); Ohtsuka (2004); Noga et al. (2010)),

$$F(U(t), x(t), t) = 0, \tag{23}$$

$$F := [H_{u,0}\ C_0^T\ \cdots\ H_{u,N_{t'}-1}\ C_{N_{t'}-1}^T]^T, \tag{24}$$

$$U(t) := [u_0^{*T}\ \mu_0^{*T}\ \cdots\ u_{N_{t'}-1}^{*T}\ \mu_{N_{t'}-1}^{*T}]^T, \tag{25}$$

with the integrals

$$H_{u,i} := \int_{t'_i}^{t'_{i+1}} H_u\ \mathrm{d}t',\ C_i := \int_{t'_i}^{t'_{i+1}} C\ \mathrm{d}t'. \tag{26}$$

For a given sequence of $u_i^*(t)$ and $\mu_i^*(t)$, $\mathrm{d}x'/\mathrm{d}t'$ is integrated over the finite horizon $t < t' < t+T$, starting from $x(t)$. Then $\mathrm{d}\lambda/\mathrm{d}t'$ is integrated backwards from $t+T$ back to t. Finally, $H_{u,i}$ and C_i are evaluated and assembled into the residuum of the necessary optimality condition F.

Since H_u expressed directly in terms of system inputs and states has high computational cost, the integrals $H_{u,i}$ are calculated using simple quadratures such as one-point rectangular or two-points trapezoidal, where H_u is evaluated exclusively at the horizon grid. However, once the intermediate variables are introduced and are then available at each state integration step, the H_u may also be evaluated at each step at low additional computational cost, resulting in a more precise calculation of the optimality condition. Also the derivative $\mathrm{d}a_{O,k}/\mathrm{d}a_{I,k}$ employed in the Newton iteration, Eq. (5), is calculated using the intermediate variables.

The performance of the C/GMRES controller generated using automatic and semi-automatic procedure has been simulated, see Fig. 1. In the simulation, the prediction horizon length increases at the beginning to reach $\max(T) = 50$ min and is discretized with $N_{t'} = 10$ intervals. The time step for system dynamics integration is 62 times shorter than the horizon grid interval, thus the integrals $H_{u,i}$ and C_i are evaluated using one point and 62 points in automatic and semi-automatic code, respectively. The observed computational cost of the semi-automatically generated code is slightly less than that of automatically generated and the controller performance is similar in both cases.

4. CONCLUSIONS

AutoGenU automatically generates the C code of the Jacobians H_x and H_u, required to evaluate the necessary optimality condition in NMPC. For the case of the SHCC, which is a distributed parameter system, the Jacobians expressed using symbolic mathematics directly in terms of system inputs, states and co-states are complex due to the propagation of expressions along the cascading structure of the discretized system model. Based on AutoGenU, a semi-automatic NMPC code generation procedure has been developed for the SHCC, where intermediate variables are used to avoid this propagation, thereby reducing the complexity of the Jacobians. In the case of a stiff system, the time step of the system state dynamics integration is small and the intermediate values, which are available at each step, are used to evaluate the optimality condition more precisely at low additional computational cost. The observed computational cost of control update calculated using the semi-automatic code is slightly less than that generated automatically and the controller performance is similar in both cases. However, this semi-automatic code generation process requires significantly less memory, which makes it possible to apply the code generation process with analytically calculated Jacobians to more complicated systems.

ACKNOWLEDGMENTS

Part of this work was done during the stay of the first author at Osaka University for the "FrontierLab@OsakaU" programme. This work was supported by Spanish Government (project DPI2009-12805), University of Valladolid, CERN, Osaka University, Japan Student Services Organization and Grant-in-Aid for Scientific Research (C) No. 21560465. This support is very gratefully acknowledged.

REFERENCES

Brüning, O., Collier, P., Lebrun, P., Myers, S., Ostojic, R., Poole, J., and Proudlock, P. (2004). *LHC Design Report.* CERN.

Bryson, A.E. and Ho, Y. (1975). *Applied optimal control : optimization, estimation, and control.* Hemisphere Pub. Corp.

Gubello, G., Serio, L., and Soubiran, M. (2006). The circuits of the LHC Cryogenic System. Engineering Specification.

Lebrun, P., Serio, L., Tavian, L., and Van Weelderen, R. (1997). Cooling Strings of Superconducting Devices below 2 K: the Helium II Bayonet Heat Exchanger. *Adv. Cryog. Eng., A*, 43A, 419 426.

Noga, R. (2007). *Modeling and control of the String2 LHC Prototype at CERN.* Master's thesis, Gdansk University of Technology, University of Karlsruhe, Grenoble Institute of Technology.

Noga, R., Ohtsuka, T., de Prada, C., Blanco, E., and Casas, J. (2010). Nonlinear Model Predictive Control for the Superfluid Helium Cryogenic Circuit of the Large Hadron Collider. In *Proceedings of the 2010 IEEE International Conference on Control Applications*, 1654 – 1659.

Ohtsuka, T. (2000). AutoGenU: Readme.txt. URL `http://www-sc.sys.es.osaka-u.ac.jp/~ohtsuka/`.

Ohtsuka, T. (2004). A continuation/GMRES method for fast computation of nonlinear receding horizon control. *Automatica*, 40(4), 563 – 574.

IMPROVEMENT OF THE DECOUPLING EFFECT OF THE PREDICTIVE CONTROLLER GPC AND PFC BY PARAMETER ADAPTATION

K. Zabet, R. Haber

Institute of Process Engineering and Plant Design, Laboratory of Process Automation,
Cologne University of Applied Science, D-50679 Köln, Betzdorfer Str. 2, Germany
fax: +49-221-8275-2836 and e-mail: khaled.zabet@smail.fh-koeln.de, robert.haber@ fh-koeln.de

Abstract: Two simple techniques are presented and compared for predictive control of TITO (Two-Input, Two-Output) processes to improve the decoupling effect. These techniques are applied for GPC (Generalized Predictive Control) and PFC (Predictive Functional Control). According to the first technique the controller parameters are tuned in synchronization to a reference signal change. According to the second one the controller parameters are set dependent on the actual control error. The second method makes the synchronization to a reference signal change superfluous and its realization is therefore very easy.

Keywords: Generalized predictive control, predictive functional control, controller parameter adaptation, control error-dependent controller parameters

1. INTRODUCTION

Improvement of the decoupling effect in multivariable processes is an important issue. It is desired that change of one reference signal would affect mainly on the corresponding controlled variable, while the effect on the others with constant reference signal would be reduced, i.e. the control error of the other controlled variables would be minimized (Maurath, Seborg and Mellichamp, 1986). MIMO (Multi-Input, Multi-Output) controllers can handle this problem using manually designed decoupling controllers or MIMO predictive controller which enhances the decoupling automatically.

The question arises how the decoupling can be improved without complicated multivariable controller design. In this paper two different methods are recommended for multivariable control of stable aperiodic processes. The TITO controller is realized by GPC (Generalized predictive control) (Clarke et. al., 1987) and PFC (Predictive Functional Control) (Richalet and O'Donavan, 2009).

The paper is structured as follows. In section 2 the TITO GPC algorithm is shown. In section 3 the TITO PFC algorithm is shown. In section 4 a TITO process is controlled by both predictive control algorithms with fixed controller parameters. In sections 5 and 6 two different methods are shown how the controller parameters of the two predictive control algorithms can be adapted to decrease the coupling effect. The results are summarized in the conclusion.

2. GENERALIZED PREDICTIVE CONTROL

The cost function of a TITO predictive control is:

$$\begin{aligned}J = \lambda_{y1} \sum_{n_e=n_{e11}}^{n_{e21}} [y_{r1}(k+d_1+1+n_e) - \hat{y}_1(k+d_1+1+n_e \mid k)]^2 \\ + \lambda_{y2} \sum_{n_e=n_{e12}}^{n_{e22}} [y_{r2}(k+d_2+1+n_e) - \hat{y}_2(k+d_2+1+n_e \mid k)]^2 \\ + \lambda_{u1} \sum_{j=0}^{n_{u1}-1} \Delta u_1^2(k+j) + \lambda_{u2} \sum_{j=0}^{n_{u2}-1} \Delta u_2^2(k+j) \Rightarrow \underset{\mathbf{\Delta u}}{MIN}\end{aligned} \qquad (1)$$

with the denotations:

- $y_{ri}(k+d_i+1+n_e \mid k)$: reference signal of the i-th output n_e steps over the dead time d_i,
- $\hat{y}_i(k+d_i+1+n_e \mid k)$: predicted i-th output signal n_e steps over the dead time.

The tuning parameters of the control algorithm in (1) are:

- $n_{e2i} - n_{e1i} + 1$: length of the prediction horizon for the i-th output,
- n_{ui}: length of the control horizon of the i-th input,
- λ_{yi}: control error weighting factor of the i-th output,
- λ_{ui}: control increments weighting factor of the i-th input.

The control increments vector in the control horizon from k to $k+n_{ui}-1$ which has to be optimized is:

$$\begin{aligned}\mathbf{\Delta u}_1 &= [\Delta u_1(k \mid k) \ \dots \ \Delta u_1(k+n_{u1}-1 \mid k)]^T \\ \mathbf{\Delta u}_2 &= [\Delta u_2(k \mid k) \ \dots \ \Delta u_2(k+n_{u2}-1 \mid k)]^T \\ \mathbf{\Delta u} &= \left[\mathbf{\Delta u}_1^T \quad \mathbf{\Delta u}_2^T\right]^T\end{aligned} \qquad (2)$$

The predicted i-th output vector in the future time domain (prediction horizon) from $k+d_i+1+n_{e1i}$ to $k+d_i+1+n_{e2i}$ can be divided into free and forced responses:

$$\begin{aligned}\hat{\mathbf{y}}_i &= \hat{\mathbf{y}}_{i,free} + \hat{\mathbf{y}}_{i,forc} \\ \begin{bmatrix} \hat{y}_i(k+d_i+1+n_{e1i} \mid k) \\ \vdots \\ \hat{y}_i(k+d_i+1+n_{e2i} \mid k) \end{bmatrix} &= \begin{bmatrix} \hat{y}_{i,free}(k+d_i+1+n_{e1i} \mid k) \\ \vdots \\ \hat{y}_{i,free}(k+d_i+1+n_{e2i} \mid k) \end{bmatrix} + \hat{\mathbf{y}}_{i,forc}\end{aligned} \qquad (3)$$

The predicted forced i-th output vector in (3) is:

$$\hat{\mathbf{y}}_{i,forc} = \sum_{j=1}^{2} \mathbf{H}_{ij} \mathbf{\Delta u}_j \qquad (4)$$

where:

$$\mathbf{H}_{ij} = \begin{bmatrix} h_{ij}(n_{e1i}+1) & h_{ij}(n_{e1i}) & \cdots & h_{ij}(n_{e1i}-n_{uj}+2) \\ h_{ij}(n_{e1i}+2) & h_{ij}(n_{e1i}+1) & \cdots & h_{ij}(n_{e1i}-n_{uj}-1) \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ h_{ij}(n_{e2i}+1) & h_{ij}(n_{e2i}) & \cdots & h_{ij}(n_{e2i}-n_{uj}+2) \end{bmatrix} \qquad (5)$$

whereas $\mathbf{H}_{ij}$ is the matrix of step response coefficients of the process model, and $h_{ij}(k)=0$ if $k<0$.

For the TITO process, the predicted vectors (in the prediction horizon) of the reference signals, process outputs, free responses and forced responses are respectively:

- $\mathbf{y}_r = \left[\mathbf{y}_{r1}^T, \mathbf{y}_{r2}^T\right]^T$: predicted reference signals,
- $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_1^T, \hat{\mathbf{y}}_2^T]^T$: predicted outputs,
- $\hat{\mathbf{y}}_{free} = [\hat{\mathbf{y}}_{1free}^T, \hat{\mathbf{y}}_{2free}^T]^T$: predicted free responses,
- $\hat{\mathbf{y}}_{forc} = [\hat{\mathbf{y}}_{1forc}^T, \hat{\mathbf{y}}_{2forc}^T]^T$: predicted forced outputs.

The predicted vector of the forced responses is:

$$\hat{\mathbf{y}}_{forc} = \begin{bmatrix} \sum_{j=1}^{2} \mathbf{H}_{1j}\Delta\mathbf{u}_j \\ \sum_{j=1}^{2} \mathbf{H}_{2j}\Delta\mathbf{u}_j \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u}_1 \\ \Delta\mathbf{u}_2 \end{bmatrix} = \mathbf{H}\Delta\mathbf{u} \tag{6}$$

The cost function (1) becomes:

$$J = \left(\mathbf{y}_r - \hat{\mathbf{y}}\right)^T \mathbf{\Lambda}_y \left(\mathbf{y}_r - \hat{\mathbf{y}}\right) + \Delta\mathbf{u}^T \mathbf{\Lambda}_u \Delta\mathbf{u} \Rightarrow \underset{\Delta\mathbf{u}}{MIN} \tag{7}$$

with the diagonal weighting matrices (for simplicity) of the control errors and the control increments:

$$\mathbf{\Lambda}_y = \mathbf{\Lambda}_y^T = \mathbf{diag}\left\langle \mathbf{\Lambda}_{y1}, \mathbf{\Lambda}_{y2} \right\rangle = \mathbf{diag}\left\langle \lambda_{y1}\mathbf{I}, \lambda_{y2}\mathbf{I} \right\rangle;$$
$$\mathbf{\Lambda}_u = \mathbf{\Lambda}_u^T = \mathbf{diag}\left\langle \mathbf{\Lambda}_{u1}, \mathbf{\Lambda}_{u2} \right\rangle = \mathbf{diag}\left\langle \lambda_{u1}\mathbf{I}, \lambda_{u2}\mathbf{I} \right\rangle$$

and $\mathbf{I}$ is the identity matrix.

Substituting of free and forced responses vectors results in:

$$\begin{aligned} J &= \left(\mathbf{y}_r - \hat{\mathbf{y}}_{free} - \mathbf{H}\Delta\mathbf{u}\right)^T \mathbf{\Lambda}_y \left(\mathbf{y}_r - \hat{\mathbf{y}}_{free} - \mathbf{H}\Delta\mathbf{u}\right) \\ &\quad + \Delta\mathbf{u}^T \mathbf{\Lambda}_u \Delta\mathbf{u} \Rightarrow \underset{\Delta\mathbf{u}}{MIN} \end{aligned} \tag{8}$$

Unconstrained minimization of the cost function (8) according to the whole sequence of input increments in the control time domain leads to:

$$\begin{aligned} \frac{dJ(\Delta\mathbf{u})}{d\Delta\mathbf{u}} &= -\mathbf{H}^T \left[\mathbf{\Lambda}_y^T + \mathbf{\Lambda}_y\right]\left(\mathbf{y}_r - \hat{\mathbf{y}}_{free} - \mathbf{H}\Delta\mathbf{u}\right) \\ &\quad + \left[\mathbf{\Lambda}_u^T + \mathbf{\Lambda}_u\right]\Delta\mathbf{u} = \mathbf{0} \end{aligned}$$

which results in

$$\Delta\mathbf{u} = \left[\mathbf{H}^T \mathbf{\Lambda}_y \mathbf{H} + \mathbf{\Lambda}_u\right]^{-1} \mathbf{H}^T \mathbf{\Lambda}_y \left(\mathbf{y}_r - \hat{\mathbf{y}}_{free}\right) \tag{9}$$

According to the receding horizon technique only the actual control signals will be used and the computation is repeated in the next control step:

$$\Delta\mathbf{u}_{actual}(k) = [\Delta u_1(k), \Delta u_2(k)]^T \tag{10}$$

3. PREDICTIVE FUNCTIONAL CONTROL

The principle of SISO PFC with constant reference signal is that the controlled variable achieves the reference trajectory at the target point using one change in the manipulated variable. The desired change in the controlled variable during the prediction horizon n_p (from the actual time k) is calculated from the change of the reference trajectory and compared to the predicted change of the non-delayed model output to define the required control signal, see Fig. 1.

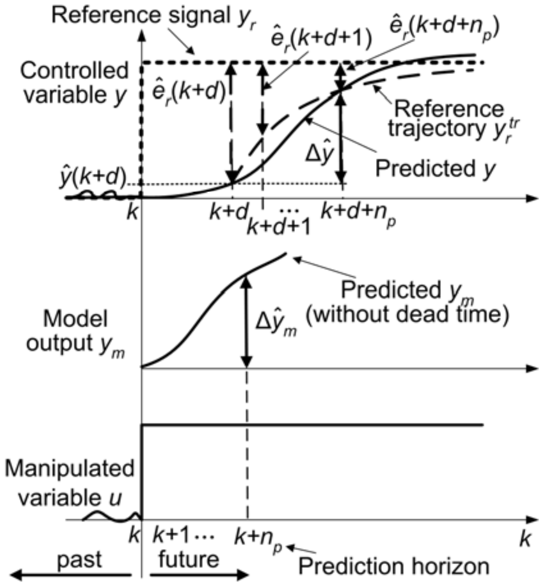


Fig. 1. PFC principle of processes with dead time

The aim of the control equation is:

$$\begin{aligned} &\hat{y}(k+d_m+n_p\,|\,k) - \hat{y}(k+d_m\,|\,k) = \hat{y}_m(k+n_p\,|\,k) - y_m(k) \\ &(1-\lambda_r^{n_p})[y_r - \hat{y}(k+d_m\,|\,k)] = \hat{y}_m(k+n_p\,|\,k) - y_m(k) \end{aligned} \tag{11}$$

with the denotations:

- $\hat{y}(k+d_m+n_p\,|\,k)$: predicted controlled variable n_p steps over the dead time d_m,
- y_r : reference signal (supposed constant in the future),
- $\hat{y}_m(k+n_p\,|\,k)$: predicted non-delayed model output n_p steps over the actual time,
- λ_r : reduction ratio of the bias between the reference signal and its trajectory.

The controller parameters (for sampling time Δt) are:

- $T_c = -3\Delta t/\ln(\lambda_r)$: desired closed loop settling time
- n_p : prediction horizon (≥ 1)

The control equation of PT1 (proportional, 1st-order) process with dead time (chosen for simplicity) is described as:

$$u(k) = k_0[y_r - \hat{y}(k+d_m\,|\,k)] + k_1 y_m(k) \tag{12}$$

where:

$$\hat{y}(k+d_m\,|\,k) = y(k) + [y_m(k) - y_m(k-d_m)] \tag{13}$$

- $k_0 = \dfrac{1-\lambda_r^{n_p}}{K_m[1-(-a_m)^{n_p}]}$, $k_1 = \dfrac{1}{K_m}$: controller coefficients
- a_m: discrete-time model parameter
- K_m: static gain of the model

In case of *n*-th order aperiodic processes the transfer function of the non-delayed model can be partitioned in parallel connection of *n* first-order models with the corresponding parameters $K_{i,m}$ and $a_{i,m}$ of *i*-th sub-model. (If the model has multiple poles then different but very similar poles have to be assigned to each multiple pole.)

The basic algorithm can be easily extended for this case, as well (Khadir and Ringwood, 2008):

$$u(k) = k_0\{y_r - [y(k) + [y_m(k) - y_m(k-d_m)]]\} + \sum_{i=1}^{n} k_i y_{i,m}(k) \quad (14)$$

where:

$$k_0 = \frac{1-\lambda_r^{n_p}}{\sum_{i=1}^{n} K_{i,m}[1-(-a_{i,m})^{n_p}]} \quad ; \quad k_i = \frac{1-(-a_{i,m})^{n_p}}{\sum_{j=1}^{n} K_{j,m}[1-(-a_{j,m})^{n_p}]}$$

and, discrete-time equation of *i*-th sub-model is:

$$y_{i,m}(k) = -a_{i,m}y_{i,m}(k-1) + K_{i,m}(1+a_{i,m})u(k-1) \quad (15)$$

The algorithm is extended for TITO processes with the following tuning parameters:

- $T_{ci} = -3\Delta t / \ln(\lambda_{ri})$: desired closed loop settling time of the *i*-th controlled variable,
- n_{pi}: prediction horizon of the *i*-th controlled variable.

The discrete dead time of *i*-th output signal is supposed as:

d_{im} = max (d_{i1m}, d_{i2m})
where d_{ijm} is the discrete dead time of the model with *j*-th input signal and *i*-th output signal.

Thus, these relations can be defined:

$$y_{im}(k-d_{im}) = y_{i1m}(k-d_{i1m}) + y_{i2m}(k-d_{i2m}) \quad (16)$$

whereas y_{ijm} is the non-delayed model output, and $y_{ijm}(k-d_{ijm})$ should represents $y_{ij}(k)$, thus:

$$y_{im}(k-d_{im}) \cong y_{i1}(k) + y_{i2}(k) = y_i(k)$$

$$\hat{y}_{im}(k|k) \cong \hat{y}_i(k+d_{im}|k)$$

$$\hat{y}_{im}(k|k) = \hat{y}_{i1m}(k-d_{i1m}+d_{im}|k) + \hat{y}_{i2m}(k-d_{i2m}+d_{im}|k) \quad (17)$$

$$\hat{y}_i(k+d_{im}|k) \cong y_i(k) + [\hat{y}_{im}(k|k) - y_{im}(k-d_{im})] \quad (18)$$

From (18), the predicted increment of *i*-th controlled variable n_{pj} step ahead the instant $k + d_{im}$ is defined as:

$$\hat{y}_i(k+d_{im}+n_{pi}|k) - \hat{y}_i(k+d_{im}|k) = (1-\lambda_{ri}^{n_{pj}})[y_{ri} - y_i(k) - \hat{y}_{im}(k|k) + y_{im}(k-d_{im})] \quad (19)$$

The predicted increment of *i*-th process model output n_{pj} step ahead the current *k* is defined based on (17) as:

$$\hat{y}_{im}(k+n_{pi}|k) - \hat{y}_{im}(k|k) = \hat{y}_{i1m}(k-d_{i1m}+d_{im}+n_{pi}|k) - \hat{y}_{i1m}(k-d_{i1m}+d_{im}|k) + \hat{y}_{i2m}(k-d_{i2m}+d_{im}+n_{pi}|k) - \hat{y}_{i2m}(k-d_{i2m}+d_{im}|k) \quad (20)$$

This equation in (20) can be reformulated using free and forced responses:

$$\begin{aligned}\hat{y}_{im}(k+n_{pi}|k) - \hat{y}_{im}(k|k) = {}& \hat{y}_{ifree}(k+n_{pi}|k) - \hat{y}_{ifree}(k|k)\\ &+\hat{y}_{i1forc}(k-d_{i1m}+d_{im}+n_{pi}|k)\\ &-\hat{y}_{i1forc}(k-d_{i1m}+d_{im}|k)\\ &+\hat{y}_{i2forc}(k-d_{i2m}+d_{im}+n_{pi}|k)\\ &-\hat{y}_{i2forc}(k-d_{i2m}+d_{im}|k)\end{aligned} \quad (21)$$

whereas:

$$\hat{y}_{ifree}(k|k) = \hat{y}_{i1free}(k-d_{i1m}+d_{im}|k) + \hat{y}_{i2free}(k-d_{i2m}+d_{im}|k)$$

$$\hat{y}_{ifree}(k+n_{pi}|k) = \hat{y}_{i1free}(k-d_{i1m}+d_{im}+n_{pi}|k) + \hat{y}_{i2free}(k-d_{i2m}+d_{im}+n_{pi}|k)$$

Based on (19) and (21), PFC goal leads to these two control equations (for *i*=1 and 2):

$$\begin{aligned}&(1-\lambda_{ri}^{n_{pi}})[y_{ri} - y_i(k) - \hat{y}_{ifree}(k|k) + y_{im}(k-d_{im})]\\ &+ \hat{y}_{ifree}(k|k) - \hat{y}_{ifree}(k+n_{pi}|k) =\\ &\quad \hat{y}_{i1forc}(k-d_{i1m}+d_{im}+n_{pi}|k)\\ &\quad -\lambda_{ri}^{n_{pi}}\hat{y}_{i1forc}(k-d_{i1m}+d_{im}|k)\\ &\quad +\hat{y}_{i2forc}(k-d_{i2m}+d_{im}+n_{pi}|k)\\ &\quad -\lambda_{ri}^{n_{pi}}\hat{y}_{i2forc}(k-d_{i2m}+d_{im}|k)\end{aligned} \quad (22)$$

The free and forced responses of the process model with *j*-th input signal and *i*-th output signal (which is partitioned in parallel connection of n_{ij} first-order sub-models) are:

$$\hat{y}_{ij,free}(k+n|k) = \sum_{r=1}^{n_{ij}} (-a_{ijm,r})^n y_{ijm,r}(k),$$

$$\hat{y}_{ij,forc}(k+n|k) = u_j(k)\sum_{r=1}^{n_{ij}} K_{ijm,r}[1-(-a_{ijm,r})^n].$$

The solutions of these equations (22) in the two manipulated variables $u_i \equiv u_i(k)$; *i*=1,2 are calculated (if they are unique) in every control step using the same algorithm. Otherwise when the solutions are not unique (one equation of two variables which has infinite solutions) the tuning parameters can be changed in order to get a unique solution of the control equations, or the solution with minimum increments can be defined by solving this criteria function:

$$J = [u_1 - u_1(k-1)]^2 + [u_2 - u_2(k-1)]^2 \Rightarrow \underset{u_1,u_2}{MIN} \quad (23)$$

$$\frac{dJ}{du_1} = [u_1 - u_1(k-1)] + [u_2 - u_2(k-1)]\frac{du_2}{du_1} = 0 \quad (24)$$

whereas u_2 and $\frac{du_2}{du_1}$ are defined from one of the two equivalent linear equations (22) of the variables u_1 and u_2.

4. DECOUPLING PREDICTIVE CONTROL OF A TITO PROCESS

In order to illustrate the problem of coupling a TITO process is considered with set of the sampling time Δt=0.1 min. The sub-processes are aperiodic with different static gains K_{ij}, time constants T_{ij}, and dead times T_{dij}. All processes have some (n_{ij}) equal time constants:

- P_{11}: K_{11}=1.5, T_{11}=1.0 min, n_{11}=2, T_{d11}=0.1 min
- P_{12}: K_{12}=0.5, T_{12}=0.5 min, n_{12}=4, T_{d12}=0.5 min
- P_{21}: K_{21}=0.75, T_{21}=0.5 min, n_{21}=3, T_{d21}=0.8 min
- P_{22}: K_{22}=1.0, T_{22}=2.0 min, n_{22}=1, T_{d22}=0.2 min

The step responses of the processes were shown in Fig. 2.

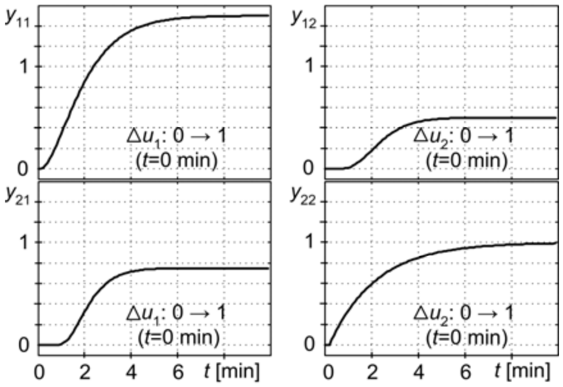


Fig. 2. Step responses of the TITO sub-processes

TITO predictive control was used; see the scheme in Fig. 3.

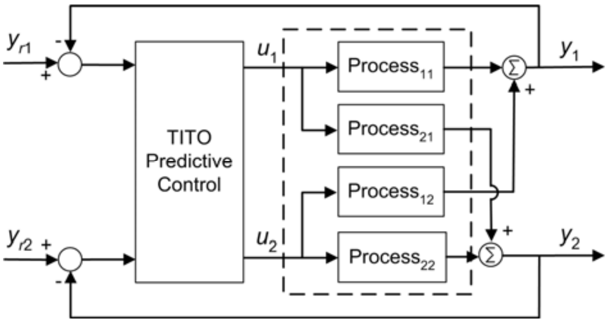


Fig. 3. TITO predictive control scheme

The control scenario was:

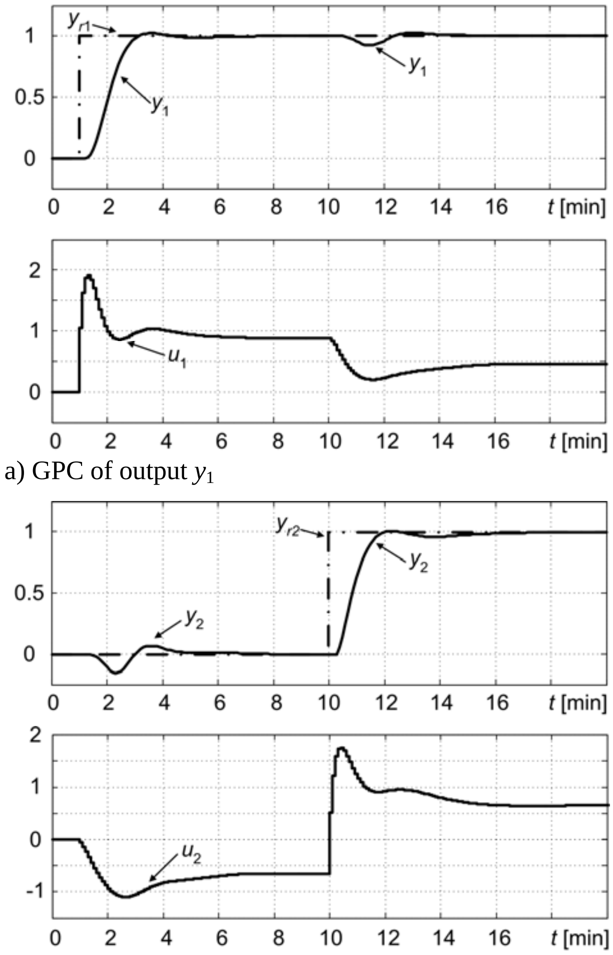
- at t=1 min stepwise increase of the reference signal of y_1 from 0 to 1,
- at t=10 min stepwise increase of the reference signal of y_2 from 0 to 1.

4.1 GPC of TITO process

GPC of TITO process is shown in Fig. 4 with the following controller parameters:

- start of control error horizons: n_{e11}=n_{e12}=0
- end of control error horizons: n_{e21}=40 and n_{e22}=30
- length of control horizons: n_{u1}=n_{u2}=3
- weighting factors of the control errors λ_{y1}=λ_{y2}=1
- weighting factors of the control increments λ_{u1}=λ_{u2}=0.5

The control of the reference signal changes is fast and aperiodic. The maximal control error of the controlled variable y_1 is about 6.5% (related to the changes of the reference signal y_{r2}), and is about 16.5% maximal control error of the controlled variable y_2 (related to the changes of the reference signal y_{r1}).



a) GPC of output y_1

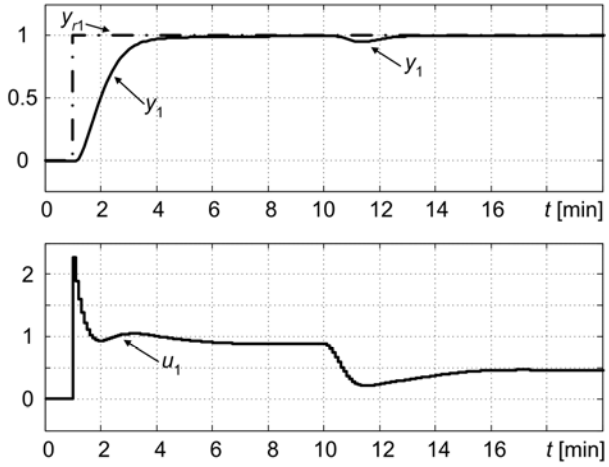
b) GPC of output y_2

Fig. 4. GPC of TITO process

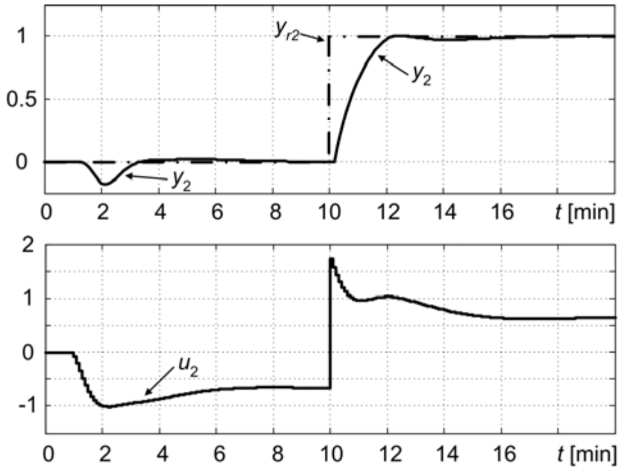
4.2 PFC of TITO process

PFC of TITO process is shown in Fig. 5 with the following controller parameters: settling times T_{c1} = 2 min. and T_{c2} = 1.5 min. and prediction horizons n_{p1} = n_{p2} = 3.

The control of the reference signal changes is fast and aperiodic. The maximal control error of the controlled variable y1 is about 5.2% (related to the changes of the reference signal yr2), and is about 22% maximal control error of the controlled variable y2 (related to the changes of the reference signal yr1).



a) PFC of output y_1



b) PFC of output y_2

Fig. 5. PFC of TITO process

Fig. 4 and 5 shows that the rising times and the maximal control error related to the coupling effect are similar with GPC and PFC for this set of controller parameters, although PFC has a smaller number of the controller parameters and less calculations than GPC. The manipulated variables of the process with GPC and PFC have a similar shape Also.

5. REFERENCE SIGNAL CHANGE-DEPENDENT ADAPTION OF THE CONTROLLER PARAMETER

The time point of the reference signal change in known sometimes by the technology in advance. Otherwise it can be detected with methods of signal analysis.

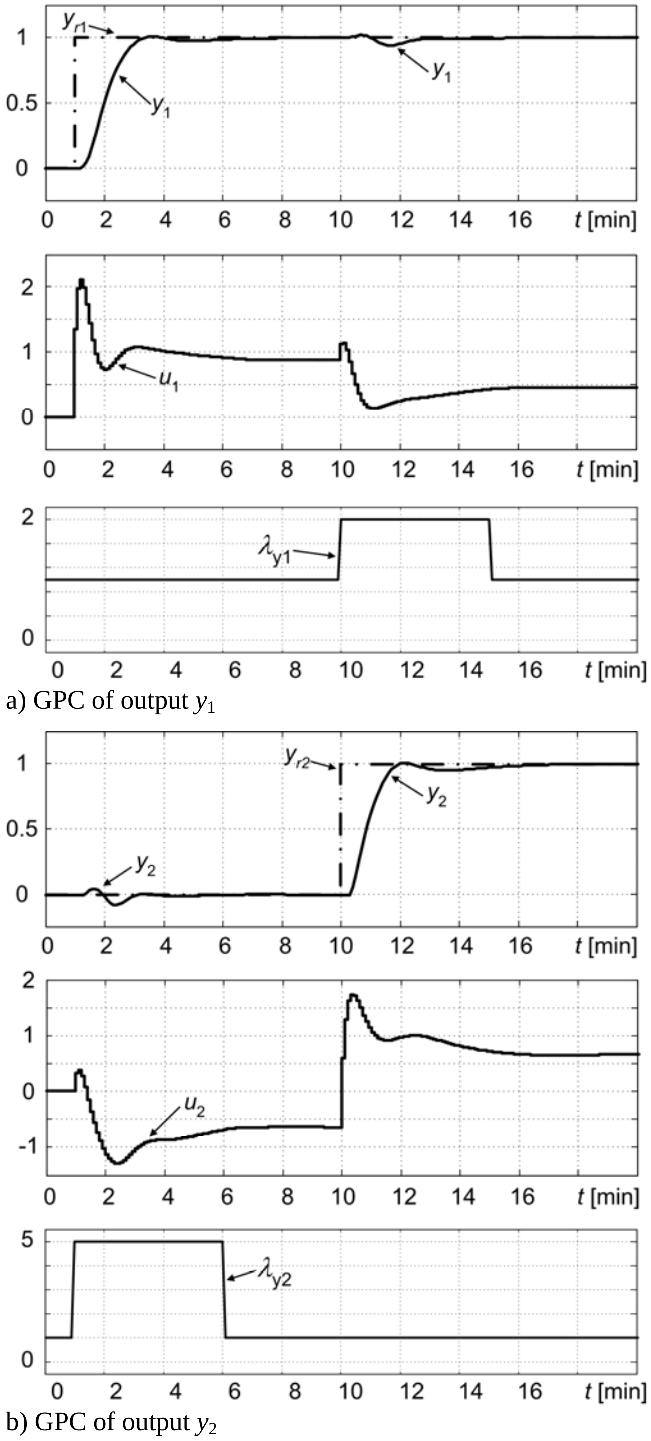
5.1 Reference signal change-dependent adaptation of GPC parameters

The control equation of GPC shows that the increasing of the control error weighting factor of one controlled variable shall reduce the control error in that variable. Therefore increasing of the control error weighting factor of the controlled variable whose reference signal was kept constant reduces the control error in this variable.

This technique is illustrated in Fig. 6 for reference signal changes. The weighting factors of both control errors were changed stepwise from λ_{y1}=λ_{y2}=1 to λ_{y1}=2 and λ_{y2}=5 for that variable whose reference signal was kept constant in the moment of the other reference signal change. The duration of the weighting factors change was 5 min which is about 2 min longer than the settling time of the controlled process.

The plots show that the two controlled variables are better decoupled. The maximal control error of the controlled variable y_1 is about 5.6% (related to the changes of the reference signal y_{r2}), and is about 8% maximal control error of the controlled variable y_2 (related to the changes of the reference signal y_{r1}).

The critical point of the manual controller parameters adaptation is the detection of the reference signal change. Nevertheless a method which does not care about the time point of the reference signal change would be preferable.



a) GPC of output y_1

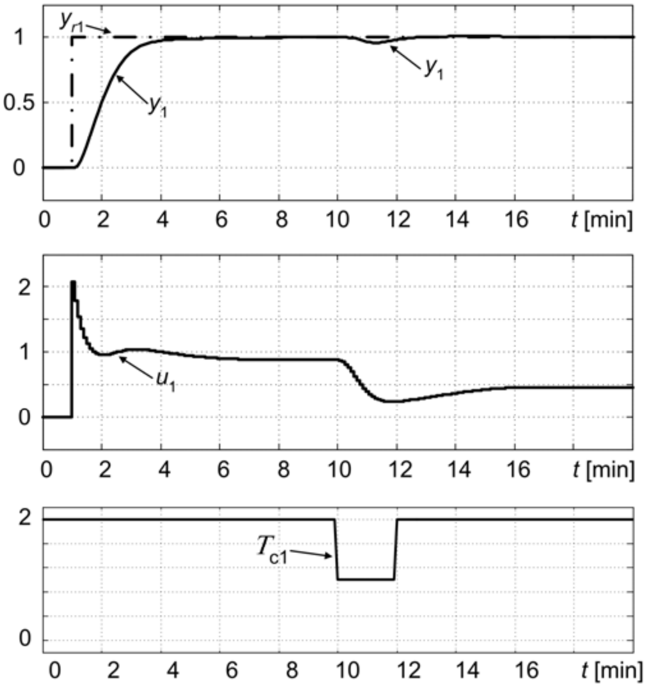
b) GPC of output y_2

Fig. 6. TITO GPC with reference signals changes-dependent adaptation of λ_y

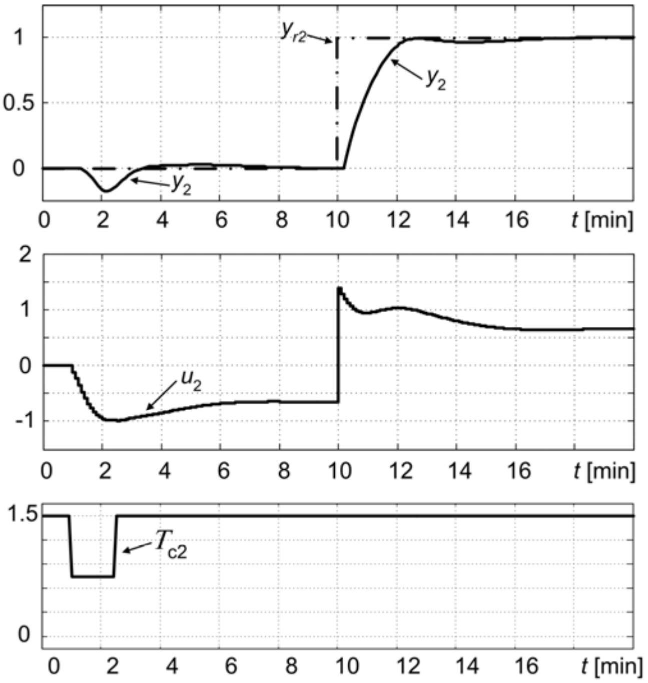
5.2. Reference signal change-dependent adaptation of PFC parameters

The main controller parameter with PFC is the settling time T_{c}. The decoupling ability with a TITO process can be improved by tuning the settling times (T_{c1} and T_{c2}). Decreasing of the desired settling time of the controlled variable whose reference signal was kept constant accelerates the control and hence reduces the control error in this controlled variable.

Fig. 7 illustrates this case for reference signal changes. The desired settling time of the first controlled variable was changed stepwise from T_{c1}=2 min to T_{c1}=1 min and the desired settling time of second controlled variable was changed stepwise from T_{c2}=1.5 min to T_{c2}=0.75 min. The desired settling times were changed for that controlled variable whose reference signal was not changed in the moment of the change of the other reference signal. The duration of the changes were equal to the desired settling times for both. The maximal control error is about 4% (related to the changes of y_{r2}) in the controlled variable y_1, and about 18% (related to the changes of y_{r1}) in the controlled variable y_2. The plots show that the two processes are fast and better decoupled than with constant settling times but worse than with GPC in Fig. 6.



a) PFC of output y_1



b) PFC of output y_2

Fig. 7. TITO PFC with reference signals changes- dependent adaptation of T_c

6. CONTROL ERROR-DEPENDENT ADAPTATION OF THE CONTROLLER PARAMETERS

The synchronisation at the reference signal change can be performed automatically if the highlighted controller parameters were decentralized functions of the control errors.

6.1 Control error-dependent adaptation of GPC parameters

The control error of the controlled variable whose reference signal was changed increases faster than the control error of the other variable whose reference signal was kept constant. Consequently, if the control error weighting factors are set inverse proportional to the control error for both controlled variables then the weighting factor of the controlled variable whose reference signal was kept constant will be higher than the weighting factor of the controlled variable whose reference signal was changed.

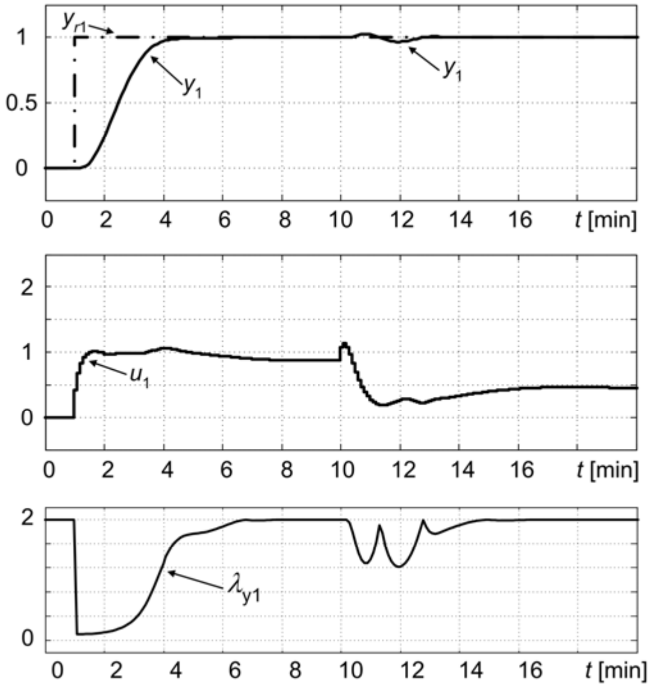
The following dependence of the weighting factors on the control error were supported (Schmitz, et. al., 2007):

$$\lambda_{yi} = \frac{\lambda_{yi,\max}}{\left(1+\left|e_i(k)\right|\cdot\lambda_{yi,damp}\right)}$$

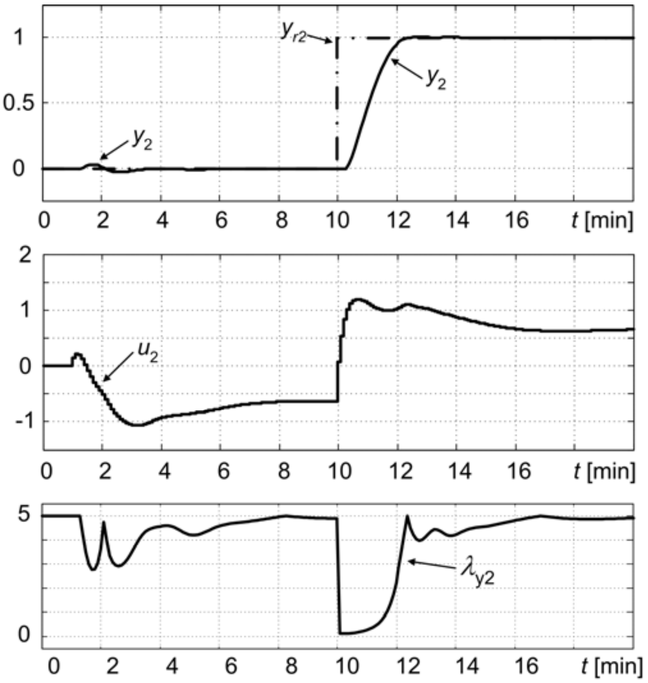
with $\lambda_{y1,\max}$=2, $\lambda_{y2,\max}$=5, $\lambda_{y1,\mathrm{damp}}$=20 and $\lambda_{y2,\mathrm{damp}}$=25 in this case.

Fig. 8 shows that the weighting factors of those controlled variables whose reference signal was changed were temporarily significantly reduced and the other weighting factor is remained big, this behaviour is in opposite to Fig. 6.

The control is slightly slower than with the changing of the weighting factors at the reference signal changes (Fig. 6) but the control is still fast and the decoupling is better than before. The automatic adaptation of the control error weighting factors shows about 3.2% maximal control error (related to the changes of y_{r2}) in the controlled variable y_1, and about 3.1% (related to the changes of y_{r1}) in the controlled variable y_2.



a) GPC of output y_1



b) GPC of output y_2

Fig. 8. TITO GPC with control error-dependent adaptation of λ_y

6.2 Control error-linear dependent adaptation of PFC parameters

The settling times can be set proportional to the related control error; therefore the settling time of the controlled variable whose reference signal was changed will be higher than the settling time of the controlled variable whose reference signal was kept constant. Consequently the controlled variable whose reference signal was not changed will be controlled faster, that acts as a forced decoupling.

The following linear dependence of the desired settling times on the control error were applied in the simulation:

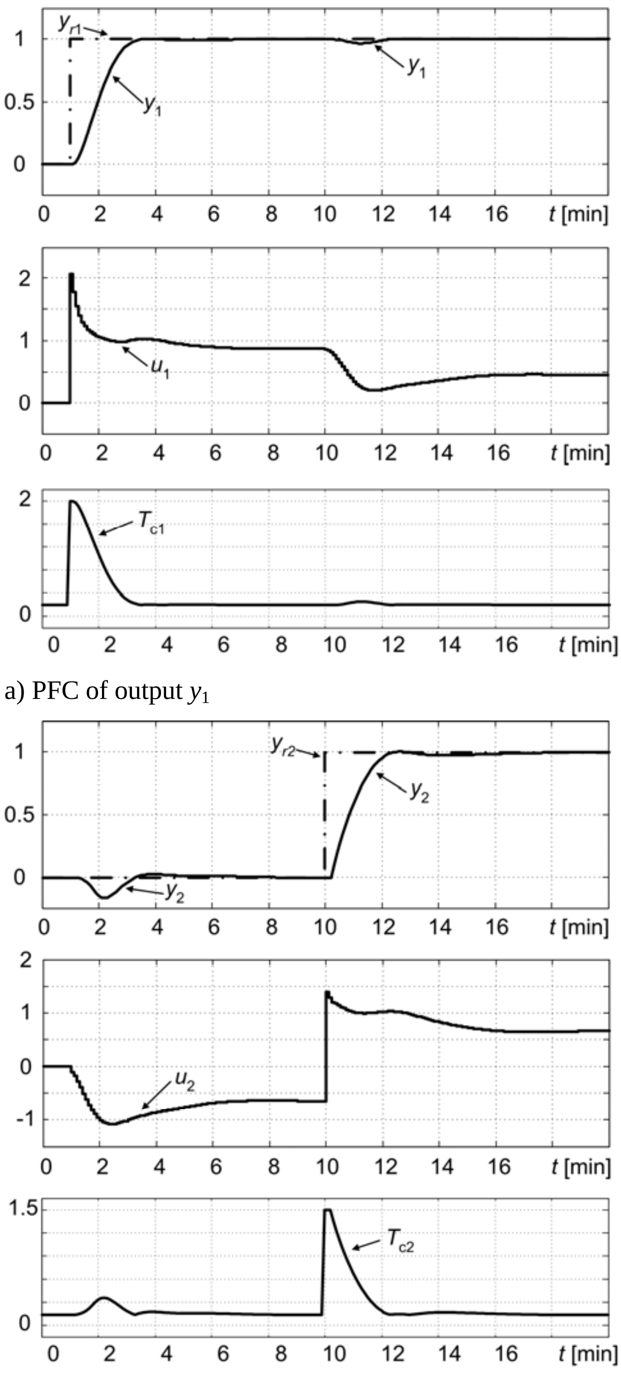
$$T_{ci} = T_{ci,\min} + (T_{ci,\max} - T_{ci,\min})|e_i(k)|$$

with $T_{c1,\max}$=2 min; $T_{c2,\max}$=1.5 min; $T_{c1,\min}$=0.2 min and $T_{c2,\min}$=0.15 min.

Fig. 9 shows that the desired settling times of those controlled variable whose reference signal was changed were temporarily significantly increased and the other settling time is remained small, this is in opposite to Fig. 7.

The maximal control error is about 3.2% (related to the changes of y_{r2}) in the controlled variable y_1, and about 16.3% (related to the changes of y_{r1}) in the controlled variable y_2.

This shows that the automatic adaptation of PFC parameters is not as good as the automatic adaptation of GPC parameters but the decoupling effect became much better in comparison with the manual adaptation in Fig. 7. And as mentioned already the realization of this control error-dependent adaptation is easier than detecting changes in the reference signals.



a) PFC of output y_1

b) PFC of output y_2

Fig. 9 TITO PFC with control error-linear dependent adaptation of T_c

6.3 Control error-exponential dependent adaptation of PFC parameters

The settling times can be set as an exponential function in the other control error; therefore the settling time of the controlled variable whose reference signal was kept constant will be smaller than the settling time of the controlled variable whose reference signal was changed, (Zabet, Haber, 2010).

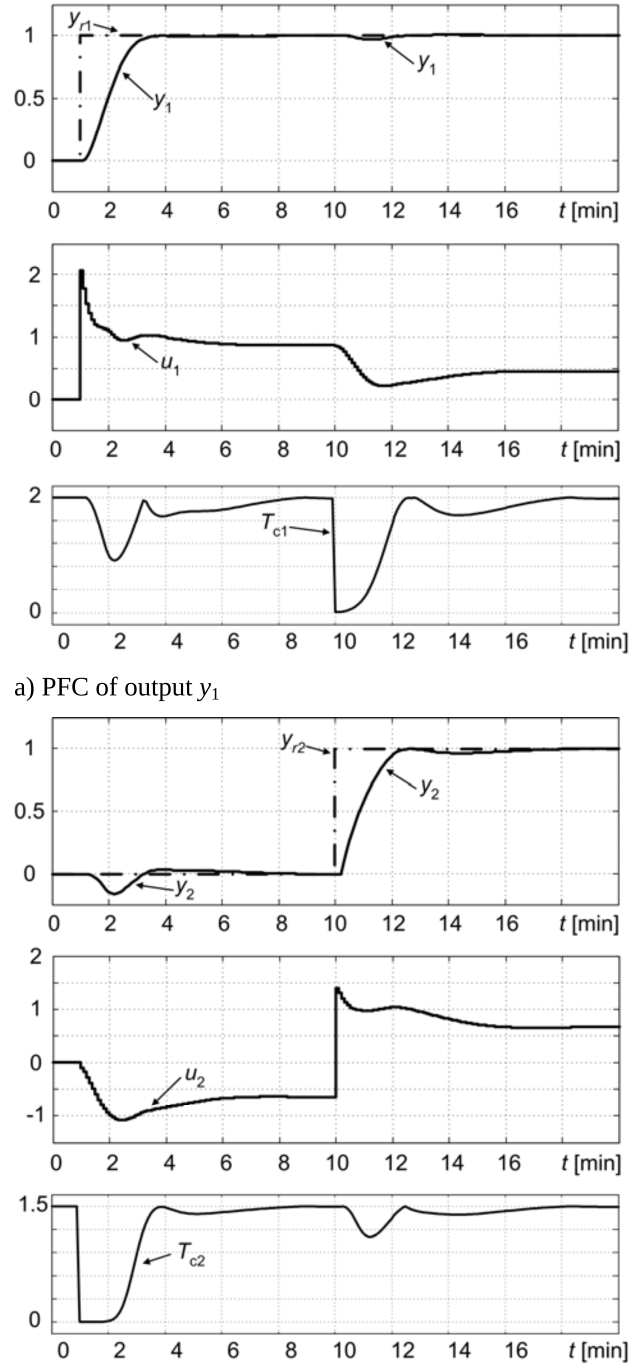
The following exponential dependence was designed:

$$T_{ci} = T_{ci,\max} \cdot \exp\left(-T_{cj,damp} \cdot |e_j(k)|\right) \quad \forall i, j = 1,2 \quad ; \quad i \neq j$$

$T_{c1,\max}$=2min; $T_{c2,\max}$=1.5min; $T_{c1,\mathrm{damp}}$=10 and $T_{c2,\mathrm{damp}}$=5.

Fig. 10 shows that the desired settling times of those controlled variables whose reference signal was kept constant were temporarily significantly reduced in the moment of the other reference signal change as in Fig. 7.

The maximal control error of y_1 is about 3.1% (related to the changes of y_{r2}), and about 16% (related to the changes of y_{r1}) in the controlled variable y_2. This shows that this automatic adaptation method is worse than with GPC controller but the decoupling effect became better in comparison with the linear dependency adaptation method (Fig. 9).



a) PFC of output y_1

b) PFC of output y_2

Fig. 10 TITO PFC with control error-exponential dependent adaptation of T_c

CONCLUSION

TITO predictive control was illustrated with two different predictive control algorithms: GPC and PFC. The controller parameters in both methods were first fixed in the simulation of the TITO control.

New simple methods were presented for reducing the decoupling effect of the TITO GPC/PFC control with proper adaptation of the controller parameters. The two methods (1) reference signal change-dependent controller parameters as an event dependent adaptation method, and (2) control error-dependent controller parameters as a signal-dependent adaptation method were designed and simulated. Both methods have shown improved decoupling effects.

With GPC algorithm the controlled variables were perfectly decoupled by both adaptation methods. The second method (control error-dependent adaptation) was prior to the first method (reference signal change-dependent adaptation).

The decoupling became better with both adaptation methods using the PFC algorithm; this fact is clarified more for a slower controlled variables. With the first method the control error was a bit smaller than without any adaptation. In the second method the linear dependency (control error-linear dependent adaptation) was better than the first method, but a bit worse than with exponential dependency (control error-exponential dependent adaptation).

The adaptation of GPC controller parameters has more affect on the decoupling feature than the adaptation of PFC controller parameters for the studied set of parameters.

Among the two controller parameter adaptation method the second one (control error dependent-adaptation) is easier to realize in practice. The presented idea can also be extended for processes with more than two controlled variables.

8. ACKNOWLEDMENTS

The first author gratefully acknowledges the scholarship of the General People's Committee for Higher Education, Great Socialist People's Libyan Arab Republic.

REFERENCES

Clarke, D. W., C. Mohtadi and P. S. Tuffs. (1987). Generalized predictive control. Part I. The basic algorithm. Automatica Vol. 23, No. 2, pp. 137–148.

Khadir, M.T., Ringwood, J.V. (2008). Extension of first order predictive functional controllers to handle higher order internal models, Int. J. Appl. Math. Comput. Sci., Vol. 18, No. 2, 229–239.

Maurath, P. R., D. E. Seborg and D. A. Mellichamp (1986). Achieving Decoupling with Predictive Controllers. Proc. Amer. Control Conf., Seattle, 1372-1377.

Richalet, J., O’Donavan, D. (2009). Elementary Predictive Functional Control. Springer Verlag, Berlin, ISBN: 978-1-84882-492-8.

Schmitz U., Haber R., Arousi F., Bars R. (2007). Decoupling predictive control by error dependent tuning of the weighting factors, AT&P Journal PLUS, 2, Bratislava, 131 – 140.

Zabet K., Haber R. (2010). Improvement of the decoupling feature of decentralized predictive functional control, ACD Conference, Ferrara.

Separating Functions for Complexity Reduction of Explicit Model Predictive Control

Ivana Rauová [*], **Michal Kvasnica** [*], and **Miroslav Fikar** [*]

[*] *Institute of Automation, Information Engineering and Mathematics,*
Slovak University of Technology,
812 37 Bratislava, Slovakia
{ivana.rauova, michal.kvasnica, miroslav.fikar}@stuba.sk

Abstract: In this work we propose to reduce memory footprint of explicit MPC controllers by eliminating a significant portion of controller's regions in which the value of the optimal control action attains saturated values. Such regions are then separated by a suitable function, which serves to recover the original control behavior. As a consequence, complexity of explicit MPC feedback laws is reduced considerably without sacrificing optimality.

Keywords: explicit model predictive control, separation, nonlinear optimization

1. INTRODUCTION

Implementing MPC in the Receding Horizon fashion (RHMPC) requires, at each sampling instance, obtaining the optimal control input by solving a suitable optimization problem. Difficulties arise when the sampling time is too short to perform the optimization on-the-fly. One way around this problem is to precompute the optimal control action u^* for all feasible initial conditions x in form of an explicit feedback law $u^* = \kappa(x)$. As shown in Bemporad et al. (2002), for a rich class of MPC problems the feedback function κ takes the form of a piecewise affine (PWA) function, which is defined over a set of polytopic regions. Computing u^* on-line then reduces to a mere function evaluation. However, the number of regions of κ, which is problem-dependent, tends to be large, easily exceeding the storage capacity of a selected implementation platform. Therefore, it is important to keep the number of regions as low as possible.

One way to reduce complexity is to construct a suboptimal replacement function $\widetilde{\kappa} \approx \kappa$ of substantially lower complexity, see e.g. Bemporad and Filippi (2003); Johansen and Grancharova (2003); Grieder et al. (2004); Cychowski and O'Mahony (2005); Jones and Morari (2009); Scibilia et al. (2009). Another line of research is concerned with finding such a replacement $\widetilde{\kappa}$ which is simpler than the original function, but maintains the equivalence $\kappa(x) = \widetilde{\kappa}(x)$ for all points x of interest, as elaborated in Baotic et al. (2008); Geyer et al. (2008); Wen et al. (2009).

In Kvasnica and Fikar (2010) we have shown how to find a simpler, equivalent feedback law $u^* = \widetilde{\kappa}(x)$ by exploiting geometric properties of explicit MPC solutions. Specifically, we have demonstrated that in majority of controller's regions the optimal control action is saturated either at the allowable maximum or minimum limits. Such regions can subsequently be eliminated and replaced by "extensions" of the regions in which the control action is unsaturated. Such a procedure leads to an equivalent controller $\widetilde{\kappa}$ which is defined over, on average, $1.3N_{\text{unsat}}$ regions, where N_{unsat} is the number of unsaturated regions. In this work we show how to construct the function $\widetilde{\kappa}$ which is *always* defined over N_{unsat} regions. This is achieved by separating the saturated regions by a suitable function, which serves to recover equivalence between $\widetilde{\kappa}$ and the original feedback κ. Two types of separating functions are considered: polynomials and piecewise affine separators encoded as binary search trees.

This paper is structured as follows. Theoretical concepts of explicit model predictive control are outlined in Section 3. The general idea of complexity reduction employing the concept of separating functions is elaborated in Section 4. Polynomial separators are then reviewed in Section 5, which also reviews various approaches to computing the separating polynomial. Construction of piecewise affine separators is reviewed in Section 6. Finally, in Section 7 we demonstrate viability of the presented approach by means of a large case study with the main focus on the states with one input variable.

2. DEFINITIONS

The interior of a set $\mathcal{R}$ is denoted by $\text{int}(\mathcal{R})$. Given a function κ, $\text{dom}(\kappa)$ denotes its domain. A set of n elements $\mathcal{R} := \{\mathcal{R}_1, \ldots, \mathcal{R}_n\}$ will be denoted as $\{\mathcal{R}_i\}_{i=1}^n$ and its cardinality by $|\mathcal{R}|$. A polytope is the bounded convex intersection of c closed affine half-spaces, i.e. $\mathcal{R} := \{x \in \mathbb{R}^{n_x} \mid Fx \leq g\}$. We call the collection of polytopes $\{\mathcal{R}_i\}_{i=1}^R$ the *partition* of a polytope $\mathcal{R}$ if $\mathcal{R} = \bigcup_{i=1}^R \mathcal{R}_i$, and $\text{int}(\mathcal{R}_i) \cap \text{int}(\mathcal{R}_j) = \emptyset$ for all $i \neq j$. Each polytope $\mathcal{R}_i$ will be referred to as the *region* of the partition. Function $\kappa : \mathcal{R} \to \mathbb{R}^{n_z}$ with $\mathcal{R} \subseteq \mathbb{R}^{n_x}$, $\mathcal{R}$ being a polytope, is called piecewise affine over polytopes if $\{\mathcal{R}_i\}_{i=1}^R$ is the partition of $\mathcal{R}$ and

$$\kappa(x) := K_i x + L_i \quad \forall x \in \mathcal{R}_i, \tag{1}$$

with $K_i \in \mathbb{R}^{n_z \times n_x}$, $L_i \in \mathbb{R}^{n_z}$, and $i = 1, \dots, R$. PWA function $\kappa(x)$ is continuous if $K_i x + L_i = K_j x + L_j$ holds $\forall x \in \mathcal{R}_i \cap \mathcal{R}_j$, $i \neq j$.

3. EXPLICIT MODEL PREDICTIVE CONTROL

We consider the class of discrete-time, stabilizable linear time-invariant systems

$$x_{k+1} = Ax_k + Bu_k, \tag{2}$$

which are subject to polytopic constraints $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ and $u \in \mathcal{U} \subset \mathbb{R}^{n_u}$. Assume the following constrained finite-time optimal control problem:

$$\min_{U_N} \sum_{k=0}^{N-1} x_{k+1}^T Q_x x_{k+1} + u_k^T Q_u u_k \tag{3a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k,\ x_k \in \mathcal{X},\ u_k \in \mathcal{U}, \tag{3b}$$

where x_k and u_k denote, respectively, state and input predictions over a finite horizon N, given the initial condition x_0. It is assumed in (3a) that $Q_x = Q_x^T \succeq 0$, $Q_u = Q_u^T \succ 0$, i.e. that (3) is a strictly convex quadratic programming (QP). The receding horizon MPC feedback then becomes $u^*(x_0) = [\mathbb{1}\ \mathbb{0}\ \cdots\ \mathbb{0}]U_N^*$, where the optimal control actions $U_N^* := [u_0^T, \dots, u_{N-1}^T]^T$ can be found by solving (3) as a QP for a given value of the initial condition x_0. For problems of a modest size (typically for $n_x < 5$), it is also possible to characterize the optimal feedback $u^*(x_0)$ explicitly as a PWA function of x_0 Bemporad et al. (2002) by solving (3) as a *parametric quadratic program* (pQP).

Theorem 3.1. (Bemporad et al. (2002)). The RHMPC feedback $u^*(x_0)$ for problem (3) is given by $u^*(x_0) = \kappa(x_0)$ where: (i) the set of feasible initial conditions $\Omega := \{x_0 \mid \exists u_0, \dots, u_{N-1} \text{ s.t. (3b) hold}\}$ is a polytope; (ii) $\kappa : \Omega \to \mathcal{U}$ is a continuous PWA function defined over R regions $\mathcal{R}_i$, $i = 1, \dots, R$; (iii) $\mathcal{R}_i$ are full-dimensional polytopes $\mathcal{R}_i = \{x \mid F_i x \leq g_i\}$; and (iv) $\{\mathcal{R}_i\}_{i=1}^R$ is a partition of Ω.

The advantage of such an explicit representation is obvious: obtaining the optimal control action for a given x_0 reduces to a mere evaluation of the function $\kappa(x_0)$, which is henceforth denoted as the *explicit RHMPC feedback law.* The crucial limitation, however, is that the number of regions tends to be large, often above the limits of typical control hardware implementation platforms. Method represented in Kvasnica et al. (2011) deals with the problem how to replace the feedback function κ by a different function $\widetilde{\kappa}$ which requires significantly less memory for its implementation in real-time arrangement and maintains the equivalence $\widetilde{\kappa}(x_0) \equiv \kappa(x_0)\ \forall x \in \Omega$.

4. COMPLEXITY REDUCTION VIA SEPARATION OF REGIONS

Denote by $\overline{\kappa}$ and $\underline{\kappa}$ the maximal and minimal values which the PWA function κ attains over its domain Ω. Since the set of admissible inputs $\mathcal{U}$ in (3) is assumed to be closed and bounded, and since all regions $\mathcal{R}_i$, $i = 1, \dots, R$ are bounded polytopes, $\overline{\kappa}$ and $\underline{\kappa}$ are always finite and can be computed by solving $2R$ linear programs of the form

$$\overline{\kappa}_i = \max\{K_i x + L_i \mid x \in \mathcal{R}_i\},\ i = 1, \dots, R, \tag{4a}$$

$$\underline{\kappa}_i = \min\{K_i x + L_i \mid x \in \mathcal{R}_i\},\ i = 1, \dots, R, \tag{4b}$$

with $\overline{\kappa} = \max\{\overline{\kappa}_1, \dots, \overline{\kappa}_R\}$, $\underline{\kappa} = \min\{\underline{\kappa}_1, \dots, \underline{\kappa}_R\}$. Then the regions of $\kappa(x)$ can be classified as follows.

(1) If $K_i = 0$ and $L_i = \overline{\kappa}$, then region $\mathcal{R}_i$ is *saturated at the maximum*,
(2) if $K_i = 0$ and $L_i = \underline{\kappa}$, then region $\mathcal{R}_i$ is *saturated at the minimum*,
(3) otherwise the i-th region is *unsaturated.*

Denote by $\mathcal{I}_{\max}$ and $\mathcal{I}_{\min}$ the index lists of regions saturated at the maximum and minimum, respectively, and by $\mathcal{I}_{\text{unsat}}$ the index list of unsaturated regions. With this classification, the RHMPC feedback $u^* = \kappa(x)$ can be written as

$$u^* = \kappa(x) = \begin{cases} K_i x + L_i & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}, \\ \overline{\kappa} & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\max}}, \\ \underline{\kappa} & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\min}}. \end{cases} \tag{5}$$

Evaluation of $\kappa(x)$ for any $x \in \Omega$ is therefore a two-stage process. First, the index r of region $\mathcal{R}_r$ which contains x needs to be identified. Then, the value of $\kappa(x)$ is either computed by $K_r x + L_r$ if $r \in \mathcal{I}_{\text{unsat}}$, or $\kappa(x) = \overline{\kappa}$ ($\kappa(x) = \underline{\kappa}$) if $r \in \mathcal{I}_{\max}$ ($r \in \mathcal{I}_{\min}$). Identification of the index r can either be done by searching through all regions $\mathcal{R}_i$, $i = 1, \dots, R$ sequentially, or by evaluating a corresponding binary search tree (Tøndel et al., 2003). In either case, the required memory storage is proportional to the total number of regions R.

If the number of saturated regions is non-zero, a simpler representation of κ can be obtained. Notice that, since the regions $\mathcal{R}_i$ are non-overlapping due to Theorem 3.1, for any $x \in \Omega$, $x \notin \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$, $\kappa(x)$ can only take two possible values: either $\kappa(x) = \overline{\kappa}$, or $\kappa(x) = \underline{\kappa}$. This fact can be exploited to derive a new PWA function $\widetilde{\kappa}(x)$ which maintains the equivalence $\widetilde{\kappa}(x) = \kappa(x)$ for all $x \in \Omega$, and requires less memory for its description compared to the memory footprint of $\kappa(x)$.

Proposition 4.1. Let a function $p : \mathbb{R}^{n_x} \to \mathbb{R}$ which satisfies $p(x) > 0$ for all $x \in \mathcal{R}_{\mathcal{I}_{\max}}$ and $p(x) < 0$ for all $x \in \mathcal{R}_{\mathcal{I}_{\min}}$ be given. Define

$$\widetilde{\kappa}(x) = \begin{cases} K_i x + L_i & \text{if } x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}, \\ \overline{\kappa} & \text{if } p(x) > 0, \\ \underline{\kappa} & \text{if } p(x) < 0. \end{cases} \tag{6}$$

Then, for all $x \in \Omega$, $\widetilde{\kappa}(x) = \kappa(x)$.

Proof. Follows directly from (5) and from the definition of p.

With a separator p at hand, $u^* = \kappa(x)$ can be evaluated by only looking at the unsaturated regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$. If $x \in \mathcal{R}_r$, $r \in \mathcal{I}_{\text{unsat}}$, then $u^* = K_r x + L_r$. Otherwise, based on the sign of $p(x)$, one either takes $u^* = \overline{\kappa}$ or $u^* = \underline{\kappa}$. The separating function p always exists. Since κ is continuous by Theorem 3.1, regions $\mathcal{R}_j$ and $\mathcal{R}_k$ cannot be adjacent for any $j \in \mathcal{I}_{\max}$, $k \in \mathcal{I}_{\min}$, and therefore they can always be separated by a (possibly discontinuous) function p.

As will be evidenced later, a typical explicit RHMPC feedback laws $u^* = \kappa(x)$ exhibits usually significantly smaller number of unsaturated regions in comparison to the number of saturated ones, i.e. $|\mathcal{I}_{\text{unsat}}| \ll |\mathcal{I}_{\max}| + |\mathcal{I}_{\min}|$. Therefore $\widetilde{\kappa}$ will require significantly less memory than κ, and will be faster to evaluate too, if p is a “simple” separator of the two sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$. Various types of separators can be considered, either continuous (e.g. linear or polynomial), or discontinuous (e.g. piecewise linear or piecewise polynomial). In this work we have opted for the polynomial type of separating functions p and the problem which we aim at solving is formally stated as follows.

Problem 4.2. Given a RHMPC feedback law $u^* = \kappa(x)$ with κ as in (5), construct the replacement feedback (6) by finding the multivariate polynomial

$$p(x) := \sum_{i_1+\cdots+i_n\leq\delta} (\alpha_{i_1,\ldots,i_\delta} x_1^{i_1} \cdots x_n^{i_n}), \tag{7}$$

of *minimum degree* $\delta_{\min}$ such that p strictly separates the sets of regions $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, i.e. $p(x) > 0\ \forall x \in \mathcal{R}_{\mathcal{I}_{\max}}$ and $p(x) < 0\ \forall x \in \mathcal{R}_{\mathcal{I}_{\min}}$.

Solving Problem 4.2 is, however, nontrivial, since the sets

$$\mathcal{R}_{\mathcal{I}_{\max}} = \{x \mid x \in \cup_i \mathcal{R}_i,\ i \in \mathcal{I}_{\max}\}, \tag{8a}$$

$$\mathcal{R}_{\mathcal{I}_{\min}} = \{x \mid x \in \cup_j \mathcal{R}_j,\ j \in \mathcal{I}_{\min}\}, \tag{8b}$$

can in general be non-convex. Even deciding whether they are convex or not is an NP-hard problem (Bemporad et al., 2001).

5. POLYNOMIAL SEPARATION

Given are the (non-convex) sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ as in (8), each of which consists of a finite number of polytopes $\mathcal{R}_k$. The knowledge of whether the sets are convex or not is not relevant here. Denote by $\mathcal{V}_k$ the vertices of $\mathcal{R}_k$ and fix some integer $\delta \geq 1$ in (7). Then the necessary condition for the existence of a polynomial p which strictly separates $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ is feasibility of the following optimization problem:

$$\epsilon^* = \max_{\epsilon,\alpha_i}\ \epsilon \tag{9a}$$

$$\text{s.t. } p(\mathcal{V}_i) \geq \epsilon, \quad \forall i \in \mathcal{I}_{\max}, \tag{9b}$$

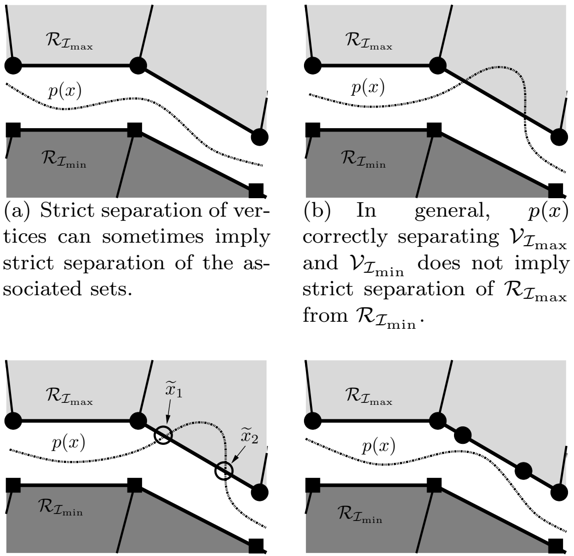
$$p(\mathcal{V}_j) \leq -\epsilon,\ \forall j \in \mathcal{I}_{\min}. \tag{9c}$$

$$\epsilon \geq 0. \tag{9d}$$

The optimal value ϵ^* then denotes the maximal separation gap between the two sets of points $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$. It is important to notice that (9) is a linear program since, for some fixed argument $x = v_k$, $v_k \in \mathcal{V}_k$, $p(x)$ in (9b)–(9c) are linear functions of the coefficients α_i. If the LP (9) is infeasible, then no strict polynomial separator p of the form of (7) exists for a given degree δ.

If $\delta = 1$ in (9) then having $\epsilon^* > 0$ is also sufficient for the linear function $p(x) := \alpha_0 + \alpha_1 x$ to strictly separate the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ (Boyd and Vandenberghe, 2004). Consider therefore $\delta > 1$. If (9) is feasible with $\epsilon^* > 0$, then one of the two possible scenarios can occur. In an ideal case, solving for coefficients of p from (9) by only considering separation of $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$ will also provide a separator for the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, as shown in Fig. 1(a). In a more general case, though, strict separation of vertices is not sufficient for $p(x)$ to separate *all points* from the associated sets.

An additional certification step therefore has to be performed. At this point we remind that all regions of $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ are polytopes described by $\mathcal{R}_i = \{x \mid F_i x \leq g_i\}$.


$\mathcal{R}_{\mathcal{I}_{\max}}$
$p(x)$
$\mathcal{R}_{\mathcal{I}_{\min}}$
$\mathcal{R}_{\mathcal{I}_{\max}}$
$p(x)$
$\mathcal{R}_{\mathcal{I}_{\min}}$
$\mathcal{R}_{\mathcal{I}_{\max}}$
$\widetilde{x}_1$
$\widetilde{x}_2$
$p(x)$
$\mathcal{R}_{\mathcal{I}_{\min}}$
$\mathcal{R}_{\mathcal{I}_{\max}}$
$p(x)$
$\mathcal{R}_{\mathcal{I}_{\min}}$

(a) Strict separation of vertices can sometimes imply strict separation of the associated sets.

(b) In general, $p(x)$ correctly separating $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$ does not imply strict separation of $\mathcal{R}_{\mathcal{I}_{\max}}$ from $\mathcal{R}_{\mathcal{I}_{\min}}$.

(c) Finding points that violate borders requires solving the problem (10).

(d) Adding the offending points to $\mathcal{V}_{\mathcal{I}_{\max}}$ and resolving (9) leads to a new separating polynomial $p(x)$.

Fig. 1. Sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$, vertices $\mathcal{V}_{\mathcal{I}_{\max}}$ (squares), vertices $\mathcal{V}_{\mathcal{I}_{\min}}$ (circles), and the polynomial separator $p(x)$.

Consider the k-th facet of $\mathcal{R}_i$, i.e. $\{x \mid f_{i,k}x - g_{i,k} = 0\}$ where $f_{i,k}$ and $g_{i,k}$ are the k-th rows of the respective matrices F_i and g_i. Denote by $\widetilde{x}_{i,k}$ all solutions to the polynomial equation $p(x) = f_{i,k}x - g_{i,k}$, restricted to $x \in \mathcal{R}_i$:

$$\widetilde{x}_{i,k} = \{x \mid p(x) - f_{i,k}x + g_{i,k} = 0,\ x \in \mathcal{R}_i\}. \tag{10}$$

Clearly, if $\widetilde{x}_{i,k} = \emptyset\ \forall i \in \mathcal{I}_{\max} \cup \mathcal{I}_{\min}$ and $\forall k$, then p as a solution to (9) strictly separates $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ (cf. Figure 1(a)). On the other hand, the situation in Figure 1(c) corresponds to the case where there exist some points $\widetilde{x}_{i,k}$ for which the polynomial $p(x)$ intersects the k-th facet of the i-th region, i.e. when $\widetilde{x}_{i,k} \neq \emptyset$ for some i and k. In such a case, the existence of *any* such point $\widetilde{x}_{i,k}$ provides a certificate that $p(x)$ does not separate $\mathcal{R}_{\mathcal{I}_{\max}}$ from $\mathcal{R}_{\mathcal{I}_{\min}}$.

When at least one offending point $\widetilde{x}_{i,k}$ exists, it can be added to the corresponding set of vertices in (9b)–(9c). I.e., if $\widetilde{x}_{i,k} \neq \emptyset$ for some $i \in \mathcal{I}_{\max}$, then $\mathcal{V}_{\mathcal{I}_{\max}} = \mathcal{V}_{\mathcal{I}_{\max}} \cup \widetilde{x}_{i,k}$. Otherwise, if $i \in \mathcal{I}_{\min}$, then $\mathcal{V}_{\mathcal{I}_{\min}} = \mathcal{V}_{\mathcal{I}_{\min}} \cup \widetilde{x}_{i,k}$. Resolving the LP (9) with the updated list of vertices will then give a new polynomial p for which the certification is repeated, cf. Figure 1(d). If more offenders are found, they are added to the list of vertices and the procedure is repeated. Otherwise, an empty solution to (10) provides a certificate that $p(x)$ strictly separates $\mathcal{R}_{\mathcal{I}_{\max}}$ from $\mathcal{R}_{\mathcal{I}_{\min}}$, whereupon the procedure terminates. The discussed mechanism can be formally stated as Algorithm 1, reported next.

Remark 5.1. Vertex enumeration in Step 1 of Algorithm 1 is considered a hard problem in general. However, for the type of small-dimensional problems considered here, enumerating $\mathcal{V}$ does not pose any significant technical difficulty and the vertices can be easily computed e.g. by CDD (Fukuda, 1997) in a matter of seconds.

Algorithm 1 Construction of a polynomial separator

```
INPUT: Sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ as in (8), polynomial degree $\delta$.
OUTPUT: Separating polynomial $p$ as in (7).
 1: Get the lists of vertices $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$.
 2: repeat
 3:     Solve the LP (9) and obtain coefficients $\alpha_0, \dots, \alpha_\delta$.
 4:     if $\epsilon^* > 0$ then
 5:         Compute the list of offending points $\widetilde{x}_{i,k}$ from (10).
 6:         Insert $\widetilde{x}_{i,k}$ to $\mathcal{V}_{\mathcal{I}_{\max}}$ or $\mathcal{V}_{\mathcal{I}_{\min}}$.
 7:     else
 8:         No strict polynomial separator of degree $\delta$ exists, abort.
 9:     end if
10: until $\widetilde{x}_{i,k} \neq \emptyset$.
```

Remark 5.2. There is no theoretical guarantee that the iterations between Steps 2–10 will terminate in finite time. However, for more than 500 random problems reported in Section 7, the number of iterations newer exceeded 10. Even more importantly, in 90% of cases Algorithm 1 terminated after a single iteration.

Solving Problem 4.2 involves finding a strict separator p of the minimum degree $\delta_{\min}$. This can be achieved e.g. by using bisection, i.e. by running Algorithm 1 multiple times for various values of δ until a feasible solution is obtained and δ is minimized.

The list of offending points $\widetilde{x}$ in Step 5 can be obtained by solving (10) in several ways, as reviewed in the sequel.

5.1 Certification via Root Finding

The list of offending points $\widetilde{x}_{i,k}$ in Step 5 of Alg. 1 can be found by interpreting (10) as a problem of finding roots of the polynomial $p(x)-f_{i,k}x-g_{i,k}$, constrained to $x \in \mathcal{R}_i$ for a particular index i. If $n_x = 1$, then the roots can be found conveniently e.g. using the `roots` command of MATLAB.

If $n_x = 2$, then the problem can be solved as follows. Consider the k-th defining hyperplane of polytope $\mathcal{R}_i$, i.e. $f_{i,k}x - g_{i,k} = 0$. Each point on this hyperplane can be expressed as

$$\{x \mid f_{i,k}^T(x - x_{0,k}) = 0\}, \tag{11}$$

where $x_{0,k}$ is any point on such hyperplane, e.g. a suitable vertex of region $\mathcal{R}_i$. This representation can in turn be expressed as

$$\{x \mid x = x_{0,k} + f_{i,k}^{\perp}\}, \tag{12}$$

where $f_{i,k}^{\perp}$ represents the orthogonal complement to $f_{i,k}$, i.e. the set of all vectors orthogonal to it

$$f_{i,k}^{\perp} = \{v \mid f_{i,k}^T v = 0\} \tag{13}$$

Substituting the orthogonal representation (12) into the definition of $p(x)$ and $f_{i,k}x + g_{i,k} = 0$ converts (10) into a problem with only one variable:

$$p(v) - f_{i,k}(x_{0,k} + f_{i,k}^{\perp}v) - g_{i,k} = 0. \tag{14}$$

Consequently, the `roots` command can be used to find all roots $\widetilde{v}_{i,k}$ of (14), from which $\widetilde{x}_{i,k}$ can be recovered from (12). The method is applied on the new polynomial to compute roots v. Note that only real roots need to be considered. Since the procedure does not limit the roots $\widetilde{x}_{i,k}$ to a particular domain ($\mathcal{R}_i$ in our case), it is necessary to obtain *all* roots of (14) and subsequently exclude those which do not belong to $\mathcal{R}_i$.

5.2 Nonlinear Programming Approach to Certification

Although the root finding procedure is easy to implement, it is limited to 1D and 2D situations, only. Another option to find out whether the list of offenders $\widetilde{x}$ is non-empty is to consider (10) as a feasibility problem with linear inequality constraints ($x \in \mathcal{R}_i$) and nonlinear equality constraint ($p(x) = f_{i,k}x - g_{i,k}$). Such an approach is applicable to arbitrary dimensions of x. Recalling that $\mathcal{R}_i = \{x \mid F_i x \le g_i\}$, the list of offenders is non-empty iff there exists a solution to the following problem:

$$\begin{aligned} &\text{find } x && (15a)\\ &\text{s.t. } p(x) = f_{i,k}x - g_{i,k}, && (15b)\\ &F_i x \le g_i, && (15c) \end{aligned}$$

which can be solved e.g. by `fmincon` of MATLAB. The practical disadvantage of such a formulation lies in the fact that equality constraints are sensitive to numerical noise.

An alternative way is to reformulate (15) as an NLP with nonlinear objective function and linear inequality constraints. Recall that a valid separator has to guarantee that $p(x) > 0$ for all $x \in \mathcal{R}_{\mathcal{I}_{\max}}$. Similarly, $p(x) < 0$ is required for all $x \in \mathcal{R}_{\mathcal{I}_{\min}}$. Let

$$\begin{aligned} f_{i,\max}^* = \min\ & p(x) && (16a)\\ \text{s.t. } & F_i x \le g_i, && (16b) \end{aligned}$$

and

$$\begin{aligned} f_{i,\min}^* = \max\ & p(x) && (17a)\\ \text{s.t. } & F_i x \le g_i. && (17b) \end{aligned}$$

The it immediately follows that if $f_{i,\max}^* < 0$ for some $i \in \mathcal{I}_{\max}$ (or if $f_{i,\min}^* > 0$ for some $i \in \mathcal{I}_{\min}$), then the point x_i^* as an optimal solution to (16) or (17) is a valid offending point which shows that $p(x)$ is *not* a strict separating polynomial.

6. SEPARATION BY BINARY TREES

An another alternative is to separate the sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ by a (possibly discontinuous) piecewise linear function p, as shown in different context by Fuchs et al. (2010). There the authors search for a separator represented as a binary search tree. Each node k of the tree represents one linear separator of the form $p_k(x) := \alpha_{k,1}x - \alpha_{k,0}$. The task then becomes to find the coefficients such that p_k correctly separates as many elements of $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ as possible. The misclassified elements are treated in a recursive fashion while building a tree. The search for p_k at each level of the tree can be cast as a mixed-integer linear program

$$\begin{aligned} \min\ & \Big|\sum R_i - \sum L_j\Big| + \sum |R_i + L_j - 1| && (18a)\\ \text{s.t. } & R_i = 1 \Leftrightarrow \{p_k(x) \ge \underline{\epsilon}\ \forall x \in \mathcal{R}_i,\ i \in \mathcal{I}_{\max}\}, && (18b)\\ & L_j = 1 \Leftrightarrow \{p_k(x) \le -\underline{\epsilon}\ \forall x \in \mathcal{R}_j,\ j \in \mathcal{I}_{\min}\}, && (18c) \end{aligned}$$

where $\underline{\epsilon} > 0$ is a given minimal separation gap introduced to avoid the trivial solution $\alpha_{k,1} = \alpha_{k,0} = 0$. Binary variables R_i (L_j) denote whether or not the corresponding region of $\mathcal{R}_{\mathcal{I}_{\max}}$ ($\mathcal{R}_{\mathcal{I}_{\min}}$) is correctly classified by a linear separator p_k, while minimizing the number of incorrectly separated regions by (18a). The crucial downside of such an approach is that a total of $|\mathcal{I}_{\max}| + |\mathcal{I}_{\min}|$ binaries needs to be introduced. If the number exceeds ~ 700 (which is considered a small case by our standards), the size of the MILP (18) becomes prohibitive to be solved even using state-of-the-art solvers, such as CPLEX.

Therefore we propose a different method of finding the linear separators $p_k(x) := \alpha_{k,1}^T x - \alpha_{k,0}$ at each level of the tree by solving a convex relaxation of (18):

$$\min_{u,v,\alpha} \ 1^T u + 1^T v \tag{19a}$$

$$\text{s.t.} \ \alpha_{k,1}^T x_i + \alpha_{k,0} \geq 1 - u_i, \quad i = 1, \ldots, \mathcal{V}_{max}, \tag{19b}$$

$$\alpha_{k,1}^T y_i + \alpha_{k,0} \leq -(1 - v_i), \ i = 1, \ldots, \mathcal{V}_{min}, \tag{19c}$$

$$u \succeq 0, \quad v \succeq 0, \tag{19d}$$

where x_i (y_i) represent extremal vertices of regions saturated at maximum $\mathcal{R}_{max}$ (regions saturated at minimum $\mathcal{R}_{min}$), and u_i and v_i are nonnegative *support vector classifiers* of x_i and y_i, respectively.

When $u = v = 1$ in (19b) and (19c), we recover original constraints $\alpha_{k,1}^T x_i - \alpha_{k,0} \geq 0$ and $\alpha_{k,1}^T y_i - \alpha_{k,0} \leq 0$ as a feasible nonstrict linear discriminant of two sets of points. Think of u_i as a measure how much the constraint $\alpha_{k,1}^T x_i - \alpha_{k,0} \geq 0$ is violated and the same holds for v_i. The goal is to find $\alpha_{k,1}$, $\alpha_{k,0}$ and sparse u and v that satisfy inequalities (19b) – (19c), maximize the slab, and minimize the number of misclassified points. In other words, (19) is a relaxation of the number of points misclassified by the function $\alpha_{k,1}^T z - \alpha_{k,0}$, plus the number of points that are correctly classified but lie in the slab of width $\{z \mid -1 \leq \alpha_{k,1}^T z - \alpha_{k,0} \leq 1\}$ given by $2/\|\alpha_{k,1}\|_2$ (Boyd and Vandenberghe (2004)).

The Algorithm 2 shows a pseudocode of the recursive function used to build a binary tree. Inputs to the function are 2 sets of saturated regions ($\mathcal{R}_{min}, \mathcal{R}_{max}$) which are immediately transformed into 2 sets of extremal vertices ($\mathcal{V}_{min}, \mathcal{V}_{max}$). Acquired PWA function splits points into correctly and incorrectly classified ones, where regions corresponding to the misclassified points are searched. If the sets are empty, algorithm reached a *leaf nodes*, otherwise the solution is referred as *node* and function is recalled with new sets of regions. Steps of the Algorithm 2 are depicted in Fig. 2.

7. EXAMPLES

7.1 Illustrative example

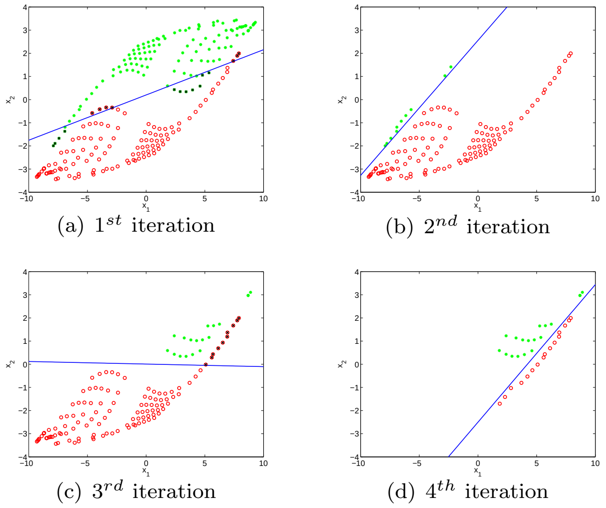
Consider a 2-state 1-input system given by

$$x^+ = \begin{bmatrix} 0.755 & 0.680 \\ 0.651 & -0.902 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.825 \\ -0.139 \end{bmatrix} u, \tag{20}$$

which is subject to constraints $\mathcal{X} = \{[\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}] \mid -10 \leq [\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}] \leq 10\}$ and $\mathcal{U} = \{u \in \mathbb{R} \mid -1 \leq u \leq 1\}$. The MPC problem (3) was formulated with prediction horizon $N = 10$, $Q_x = \mathbb{1}$ and $Q_u = 1$ and solved as a parametric QP according to Theorem 3.1. Using the MPT Toolbox (Kvasnica et al., 2004), the explicit RHMPC feedback $u^* = \kappa(x)$ was obtained in 4 seconds[1] as a PWA function defined over 225 regions. The domain of κ consists of 29 unsaturated regions, 98 regions where $\kappa(x) = 1$, and 98 regions where $\kappa(x)$ is saturated at -1.

Algorithm 2 Construction of a PWA form of separator $p(x)$

```
INPUT: Sets $\mathcal{R}_{\mathcal{I}_{\max}}$ and $\mathcal{R}_{\mathcal{I}_{\min}}$ as in (8).
OUTPUT: Separator $p$ encoded as a set of linear functions.
function BINARYTREE($\mathcal{R}_{\mathcal{I}_{\min}}, \mathcal{R}_{\mathcal{I}_{\max}}$)
Get the lists of vertices $\mathcal{V}_{min}$ and $\mathcal{V}_{max}$.
Solve the LP (19) and create a new node in the tree
    defined by $p_k(x) := \alpha_{k,1}^T x - \alpha_{k,0}$.
Find misclassified points $\mathcal{V}_{min}$, $\mathcal{V}_{max}$ and keep corresponding regions $\mathcal{R}_{min}$, $\mathcal{R}_{max}$.
if $\mathcal{R}_{min} \neq \emptyset$ then
    return BINARYTREE($\mathcal{R}_{min}$, $\mathcal{R}_{\mathcal{I}_{\max}}$).
end if
if $\mathcal{R}_{max} \neq \emptyset$ then
    return BINARYTREE($\mathcal{R}_{\mathcal{I}_{\min}}$, $\mathcal{R}_{max}$).
end if
end function
```



(a) 1^{st} iteration

(b) 2^{nd} iteration

(c) 3^{rd} iteration

(d) 4^{th} iteration

Fig. 2. Illustration of finding a PWA separator by Algorithm 2. At each iteration a new node is created which contains a linear separator p_k correctly separating as many points as possible. Shown are the vertices $\mathcal{V}_{max}$ (green stars), vertices $\mathcal{V}_{min}$ (red circles), misclassified points (black x-marks), and the linear separator $p_k(x)$ (continuous line) at each iteration.

As can be clearly seen from Figure 2(a), no linear separation between the sets of points $\mathcal{V}_{\mathcal{I}_{\max}}$ and $\mathcal{V}_{\mathcal{I}_{\min}}$ could be found. A polynomial separator $p(x) = -x_1 - x_2 + 0.6103 x_1 x_2^2 - 0.2076 x_1^2 x_2 + 0.0458 x_1^3 + x_2^3$ of the minimal degree $\delta_{\min} = 3$ was then found by applying bisection in conjunction with Algorithm 1. The algorithm converged within of one iteration. The vertices in Step 1 were com-

[1] On a 2.4 GHz CPU with 4GB of RAM using MATLAB 7.9 and MPT 2.6.3

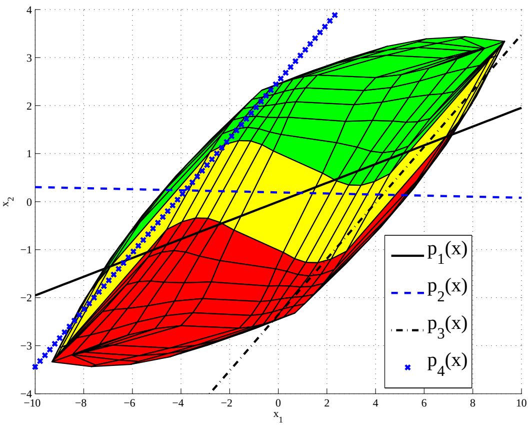


Fig. 3. Final form of a binary tree division. Linear separators $p_k(x)$ represents each of the nodes from binary tree.

puted by CDD in 0.01 seconds. Coefficients of the polynomial were obtained by solving the LP (9), which took 0.7 seconds using CPLEX. Implementing the certification in Step 5 using root finding (cf. Section 5.1) took 1.85 seconds, while the NLP-based certification of Section 5.2 took 6.55 seconds.

A binary separation tree can also be constructed by recursively solving LP problems (19). For the sets of points depicted in Figure 2, the procedure generated a tree consisting of four nodes:

$$p_1(x) = -0.38x_1 + 1.96x_2 \tag{21}$$
$$p_2(x) = 0.019x_1 + 1.75x_2 - 0.34 \tag{22}$$
$$p_3(x) = -2.26x_1 + 3.87x_2 + 9.12 \tag{23}$$
$$p_4(x) = -0.22x_1 + 0.37x_2 - 1.0 \tag{24}$$

The tree is rooted at p_1, with p_2 visited if $p_1(x) < 0$ and p_3 if $p_1(x) > 0$. The next node is rooted at p_2, with p_4 visited if $p_2(x) > 0$ which betrays the unbalanced binary tree. The total runtime of LPs (19) was 4.7 seconds with CDD.

However, as can be seen from Fig. 3, a binary tree with just 3 nodes would be sufficient to correctly separate the corresponding vertices. However, Algorithm 2 finds 4 nodes. This is due to the fact that Alg. 2 attempts to find, at each iteration, a best subdivision of the points by solving the LP (19). No effort is made to minimize the cardinality or the depth of the resulting PWA tree. This can, however, be achieved by an adequate post-processing procedure.

The total memory footprint of κ (which consists of the regions $\mathcal{R}_i$ and the feedback laws $K_i x + L_i$) with 225 regions, is 27 kilobytes. On the other hand, by devising a separator p, the storage requirements of $\widetilde{\kappa}$ drops to a mere 3.5 kilobytes. Here, the unsaturated regions $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ contribute by 2.8 kB, the associated feedback laws by 0.7 kB, and the memory footprint is just 24 bytes for the polynomial separator, and 48 bytes for the binary tree. It follows that complexity of the on-line implementation of the RHMPC feedback law can be reduced by a factor of 7.7 by using the modified feedback $u^* = \widetilde{\kappa}(x)$ instead of the original function $u^* = \kappa(x)$.

Table 1. Comparison of total runtimes for different routines in polynomial separation ($\delta_{min} = 3$), selected problems.

No. of regions	Runtime [sec]	
	`fmincon`	`roots`
189	6.15	2.73
199	5.69	2.85
225	7.82	2.96
257	7.61	3.46
495	14.23	6.36

Table 2. Number of nodes and runtimes necessary for rooting, comparison of LP (19) with MILP algorithm of Fuchs et al. (2010)

No. of regions			LP		MILP	
total	sat	unsat	No. of nodes	total runtime [sec]	No. of nodes	total runtime [sec]
189	172	17	4	4.2	3	8.6
199	154	45	4	6.7	3	7.0
225	196	29	4	6.7	3	9.1
257	208	49	4	5.0	4	17.3
283	244	39	5	6.1	4	23.7
319	260	59	4	5.6	3	12.5
495	460	35	5	13.7	48	1176.9

7.2 Random examples

Next, we have analyzed a large number of random RHMPC feedback laws κ generated by solving problem (3). We have considered 100 random problems with 2 states and 1 input. For each PWA function we have constructed the replacement $\widetilde{\kappa}$ as in (6). Both polynomial and binary tree separation were considered.

The purpose of this study was twofold. First, we have investigated how the root-finding approach to certification compares to the more general NLP-based procedure of Section 5.2. Table 1 shows that the root finding procedure performs twice as fast as the NLP approach. Moreover, it illustrates how the computation scales with increasing complexity of the problem. It should be noted, though, that the NLP procedure is applicable to arbitrary dimensions, while the root finding approach is limited to 2D scenarios only.

Next, we compared the runtime needed to construct a PWA separator, encoded as a binary search tree, using the MILP procedure of Fuchs et al. (2010) and by the convex LP relaxation (19). The results summarized in Table 2 show that the LP relaxation is significantly more efficient, for it being able to construct the PWA separator even for large scenarios. Note also that the number of unsaturated regions (denoted by *unsat*) is significantly smaller in comparison to the number of saturated regions (*sat*).

8. CONCLUSIONS

Given an explicit RHMPC feedback function κ, we have shown how to construct its simpler replacement $\widetilde{\kappa}$ which maintains the equivalence $\kappa(x) = \widetilde{\kappa}(x)$ for all $x \in \operatorname{dom}\kappa(x)$. The mechanism was based on devising a function $p(x)$, which separates regions over which κ attains a saturated value. The replacement $\widetilde{\kappa}$ then requires only the storage of the unsaturated regions of κ, along with the separator p. We have shown how to build such a p by solving a linear optimization problem, followed by a certification step which requires solution to a polynomial equation. Two approaches to such a certification were proposed: one based on finding roots of a polynomial, and second one based on solving a nonlinear programming problem. By means of a case study we have illustrated how different approach to certification influence the total computation time. When a piecewise affine separator is desired, we have proposed to use a convex relaxation of the separation problem, which is significantly more efficient compared to the approach based on solving a mixed integer separation problem.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grant 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contracts No. VV-0029-07 and No. LPP-0092-07. This work was also financed by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Baotic, M., Borrelli, F., Bemporad, A., and Morari, M. (2008). Efficient On-Line Computation of Constrained Optimal Control. *SIAM Journal on Control and Optimization*, 47(5), 2470–2489.

Bemporad, A. and Filippi, C. (2003). Suboptimal explicit RHC via approximate multiparametric quadratic programming. *Journal of Optimization Theory and Applications*, 117(1), 9–38.

Bemporad, A., Fukuda, K., and Torrisi, F.D. (2001). Convexity Recognition of the Union of Polyhedra. *Computational Geometry*, 18, 141–154.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Cychowski, M. and O'Mahony, T. (2005). Efficient offline solutions to robust model predictive control using orthogonal partitioning. In *Proceedings of the 16th IFAC world congress*.

Fuchs, A., Jones, C., and Morari, M. (2010). Optimized Decision Trees for Point Location in Polytopic Data Sets - Application to Explicit MPC. In *American Control Conference*. Baltimore, USA.

Fukuda, K. (1997). *Cdd/cdd+ Reference Manual*. Available from www.cs.mcgill.ca/~fukuda/soft/cdd_home/cdd.html.

Geyer, T., Torrisi, F., and Morari, M. (2008). Optimal complexity reduction of polyhedral piecewise affine systems. *Automatica*, 44(7), 1728–1740.

Grieder, P., Wan, Z., Kothare, M., and Morari, M. (2004). Two level model predictive control for the maximum control invariant set. In *American Control Conference*. Boston, Massachusetts.

Johansen, T. and Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Trans. on Automatic Control*, 48, 810–815.

Jones, C. and Morari, M. (2009). Approximate Explicit MPC using Bilevel Optimization. In *European Control Conference*. Budapest, Hungary.

Kvasnica, M. and Fikar, M. (2010). Performance-lossless complexity reduction in explicit MPC. In *Conference on Decision and Control, CDC*, (accepted). Atlanta, USA.

Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT). Available from http://control.ee.ethz.ch/~mpt/.

Kvasnica, M., Rauová, I., and Fikar, M. (2011). Simplification of Explicit MPC Feedback Laws via Separation Functions. In *IFAC World Congress*. Milano, Italy.

Scibilia, F., Olaru, S., and Hovd, M. (2009). Approximate explicit linear MPC via delaunay tessellation. In *Proceedings of the 10th European Control Conference*. Budapest, Hungary.

Tøndel, P., Johansen, T.A., and Bemporad, A. (2003). Evaluation of Piecewise Affine Control via Binary Search Tree. *Automatica*, 39(5), 945–950.

Wen, C., Ma, X., and Ydstie, B.E. (2009). Analytical expression of explicit MPC solution via lattice piecewise-affine function. *Automatica*, 45(4), 910 – 917. doi: DOI:10.1016/j.automatica.2008.11.023.

Lifetime Estimation of Heat Exchangers with Consideration of On-line Cleaning

T. Friebel* R. Haber* U. Schmitz**

* *Cologne University of Applied Science*
Institute of Process Engineering and Plant Design, Laboratory for Process Control,
Köln, Germany {thomas.friebel, robert.haber}@FH-Koeln.de

** *Shell Deutschland Oil GmbH*
Rheinland Raffinerie Godorf, Germany
ulrich.schmitz@shell.com

Abstract: In the presented paper two quality parameters are used to represent the state of a heat exchanger. The remaining lifetime can be estimated by trend regression. Also of interest is the uncertainty of the predicted lifetime which is determined by the confidence interval of the parameter estimation. These algorithms developed are used in this paper in an off-line evaluation of the measurements on a heat exchanger in a refinery. It is shown that the time point of the heat exchanger cleaning can be predicted. So the presented method can be used for planning the cleaning time point in advance and saving money in maintenance.

1. INTRODUCTION

In practice there is always a risk that the pipes of a heat exchanger become clogged with solid particles due to strong temperature differences. During the operation it is not possible to look into the interior of the heat exchanger. So the state of heat exchanger has to be monitored based on measurable quantities. Such measurements are temperature, flow and pressure drop. These measures allow not only the description of the current state but the planning of maintenance in due time. Most methods, which are known in the literature, are based on models. The models can be separated into two groups. In the first case, the models are based on multivariate regression, PCA, neural networks and so on. An advantage of all these methods is that they can be used without detail knowledge about the inner states and chemical reactions in the heat exchanger. All these models commonly need fault free training data to generate e.g. the regression model. Also there are disadvantages. If the heat exchanger leaves the normal working point, probably a neural network becomes bad, or a fault with no effect on the used principal components will be not detected, because not all possible conditions can be realized with a real plant. In the other case physical models based on inlet and outlet are used. There are several methods, which try to observe the inner state of the heat exchanger. Using complex physical models can cause problems with the generalization far from the working point. However, if there is only one question: “Is the heat exchanger in a normal condition?” a simple model can be used. In this paper, two quality parameters both based on an easy physical model are compared. The first one is ε the degree of efficiency and the second one is UAF a combination of the heat transmission coefficient with the inner surface and the flow. Big advantages of these methods are that they can be used without any training data and the observation can start from any state of the heat exchanger. A great and important target of the conditioning monitoring is to predict the time interval until a detected disturbance reaches a tolerance level and becomes a fault. The dwell time of the fluids in the heat exchanger is very small against a normal observation period. In addition, if there are complex chemical and physical reactions it is probably not possible to predict the future with an exact model. As explained before the two quality parameters ε and UAF are observed and the remaining lifetime is estimated by trend regression. The quality of the regression can be observed by using statistical tests like a *t*-test. Also of interest is the uncertainty of the predicted lifetime which is determined by the confidence interval of the parameter estimation. The presented method is used with real measurements from an oil refinery. The target is to predict the time points of the cleaning on-line. Thereby the measurements are evaluated in on-line mode. Also it is shown that the quality parameter UAF allows a better prediction than with the classical degree of efficiency ε .

2. METHODS AND THEORY

In the following a counter current heat exchanger will be dealt with. In the actual application the inlet and the outlet flow are coupled on thermal side, see Fig. 1. The cold reactant with temperature T_{1E} enters the heat exchanger on the cold side. It is preheated by the product flow and it leaves the heat exchanger with temperature T_{2E} on the hot side. The hot product enters the heat exchange with temperature T_{2P} on the hot side. The fluent is cooled down by the reactants and it leaves the heat exchanger with temperature T_{1P} on the cold side. In the following chapters both quality parameters for the monitoring of the heat exchanger are presented. After that the method for rest live time prediction is shown.

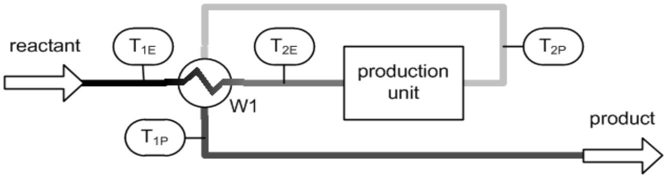


Fig. 1. Simplified flow diagram of the plant

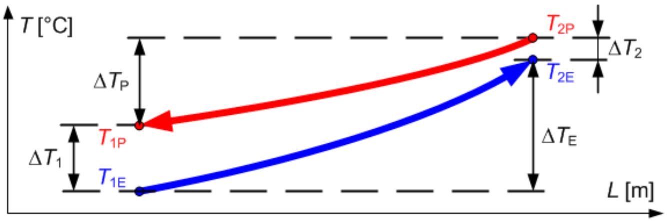


Fig. 2. Temperature profile of a current flow heat exchanger with tube length L

2.1 *Quality parameter – degree of efficiency*

Fig. 2 shows a typical temperature profile over a counter current heat exchanger. The state of the heat exchanger can be described by the ratio of the actually transferred energy and the maximum transferable energy. The actual amount of transferred energy is proportional to the temperature difference between inlet and outlet of the reactant, $\Delta T_E = T_{2E} - T_{1E}$. The maximum transferable energy is proportional to the temperature difference between the fluid inlet temperatures, $T_{2P} - T_{1E}$. For this definition the assumptions $\Delta T_E > \Delta T_P$ and $\Delta T_1 > \Delta T_2$ are used. The ratio of actual and maximal energy transfer is given by (1), where ε is the degree of efficiency see Wagner (2005).

$$\varepsilon = \frac{T_{2E} - T_{1E}}{T_{2P} - T_{1E}} \tag{1}$$

The efficiency depends on the set point and the inner state of a heat exchanger. The set point is defined by the inlet temperatures and the rate of fluid flow. The inner state depends on the fouling. Therefore in the present paper nearly steady-state conditions are assumed for the inlet temperatures and the amount of fluid flow. In this case a change in the degree of efficiency is caused by fouling.

2.2 *Quality parameter – heat transfer coefficient*

As discussed above the quality parameter (degree of efficiency) depends on the set point and the inner state of the heat exchanger. Therefore an additional formulation will be used. The model equation for heat transfer can be written as (2) see Wagner (2005).

$$Q = U \cdot A \cdot \Delta T_{\log} \tag{2}$$

In (2) Q stands for the heat flow, U for the coefficient of heat transmission, A for the surface of the heat exchanger and $\Delta T_{\log}$ for the logarithmic mean temperature difference. The logarithmic mean temperature difference is defined in (3). Increasing fouling leads to a decreasing coefficient of heat transfer due to additional heat resistance; see (4).

$$\Delta T_{\log} = \frac{\Delta T_1 - \Delta T_2}{\ln\left(\dfrac{\Delta T_1}{\Delta T_2}\right)} \tag{3}$$

$$U = \frac{1}{\dfrac{1}{\alpha_i} + \dfrac{s_{wall}}{\lambda} + \dfrac{s_{dep}}{\lambda_{dep}} + \dfrac{1}{\alpha_a}} \tag{4}$$

In (4) α denotes the heat transfer coefficient, s the coat thickness and λ the heat conductance coefficient. The index i stands for the inner and a for the outer side of the pipe and dep denotes biomass coat or solid deposition. The heat flow Q can be calculated from the measured process parameters by (5) see Friebel et al. (2009) and Wagner (2005).

$$Q = F_E \cdot c_{pE} \cdot \Delta T_E \tag{5}$$

In (5) F_E stands for mass flow of the reactant, c_{pE} for the heat capacity of the reactant and ΔT_E for the temperature difference from outlet to inlet of the reactant, see Fig. 2. By combining (2) and (5) the quality parameter UA can be defined as shown in (6) see Friebel et al. (2009).

$$UA = F_E \cdot c_{pE} \cdot \frac{\Delta T_E}{\Delta T_{\log}} \tag{6}$$

Simulations presented in Friebel et al. (2009) show that the quality parameter UA is sensitive for fouling. This is a big advantage against the degree of efficiency because fouling can be distinguished from model input drift. This means the degree of efficiency is sensitive for

- a drift in one or both of the inlet temperature,
- a drift in one or both of the fluid flows and
- a drift in the model parameter UA i.e. in the heat transfer.

By analyzing the above listing it is clear that it is nearly impossible to differ among all possible combinations of drifts. Therefore, it is more practical to estimate the model parameter UA at every steady-state sampling point k according to (6). In the special case where the flow of the reactant F_E cannot be measured the flow and c_{pE} are assumed constant. Then (6) can be transformed to (7)

$$UAF = \frac{\Delta T_E}{\Delta T_{\log}} \tag{7}$$

It is important to see that the parameter UAF is sensitive to fouling and disturbances in the flow.

2.3 Extrapolation of a regression model

Above a proper quality parameter for the heat exchanger is defined. The last question is: how long is the rest lifetime until a critical level is reached. Therefore a linear discrete-time regression model is used, see (8). Here u_k is the independent variable and y_k the dependent variable, see Montgomery et al. (2001). In the following application the actual measurement is the best representation of the inner state of the heat exchanger. Therefore the point of origin is equal to the actual measurement. The regression model contains only the unknown parameter $\hat{c}_1$.

$$y_k = c_1 u_k = \hat{y}_k + e_k = \hat{c}_1 u_k + e_k \tag{8}$$

With a simple extrapolation the rest lifetime u_{pred} can be calculated, while the regression model $\hat{y}_{\text{pred}}$ is equal to the tolerance limit y_{tol}. As it was shown in Friebel et al. (2009) the quality of the regression can be proven by a statistical *t*-test. The uncertainty of a regression can be shown by his confidential intervals. With the assumption, that the uncertainty at the actual measurement is equal to zero the confidential limits for future measurements can be calculated by (9) see Montgomery et al. (2001)

$$\hat{y}_k \pm s_{\text{reg}} = \hat{y}_k \pm u_k \cdot t_{\frac{\alpha}{2};N-1} \cdot \sqrt{\frac{\sigma^2}{\sum_{k=0}^{N-1} u_k^2}} \tag{9}$$

Here σ is the residual error, N the amount of used data with the regression and t the value of a *t*-distribution for a given significance level α and $N-1$ degrees of freedom. By setting the left term in (9) equal to y_{tol} and solving the equation the predicted rest lifetime $u_{\text{k}} = u_{\text{pred}}$ can be calculated with its uncertainty s_{reg}.

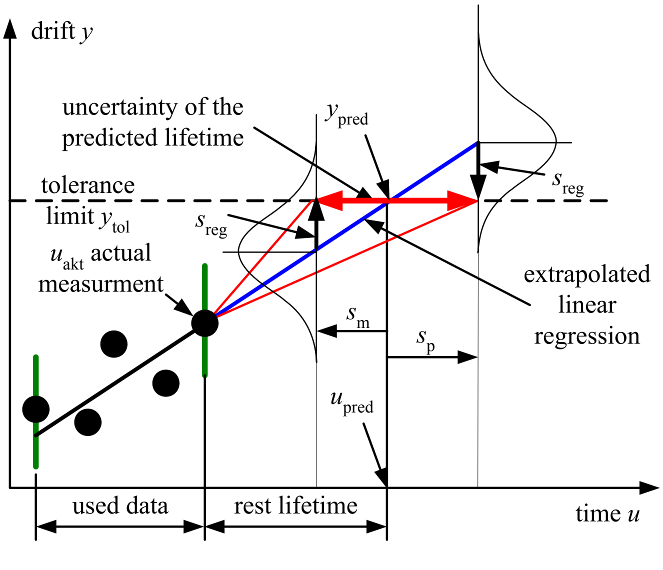

drift y
uncertainty of the predicted lifetime
y_{pred}
tolerance limit y_{tol}
s_{reg}
s_{reg}
u_{akt} actual measurment
extrapolated linear regression
s_{m}
s_{p}
u_{pred}
used data
rest lifetime
time u

Fig. 3. Estimation of the rest lifetime

The principle of the calculation is shown in Fig. 3. The calculated uncertainties in time s_{m} and s_{p} have different values. For an easier interpretation in the practical use a middle uncertainty $\bar{s}$ is defined in (10)

$$\bar{s} = \frac{s_{\text{m}} + s_{\text{p}}}{2} \quad \text{with} \quad s_{\text{m}} < s_{\text{p}} \tag{10}$$

3. APPLICATION

3.1 Problem description

As it is seen in Fig. 1 the reactant is preheated by the product and the product is cooled by the reactant. In Fig. 4a the tubes of a cleaned heat exchanger are shown. In Fig. 4b the problem with fouling, adhesion on the surface inside and outside the tubes is shown. On the right side in Fig. 4c the totally blocked tubes can be seen. Blocked tubes cause the following problems:

- pressure drop over the heat exchanger increase
- the maximal cooling power of the heat exchanger decreases
- product have to be cooled additional before entering the storage
- reactant have to be heated additional before entering the production unit

Fouling costs some money. Normally a heat exchanger is observed by the degree of efficiency ε. To prevent the above listed problems the quality parameters UA and UAF are used.

3.2 Problem solution

Typical measurements (temperature) are shown exemplary in Fig. 5. In this analysis, several years are taken into account, but only some examples are shown in this paper. Therefore, the discrete time *k* does not start at one. The period starts and ends with a cleaning of the heat exchanger, all temperatures are low. The cleaning was performed if the degree of efficiency reached a value of e.g. 90%. Now the target is to predict these cleaning time point in order to plan a cleaning in advance. The temperatures of the production process are in a range between 350 and 450 °C see Friebel et al. (2010). The reactant is drawn with solid lines and the product in dashed lines.

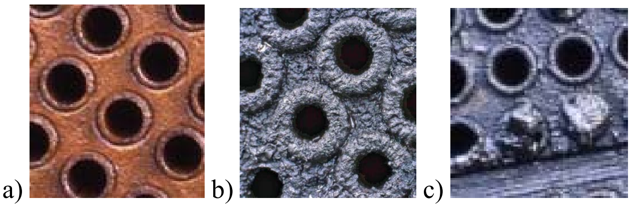


Fig. 4. Problems with fouling in a tube bundle heat exchanger a) clean surface and tubes, b) surface with fouling and c) blocked tubes after fouling see Friebel et al. (2010)

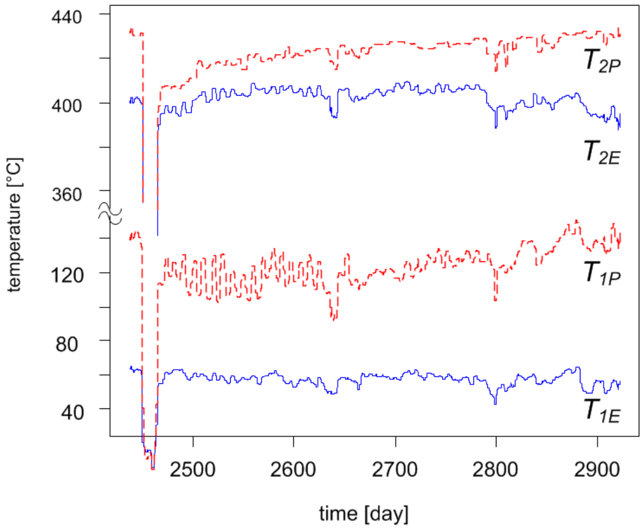


Fig. 5. Temperature measurements of the heat exchanger for a time period between the plant revisions

It can be seen, that the product temperatures (dashed lines) increase over the time. Also the temperatures are noisy especially the product outlet T_{1P} and the reactant outlet T_{3E}. The reason is a periodical sinusoidal disturbance with a period of approximately 11 days, which is caused by the plant management. The problem is that the amplitude is not constant and also there are some stepwise phase shifts in the periodical signal. In Fig. 6 the degree of efficiency ε and the model parameter UAF calculated by (1) and (7) are shown for an interesting part of this time period.

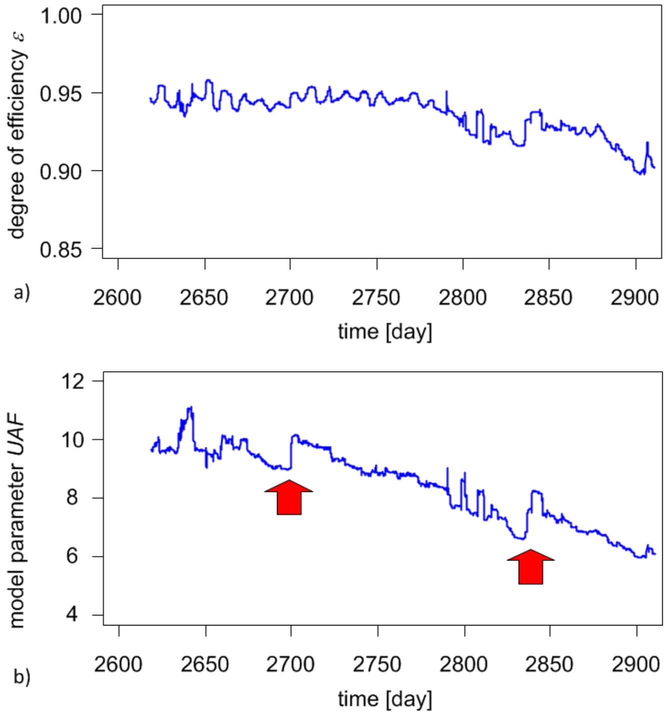


Fig. 6. Example for calculated a) degree of efficiency ε and b) model parameter UAF

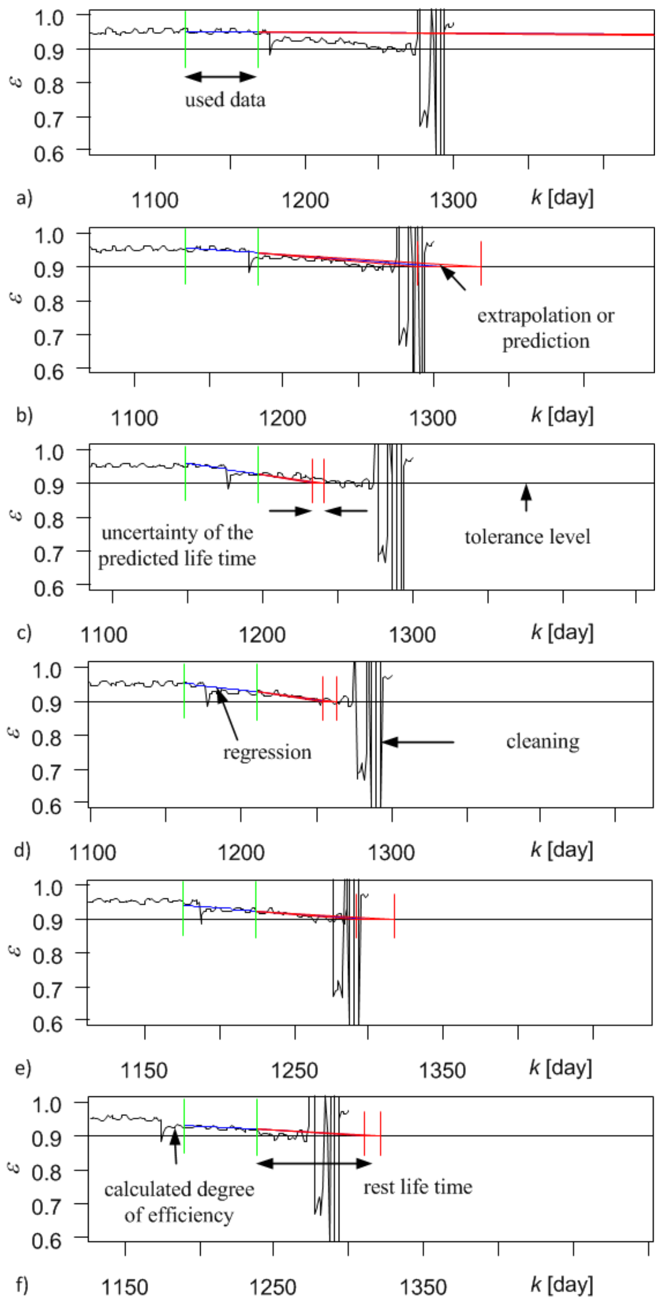


Fig. 7. Regression based lifetime estimation a) 19, b) 17, c) 15, d) 13 e) 11 and f) 9 weeks before shutdown.

It can be seen, that the calculated degree of efficiency ε is very noisy. The sinusoidal disturbance is clearly visible. But by observing the model parameter UAF these disturbances are eliminated. In the model parameter some additional information can be detected. There are two steps which are marked with arrows. These steps are caused by not recorded technological handlings. It is not possible to detect both steps in the degree of efficiency. Because this signal is caused by periodical disturbances and the parameter is not so sensitive for this case. Therefore it is a good idea to use the model parameter UAF instead the degree of efficiency ε.

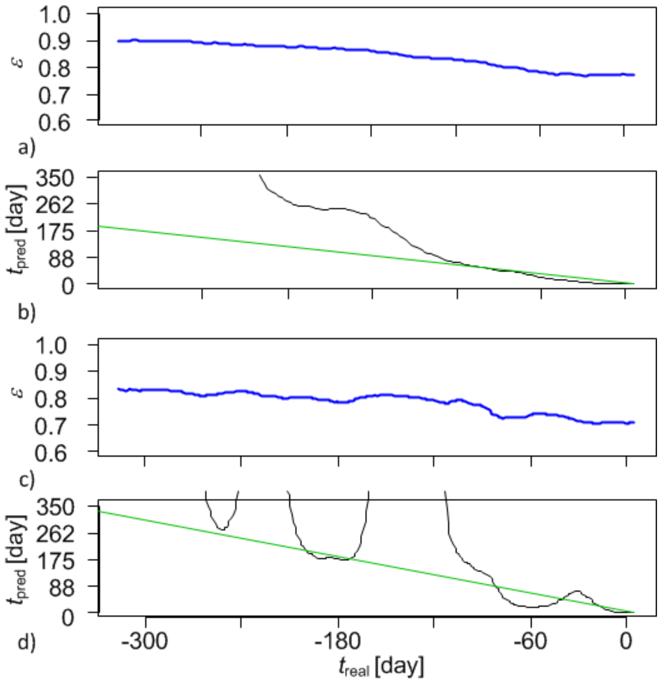


Fig. 8. Two observation periods; a) and c) degree of efficiency (solid line) and filtered signal (dashed line), b) and d) predicted rest lifetime t_{pred} versus real time t_{real} until shutdown

In Fig. 7 the calculated efficiency for another example is presented. The regression was carried out with 50 measurements in the on-line mode. Fig. 7a shows the regression 19 weeks before the shutdown. The following figures show in turn the regression always two weeks later. The rest lifetime is predicted with a small uncertainty, because the interval limits on the tolerance level are close together. As a conclusion of Fig. 7 the following points can be marked out, see also Friebel et al. (2010).

- The rest lifetime t_{pred} could be predicted nearly exact several weeks before the shutdown.
- With a *t*-test it could be shown, that the used regression model is always significant.
- The corresponding significance values α are nearly 0 %.
- The confidence limits lie near to the predicted rest lifetime t_{pred}.

In Fig. 8 two additional periods are analysed. The time until the next realized shutdown t_{real} is shown on the horizontal axis. In Fig. 8a the filtered and calculated heat exchanger efficiency is shown for a further period between two cleaning cycles. It can be seen, that the signal is not very noisy. In Fig. 8b the predicted rest lifetime t_{pred} is plotted on the vertical axis. In the ideal case the times t_{pred} and t_{real} are equal, which is marked by the diagonal line. Two month before the shutdown an acceptable prediction is possible. In the second period in Fig. 8c it can be seen that calculated degree of efficiency temporally increases because of a not recorded technological handling. This has a direct impact on the prediction of rest lifetime in Fig. 8d. The trend is clearly visible along the diagonal line. Because of the noisy signal and the technological handling there are changes in sign of the slope.

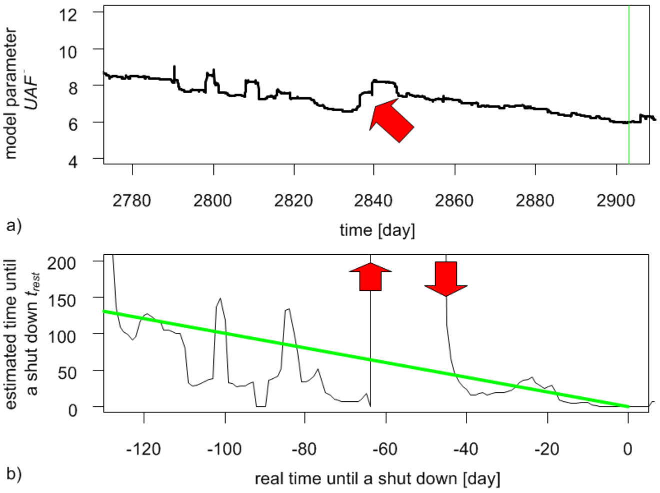


Fig. 9. Model parameter UAF with a step caused by online cleaning in a) and the corresponding rest lifetime estimation in b)

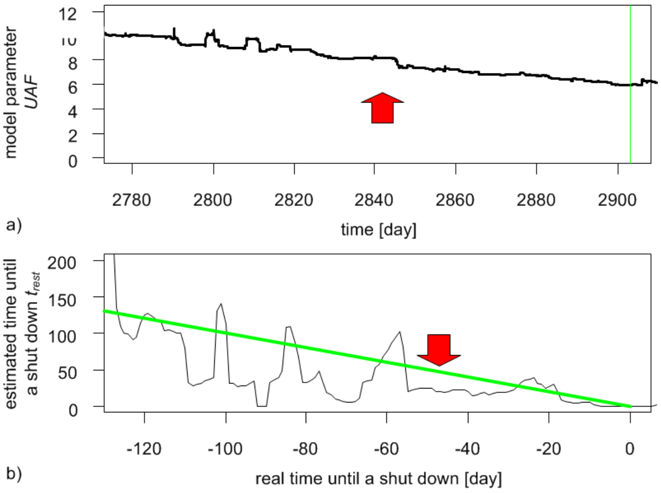


Fig. 10. Model parameter UAF for the time period from Fig. 9 with considered step in a) and the corresponding rest lifetime estimation in b)

The calculated model parameter UAF is sensitive for any not recorded technological handlings. During such a procedure some cleaning solution is added to the reactant. The result is a shortly better heat exchanger condition (higher value for the model parameter), because the amount of solid depositions decreases and the heat transfer increases.

Fig. 9a shows the last section of the data form Fig. 6b. Around day 2840 a step in the model parameter UAF was detected, see the arrow. The estimated rest lifetime is shown in Fig. 9b. It is easy to see that around the step in the model parameter the estimated rest lifetime becomes infinite large. In the period marked by the arrows no practical prediction is possible. 100 and 80 days before the shut down the prediction were disturbed by additional changes in the inlet temperatures.

In the new approach all regressions before the stepwise disturbance are calculated with the data from Fig. 9a. For all regressions after the stepwise regression the values before the disturbance are shifted upwards that the step in the model parameter disappeared. The new course of the model parameter is shown in Fig. 10a and the corresponding estimation of the rest lifetime is shown in Fig. 10b. It is easy to see that in this case in every time point a practical prediction is possible. It is important to know that Fig. 10a shows the model parameter for the view after day 2840. Fig. 11 shows all data from Fig. 6b and also at the first step at day 2720 a proper prediction is possible. The values in Fig. 11d are not infinite high; they are smaller than 1900 days.

4. CONCLUSION

In Friebel et al. (2009 and 2010) a simple method for lifetime estimation was presented and analyzed in some case studies. In the presented paper an additional representative parameter for the state of the heat exchanger is used with the explained method. The assumptions and simulations were tested by analyzing measurements of a real plant. By comparing the results and simulations the following results can be summarized.

- The classical parameter for the observation of the state of the heat exchanger is the degree of efficiency ε. This quality parameter is sensitive for drifts in the inlet temperatures, the fluid flows and the heat transfer coefficient.
- The model parameter UA, a combination of the heat transfer coefficient and the inner surface of the heat exchanger, is not sensitive for a drift in inlet temperature and fluid flows.
- If the flow cannot be measured then only the observed parameter UAF can be calculated. Hereby a constant flow is assumed, otherwise a drift in flow and in the heat transfer cannot be differed.
- Using the model parameter UAF is better than using the degree of efficiency ε.
- It would be better to use a quality parameter which is independent of the working point. Therefore the universal model parameter UA should be preferred if possible.
- The prediction of the rest lifetime can be made by a simple linear regression. The quality of the regression can be proven with a statistical t-test. Also a statistical based uncertainty of the predicted rest lifetime can be formulated.
- It is easier to detect and compensate the on-line cleaning in the model parameter UAF than in the degree of efficiency ε.
- By considering the steps caused by the on-line cleaning, the predicted rest lifetime becomes more practical with realistic predictions.

Further research work is planned in order to detect and to consider the change in flow during the data recording.

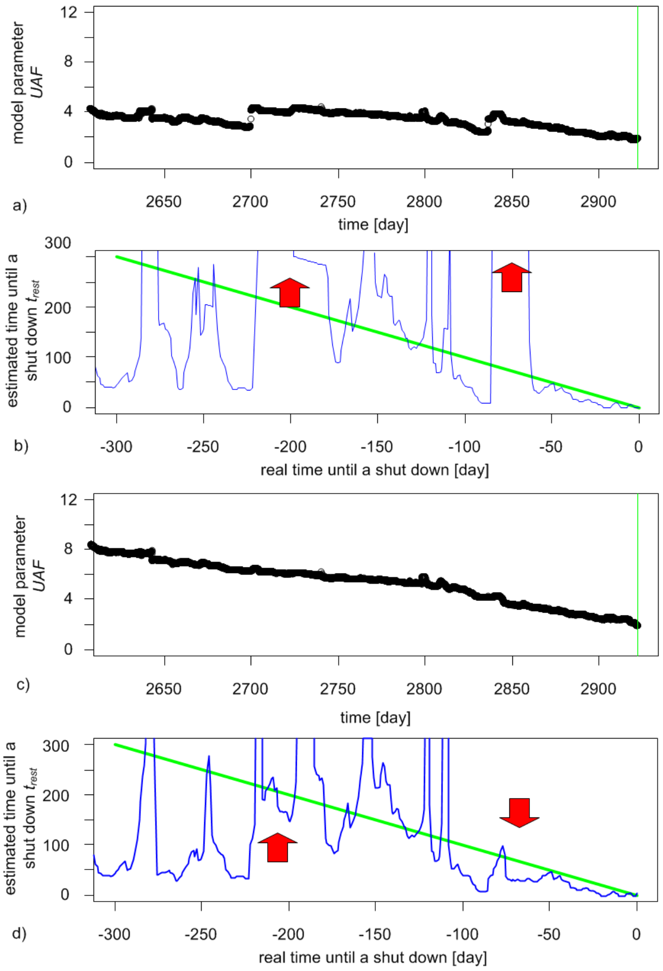


Fig. 11. Model parameter UAF for the time period from Fig. 6b with considered step in a) and the corresponding rest lifetime estimation in b)

REFERENCES

Friebel, T., Haber, R., Schmitz, U. (2010). *Cleaning time point prediction with heat exchanger fouling in a refenery*. Proceedings of 9th Int. Conf. Process Control, Kouty nad Desnou, Czech Republic

Friebel, T., Stockmann, M., Haber, R. (2009). *Heat exchanger fouling detection and lifetime estimation by regression*. Proceedings ICONS 2009 - 2nd IFAC Int. Conf. on Intelligent Control Systems and Signal Processing, Istanbul - Turkey

Montgomery, D., Peck, E.A., Vining, G.G. (2001). *Introduction to linear regression analysis*. Wiley and Sons, 2001, 3th edition, 44ff

Pearson, R.K. (2002). Outliers in process modeling and identification. *IEEE Trans. on Control Systems Technology*, 10, 1, 55-63.

Wagner, W. (2005). *Wärmetauscher - Grundlagen, Aufbau und Funktion thermischer Apparate* (Heat exchangers - Fundamentals, construction and function of thermal apparatus). Vogel Buchverlag, 3rd edition, 194ff

New Mathematical Tools for Analysis and Control of Platoons of Cars in Future Automated Highway Systems

Michael Šebek and Zdeněk Hurák*

Faculty of Electrical Eng., Czech Technical University in Prague, Czech Republic
{michael.sebek,zdenek.hurak}@fel.cvut.cz
http://dce.fel.cvut.cz/

Abstract. This paper introduces new mathematical tools for stabilization and asymptotic following in infinite platoons of vehicles in future automated highway systems. The platoon description, behavior analysis and control is approached in 2-D polynomial framework, that is, the dynamics of the problem are described using a fraction of two bivariate polynomials. In contrast to some previous works, the platoon here assumes a leader (and an infinite number of followers), therefore the often used bilateral z-transform should not be used here since it was developed for doubly infinite vehicular strings. The unilateral z-transform seems better suited. However, it brings about the need to take the boundary conditions into consideration; among other, the leader vehicle comes into the scene. The necessary formalism is introduced in the paper and used to provide elegant alternative proofs of some well-known facts about the platooning problem.

Keywords: Automated highway systems, automated guided vehicles, multidimensional systems, multivariable polynomials, polynomial equations, polynomial methods.

In current studies automated highway systems [2], platooning is conceived as a way of expanding the envelope of capacity and safety that can be achieved by road vehicles. When the vehicles are organized in platoons, they can operate much closer together than is possible under manual driving conditions. Each highway lane can therefore carry several times as much traffic as it can today, which should make it possible to greatly reduce highway congestion. Also, at close spacing aerodynamic drag is significantly reduced, which can lead to major reductions in fuel consumption and exhaust emissions. The high-performance vehicle control system also increases the safety of highway travel, reduces driving stress and tedium, and provides a very smooth ride.

In the microscopic description of the highway, vehicles are individually modeled. The vehicles headway is defined as the time taken for the vehicle to traverse

[*] This work has been supported by the Czech Ministry of Education within a project AMVIS ME10010.

the inter-vehicle spacing ahead of it. There are various control policies for the platoon characterized by different speed-spacing relationships that the vehicle control system aims to guarantee. A popular control policy is that of constant time headway which makes the desired ranges proportional to vehicle speeds.

When vehicles follow each other automatically in a lane, they behave like a coupled system, the behavior of which depends on the control actions of the individual vehicles. A phenomenon known as "slinky effect" [2] where a small tracking error or disturbance in the response of a lead vehicle gets amplified as it propagates along the platoon or string of vehicles is commonly observed in today's driving. The longitudinal control system of each system has to be designed so to guarantee platoon or string stability, which in turn implies the absence of slinky effects.

As the highway lane capacity increases with the length of the platoon [2], very long platoons are desirable. This fact is reflected in theoretical literature differently: Either platoons with a finite number of cars are studied and then the limit infinite case is taken. Or doubly infinite strings of vehicles are considered allowing to apply bilateral transforms that neglect any boundary conditions.

Instead, we advocate the use semi-infinite platoons with a leader where the leading vehicle is labeled by 0 and the follow-up cars are numbered by $1, 2, \ldots$. Positions, distances and velocities in the platoon are described by spatial sequences of time functions corresponding to the equally indexed vehicles $\{f(t,k)\} = f(t,0), f(t,1), f(t,2), \ldots,\ t \in [0,\infty)$. To handle such sequences, we introduced an original joint unilateral Laplace and (shifted) unilateral z-transform denoted $\mathcal{LZ}_1$which is defined by

$$\mathcal{LZ}_1\left\{f(t,k)\right\} = \int_{0^-}^{\infty}\left(\sum_{k=1}^{\infty} f(t,k)z^{-k}\right)e^{-st}\,dt.$$

Application of the $\mathcal{LZ}_1$ transform opens the door to rich world 2-D polynomial systems theory [5]. In addition, it reveals that the leading vehicle movement is actually boundary condition for the coupled platoon system.

The transform has been employed to solve various problems of analysis and control for platoons with leader [1, 6].

References

1. Hurak, Z., Sebek, M.: 2D polynomial approach to stability of platoons of vehicles. In: Proceedings of the 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems. vol. 2. Annecy, France (2010)
2. Ioannou, P.: Automated Highway Systems. Springer, 1st edition. edn. (Nov 2010)
3. Liang, C.Y., Peng, H.: String stability analysis of adaptive cruise controlled vehicles (2000)
4. Santhanakrishnan, K., Rajamani, R.: On spacing policies for highway vehicle automation. In: Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334). pp. 1509–1513. Chicago, IL, USA (2000)
5. Sebek, M.: On 2-D pole placement. IEEE Transactions on Automatic Control 30(8), 819–822 (1985), 10.1109/TAC.1985.1104065

6. Sebek, M., Hurak, Z.: 2-D polynomial approach to control of leader following vehicular platoons. In: 18th IFAC World Congress. IFAC, Milano, Italy (Aug 2011), submitted for publication.

Use of cross wavelet transform for diagnosis of oscillations due to multiple sources

Selvanathan Sivalingam [*] **Morten Hovd** [**]

[*] *Engineering Cybernetics Department, NTNU, Trondheim 7491, Norway (e-mail: selvanat@itk.ntnu.no)*
[**] *Engineering Cybernetics Department, NTNU, Trondheim 7491, Norway (e-mail: morten.hovd@itk.ntnu.no)*

Abstract: Oscillations are the most prominent indications of deteriorated controller performance. Control loop oscillations are a common type of plant-wide disturbance and the root-causes can be one or more among poorly tuned controllers, process or actuator non-linearities, presence of model plant mismatch and oscillatory disturbances. This article addresses detection and diagnosis of oscillations in measurements due to multiple sources under a framework of internal model control. A pattern recognition based approach using cross wavelet transforms is proposed to pinpoint the source(s) of oscillation in the control loops. The phase information in wavelet domain between input and output signals is exploited to diagnose the source(s) of oscillations.

Keywords: wavelet transform, oscillation, valve stiction, phase, pattern recognition

1. INTRODUCTION

It is well known that performance degradation in control loops manifests as one or more of the following: (*i*) poor set point (SP) tracking (*ii*) oscillations (*iii*) poor disturbance rejection and (*iv*) excessive final control element variation. Oscillations are attributed to one or more among poor controller tuning, process or actuator non-linearities, presence of model plant mismatch or oscillatory disturbances. A tool to help the engineer should therefore automatically bring oscillatory loops to his or her attention, characterize them and highlight the presence of plant wide oscillations. Several authors have addressed the detection of oscillatory measurements in process data. Early works appear in Hägglund (1995) followed by (Thornhill and Hägglund 1997, Forsman and Stattin 1999, Rengaswamy *et al.* 2001, Tangirala *et al.* 2007). Hägglund (1995) proposed a technique to detect oscillating loops "on-line" using the IAE criterion. This method does not assume any particular shape for oscillation and only requires the measurement to deviate significantly from the set point. Hägglund (1995) also proposed a diagnostic procedure for finding the source of oscillation and eliminating it. The diagnostic procedure is carried out by disconnecting the feedback (*i.e.* switching the controller to manual mode). This approach is simple and efficient and probably the most comprehensive procedure available for diagnosing root cause for oscillations. However, switching the controller to manual mode may not always be allowed, especially if the loop is deemed critical. Further, it will not be possible to apply this approach on thousands of loops in a routine fashion. Thornhill and Hägglund (1997) presented an offline technique for detecting oscillation using a regularity factor. This method requires the user to specify the root-mean-square value of the noise and a thresholds a nontrivial task when applied to hundreds of loops.

Thornhill and Hägglund (1997) and Thornhill *et al.* (2003) proposed a set of procedures to detect and diagnose oscillating loops using offline data. They combine the techniques of controller performance assessment along with operational signatures (OP-PV plots) and spectral analysis of the controller error for diagnosis. This technique, though not completely automated, can distinguish the cause of oscillation as one of the following: (*i*) poor tuning (*ii*) nonlinearity or (*iii*) external disturbance. However, the downside lies in manually inferring the loop signatures that are based on spectral analysis or on a map of controller output (OP) versus process variable (PV) and isolating the oscillating portion from the entire data. Horch (1999) presented a simple, practical approach to distinguish oscillating loops that are caused by external disturbances and static friction. This approach is based on cross-correlation between the controller output (OP) and process output (PV). The cross-correlation technique failed when the data had intermittent oscillations and the set-point was also changing. Horch and Isaksson (1998) also proposed a technique to identify stiction using nonlinear filters. The method assumed that information such as mass of stem, diaphragm area, and so on for each valve is readily available. Since in a typical process industry facility there can be thousands of control loops, it may be nearly impossible to build/maintain a knowledge base of control valves, making this technique difficult to implement.

Choudhury *et al.* (2004) used higher order statistics for detecting nonlinearity in data and have extended the method for diagnosing stiction by fitting an ellipse of the OP-PV plot and inferring the stiction from an assumed stiction model. The success of this approach lies in correctly identifying the oscillation period and its start and end point in the OP-PV data. Tangirala *et al.* (2007) proposed non-negative matrix factorization for detection and diagnosis of plant-wide oscillations based on source separation techniques. As can be seen, the task of detecting stiction or other nonlinearities in valves from routine operating data is a challenging task. To summarize, data driven techniques that are presented in the literature till date are useful in (a) assessing the performance of the controller by calculating a figure of merits given that the cause of poor-performance is only due to either an aggressive or sluggishly tuned controller in pure feedback control, (b) detecting oscillating loops with an user-specified parameter, and (c) limited diagnosis of the cause of oscillation based on cross-correlation, power spectral analysis, or OP-PV plots. The current approaches lack (a) the capability to efficiently diagnose oscillations due to multiple sources, (b) the ability to diagnose the causes of time-varying oscillations and (c) an automated means of oscillation diagnosis.

In this work, we have attempted to address some of the aforementioned drawbacks by using wavelet and cross wavelet transforms. This paper is organized as follows: A brief introduction on wavelet transforms is given in Section 2. Problem statement and proposed methodology for an IMC framework are given in Section 3 followed by simulation studies in Section 4. The paper ends with concluding remarks in Section 5.

2. WAVELET ANALYSIS

The main benefit of wavelet analysis over Fourier analysis is that both time and frequency localization can be achieved in the former. This is because wavelet analysis employs a wave packet whereas Fourier analysis uses an infinite wave train of sines and cosines. In recent years, wavelet power transforms have become increasingly popular (Bloomfield *et al.* 2004) while the additionally available phase information has remained untapped.

Wavelet analysis has become a common tool for analyzing localized variations of power within a time series. By decomposing a time series into time–frequency space, one is able to determine both the dominant modes of variability and how those modes vary in time. The wavelet transform can be used to analyze time series that contains varying power at different frequencies. The term “wavelet function” is used generically to refer to either orthogonal or non-orthogonal wavelets. The term “wavelet basis” refers only to an orthogonal set of functions (Torrence and Compo 1998). The use of an orthogonal basis implies the use of the discrete wavelet transform, while a non-orthogonal wavelet function can be used with either the discrete or the continuous wavelet transform. The continuous wavelet transform was developed as an alternative approach to the short-time Fourier transform because the spectrogram is limited in resolution by the extent of the sliding window function. The wavelet analysis is done in a similar way to the short-time fourier transform (STFT) analysis in which the signal is multiplied with a function (*i.e.* the wavelet) similar to the window function in the STFT and the transform is computed separately for different segments of the time-domain signal. However, there are two main differences between the STFT and continuous wavelet transform (CWT). The Fourier transforms of the windowed signals are not taken, and therefore single peak will be seen corresponding to a sinusoid, *i.e.,* negative frequencies are not computed. The width of the window is changed as the transform is computed for every single spectral component, which is the most significant characteristic of the wavelet transform. Throughout this work, we use the standard Morlet wavelet: a Gaussian modulated sine wave of the form (Torrence and Compo 1998)),

$$\psi(\eta) = \pi^{-1/4} e^{i\omega_0 \eta} e^{-\eta^2/2} \tag{1}$$

The quantity $\pi^{-1/4}$ is a normalization factor, $\eta = \dfrac{n}{s}$ is the dimensionless time parameter, n is the time parameter and s is the scale of the wavelet, $\omega_0 = s\omega$ is the dimensionless frequency parameter and ω is the frequency parameter.

It is to be noted that an infinite number of mother wavelets are available, including the derivative-of-a-Gaussian (DOG) and Paul wavelets. In this work, the complex Morlet wavelet is chosen since it yields a complex wavelet transform, containing information on both amplitude and phase. Since DOG wavelets are entirely real, they may not be used for phase analysis, as their real transforms hold only information on amplitude. Alternatively, the complex Paul wavelet could be employed. However, as the Paul function is more sharply defined in time (in comparison to the more sinusoidal Morlet function), it is better suited for studying pulse-like variations.

2.1 Continuous wavelet transform

The continuous wavelet transform is defined as follows:

$$W(a, \tau) = \frac{1}{\sqrt{a}} \int x(t) \psi^* \frac{(t - \tau)}{a} dt \tag{2}$$

where $\psi(t)$ denotes the mother wavelet. The parameter a represents the scale index which is the reciprocal of frequency and the parameter τ indicates the time shifting (or translation). High scales (low frequencies) correspond to the global information of a signal that usually spans the entire signal, whereas low scales (high frequencies) correspond to detailed information of a hidden pattern in the signal that usually lasts for a relatively shorter time.

The CWT has edge artifacts because the wavelet is not completely localized in time. Cone of Influence (COI) has been defined as the area in which the wavelet power caused by a discontinuity at the edge has dropped by e^{-2} of the value at the edge. Due to the edge effect, confidence limits for the wavelet spectra are required and hence to determine significance levels for wavelet spectra an appropriate background spectrum is required. Red noise spectrum is used as background spectrum and it has the characteristic feature of increasing power with decreasing frequency. In this work, continuous wavelet transform and cross wavelet transform are performed using Morlet wavelet to study the time-frequency properties of the of the output sequences.

2.2 Wavelet scale Vs. Fourier period

The scale can be defined as the distance between oscillations in the wavelet, or it can be some average width of the entire wavelet . The period (or inverse frequency) is the approximate Fourier period that corresponds to the oscillations within the wavelet. For all wavelets, there is a one-to-one relationship between the scale and period. The relationship can be derived by finding the wavelet transform of a pure cosine wave with a known Fourier period, and then computing the scale at which the wavelet power spectrum reaches its maximum.

For some wavelets the period has more meaning than others. For the Morlet, which has several smooth oscillations, the period is a well-defined quantity which measures the approximate Fourier period of the signal. For the Morlet wavelet , l=1.03a, where l is the Fourier period, indicating that for the Morlet wavelet the wavelet scale (a) is almost equal to the Fourier period.

2.3 Cross-wavelet transform (XWT)

The cross-wavelet transform between two time series X and Y, with wavelet transforms $W_x(f,\tau)$ and $W_y(f,\tau)$ is simply the multiplication of the first complex wavelet transform with the complex conjugate of the second

$$W_{xy}(f,\tau) = W_x(f,\tau)W_y^*(f,\tau) \tag{3}$$

where $f \approx \dfrac{1}{a}$ when $f_0 = 2\pi$ for the Morlet wavelet. The wavelet scale, a, is inversely proportional to the central frequency of the wavelet (f_o).

While a wavelet power spectrum depicts the variance of a time series with times of large variance showing large power, the cross wavelet power of two time series depicts the covariance between these time series. Additionally, cross wavelet power has a known distribution of confidence levels which is proportional to the square root of the product of two χ^2 distributions (Torrence and Compo 1998). This allows cross wavelet power to be used as a quantified measure of the similarity of power between two time series.

2.4 Phase difference analysis

In complement to wavelet analysis, the phase spectrum analysis can be used to characterize the association between signals. The phase difference provides information on the sign of the relationship (*i.e.,* in phase or out of phase). As the Morlet wavelet is a complex wavelet, the cross wavelet transform relation can be written in terms its modulus $|W_x(f,\tau)|$ and its phase, $\phi_x(f,\tau) = tan^{-1}\dfrac{\text{Imag}(W_x(f,\tau)}{\text{Real}(W_x(f,\tau)}$. Similarly, with the cross wavelet transform $W_{xy}(f,\tau)$ the phase relation between the time series X and Y can be computed using the relation, $\phi_{xy}(f,\tau) = tan^{-1}\dfrac{\text{Imag}(W_{xy}(f,\tau)}{\text{Real}(W_{xy}(f,\tau)}$.

The estimation of phase spectrum in Fourier domain between two time series contains relatively large errors compared to that computed in wavelet domain(**?**). Moreover, phase difference is localized in time and frequency in wavelet domain. The arrows in the cross-wavelet transform plot indicate the direction of the phase difference between the variables. The phase arrows pointing right indicate that the variables are in-phase, pointing left indicate the variables are anti-phase, down indicate phase lead of 90° and up refer to phase lag of 90°. The direction of the phase difference between the variables plays a crucial role in diagnosing the source(s) of oscillation in the work.

2.5 Average angle

As we are interested in the phase difference between the components of the two time series, it is necessary to estimate the mean and confidence interval of the phase difference. The circular mean of the phase are used over regions with higher than 5% statistical significance that are outside the COI to quantify the phase relationship. This is a useful and general method for calculating the mean phase. The circular mean of a set of angles (ai , i=1...n) is defined as

$$a_m = \arg(X,Y) \tag{4}$$

with $X = \sum_{i=1}^{n} cos(ai)$ and $Y = \sum_{i=1}^{n} sin(ai)$. It is difficult to calculate the confidence interval of the mean angle reliably since the phase angles are not independent. The number of angles used in the calculation can be set arbitrarily high simply by increasing the scale resolution. However, it is interesting to know the scatter of angles around the mean. For this the circular standard deviation is defined as

$$s = \sqrt{-2\ln\frac{R}{n}} \tag{5}$$

where $R = \sqrt{X^2 + Y^2}$. The circular standard deviation is analogous to the linear standard deviation in that it varies from zero to infinity. It gives similar results to the linear standard deviation when the angles are distributed closely around the mean angle. In some cases there might be reasons for calculating the mean phase angle for each scale, and then the phase angle can be quantified as a number of years. The XWT phase angle within the 5% significant regions and outside the COI has the mean phase 176±12 (where ± designates the circular standard deviation).

3. PROBLEM STATEMENT AND PROPOSED METHOD

Oscillations in model based control loops occur due to either one of (i) valve stiction (ii) model plant mismatch, (iii) external oscillatory disturbances or combination of any of these. It becomes vital to diagnose the causes of oscillations in order to take the appropriate remedial action. A procedure based on pattern recognition techniques using cross wavelet transform is devised in this article to diagnose the cause(s) of the oscillation. The problem is setup in the internal model control (IMC) framework (Figure 1). Cross wavelet transform of input and plant and that of input and model output are computed and thereby a specific pattern is sought for root cause diagnosis of oscillation using the direction of wavelet phase difference between the variables.

To illustrate the idea of cross-wavelet transform for an input-output system, an open-loop process with $G_p(s) = \dfrac{1}{10s+1}$ is considered. The process is simulated for a sinusoidal input having two frequencies and the time domain plots of input and output are given in Figure 2.

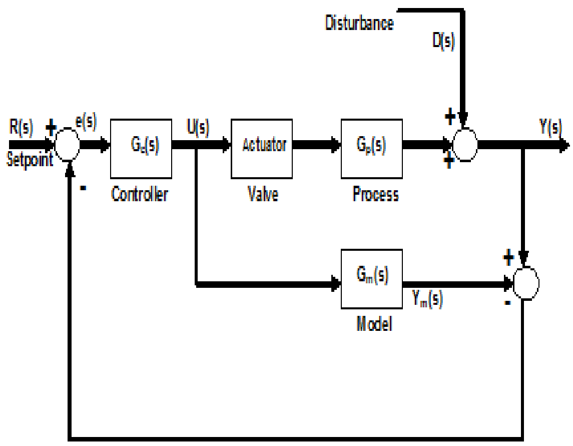


Fig. 1. Schematic representation of internal model control with actuator

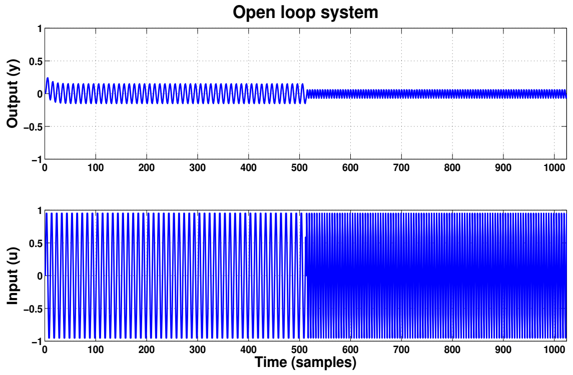


Fig. 2. Time domain behavior of input and output signals considered for interpretation of wavelet analysis

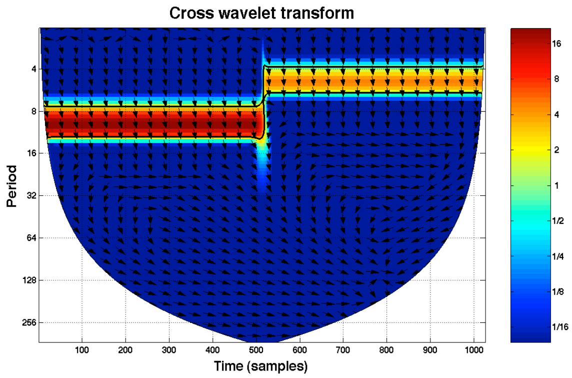


Fig. 3. Cross wavelet transform between input and output signals

The cross wavelet transform plot between two quantities u and y is shown in Figure 3.

It is known from Figure 3 that the quantities u and y show high common power at two frequencies between two different time intervals (0.1 Hz, 0-511 and 0.2 Hz, 512-1024) and the arrows indicate the direction of the wavelet phase between u and y *i.e.,* u leads y by 90° (pointing down). Based on the properties of cross wavelet transform, wavelet phase difference and linear time invariant systems theory, the following methodology is proposed to diagnose the source(s) of oscillation in a control loop.

The quantities controller output (u), process output (y) and model output (y_m) of an oscillating control loop are obtained either from simulation or from industry. The cross wavelet transforms, $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ are computed. By comparing the direction of wavelet phase, the following conclusions can be drawn.

Based on the properties of cross wavelet transform, wavelet phase difference and linear time invariant systems theory, the following methodology is proposed to diagnose the source(s) of oscillation in a control loop.

- **Valve stiction**: If the oscillating source is only due to valve stiction, the cross wavelet transform plots should not only exhibit harmonics but also discontinuities.
- **Model plant mismatch:** If the source is due to model plant mismatch, which among the gain, time constant and delay causes the oscillation needs to be pinpointed.
  Gain mismatch: Gain mismatch theoretically does not affect the wavelet phase spectrum. Hence, the phase difference between $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ is zero at the fundamental frequency of oscillation. Cross wavelet spectrum ratio is constant at non-zero value at the frequency of oscillation. Further, the average phase angles of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ estimated at the frequency of oscillation are theoretically same. In addition to this, the arrows in cross wavelet transform plots will be in same direction.
  Time constant mismatch: Time constant mismatch affects both cross wavelet spectrum ratio and phase spectrum. The plots of both $\frac{|W_{uy}(f,\tau)|}{|W_{uy_m}(f,\tau)|}$ and $\phi_{uy}(f,\tau) - \phi_{uy_m}(f,\tau)$ clearly show that the time constant mismatch significantly changes behavior of absolute cross wavelet spectrum ratio and the phase spectrum. Consequently, the average phase angles of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ estimated at the frequency of oscillation are different. The arrows in cross wavelet transform plots will be in same direction since the effect of time constant mismatch on phase spectrum is minimum.
  Delay mismatch: Delay mismatch theoretically does not affect the magnitude of cross wavelet phase spectrum. Hence, cross wavelet spectrum ratio is unity at the frequency of oscillation. In contrast, the phase difference between $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ is non-zero at the fundamental frequency of oscillation and the average phase angles of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ estimated at the frequency of oscillation are different. Consequently, the arrows in cross wavelet transform plots will be in opposite direction.

4. SIMULATIONS

A control system consisting of a process characterized by the transfer function $G_p = \dfrac{K_p}{\tau_p s+1}e^{-d_p s}$and model $G_m = \dfrac{K_m}{\tau_m s+1}e^{-d_m s}$ is simulated with IMC controller for a unit step change in the set point. The different

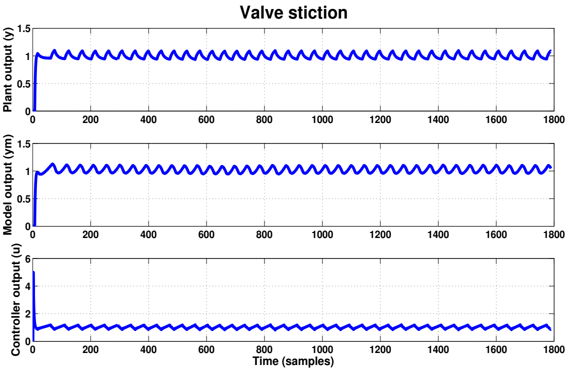


Fig. 4. Time domain behavior of plant, model and controller outputs for the valve stiction as the source of oscillation.

case studies analyzed for the diagnosis of oscillation in a control loop are (i) oscillation due to valve stiction (ii) oscillation due to valve stiction and oscillatory disturbance (iii) oscillation due to gain mismatch (iv) oscillation due to gain mismatch and oscillatory disturbance and (v) oscillation due to delay mismatch.

4.1 Diagnosis of valve stiction

A simple yet efficient one parameter model proposed by (Hägglund 1995) is used to generate oscillations due to valve stiction. The model is

$$x(t) = \begin{cases} x(t-1) & |u(t) - x(t-1)| \leq d \\ u(t) & \text{otherwise} \end{cases} \tag{6}$$

Here $u(t)$ and $x(t-1)$ are present and past valve outputs, $u(t)$ is the present controller output, and d is the valve stiction band. The valve stiction band is expressed in terms of the percentage or fraction of valve movement corresponding to the amount of stiction present in the valve. For instance, if 100 units of force are required to open the valve completely from completely closed position and 10 units of force is required to overcome the amount of static friction in the valve, stiction band is 10% or 0.1. The stiction band of 0.1 is used in the simulation. Model plant mismatch is introduced by changing the values of gain, time constant and delay appropriately in the process. The sinusoidal disturbance of frequency 0.01 Hz is considered for the simulation.

The cross wavelet transform computed between controller output and plant output is compared with that computed between controller output and model output. In the case of oscillation due to valve stiction (Figure 4), the plots of cross wavelet transform (Figures 6 & 7) not only show harmonics but also discontinuities which are the characteristics of a sticky valve. Figures 8 and 9 clearly indicate the presence of the valve stiction as one of the sources of oscillation between 800 and 1600 s and the other being the oscillatory component of frequency 0.01 Hz throughout.

4.2 Gain mismatch

If the oscillation is only due to MPM, there will be clearly a single frequency in the cross wavelet transform plot. The

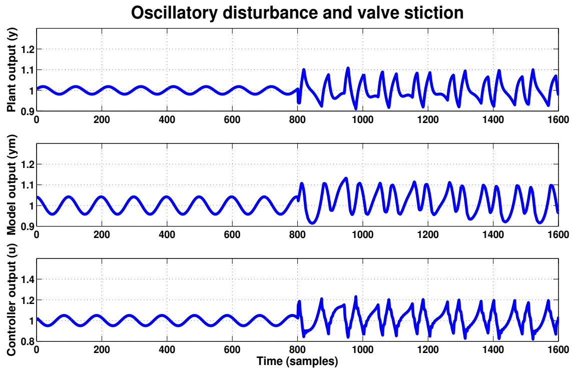


Fig. 5. Time domain behavior of plant, model and controller outputs for the case oscillatory output and valve stiction as the sources of oscillation

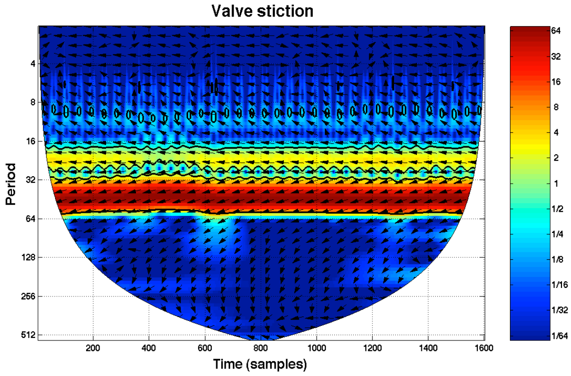


Fig. 6. Cross wavelet transform plot between u and y_p when the oscillation is only due to valve stiction.

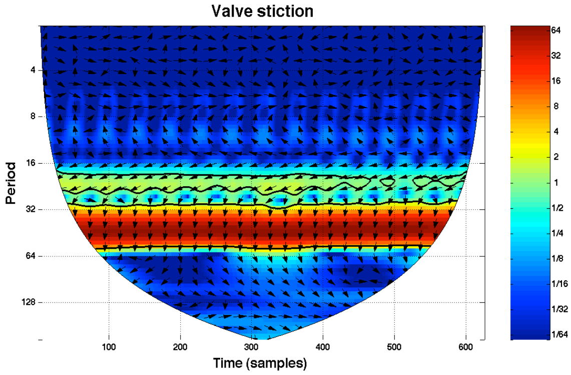


Fig. 7. Cross wavelet transform plot between u and y_m when the oscillation is only due to valve stiction.

system is simulated to generate oscillation in the plant output due to gain mismatch (Figure 10). The cross wavelet transforms, $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ are estimated from where the quantities, absolute cross wavelet transform ratio $\frac{|W_{uy}(f,\tau)|}{|W_{uy_m}(f,\tau)|}$, the phase difference, $\phi_{uy}(f,\tau) - \phi_{uy_m}(f,\tau)$ and average phase angles of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ at the frequency of oscillation are obtained. The value of $\frac{|W_{uy}(f,\tau)|}{|W_{uy_m}(f,\tau)|}$ is found constant at 2.4 (Figure 13), the phase difference is zero (Figure 14) at the frequency of oscillation and the average phase angles are -2.551 and -2.554. The plots of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ (Figures 11 & 12) show

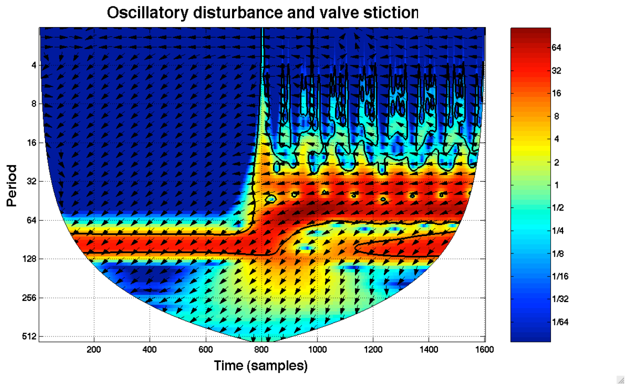


Fig. 8. Cross wavelet transform plot between u and y_p when the oscillation is due to oscillatory disturbance and valve stiction.

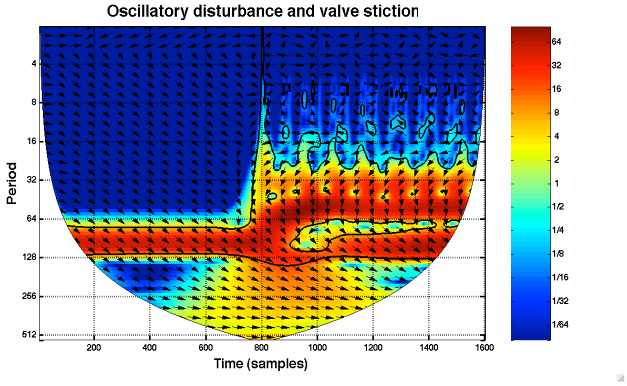


Fig. 9. Cross wavelet transform plot between u and y_m when the oscillation is due to oscillatory disturbance and valve stiction.

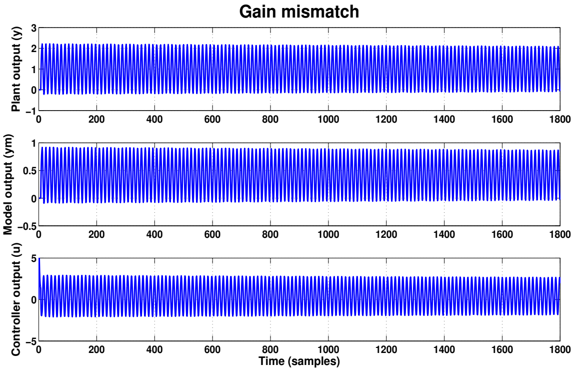


Fig. 10. Time domain behavior of plant, model and controller outputs for the case of gain mismatch as the source of oscillation.

that the arrows are in same direction. This is also in line with the fact that the phase spectrum is unaffected by the changes in gain.

4.3 Time constant mismatch

The control loop whose outputs are given in Figure (15) is analyzed for diagnosing the source(s) of oscillations. Figures (18) and (19) indicate the presence of time constant mismatch as the source of oscillation. Further, the closeness of the values of average phases angles (-2.4244 and -2.1488) strengthens the presence of time constant mismatch. Figures (16) and (17) indicate that the arrows are in the same direction. This is also expected in the case of time constant mismatch as the source of oscillation since the effect of time constant mismatch on phase spectrum is minimal.

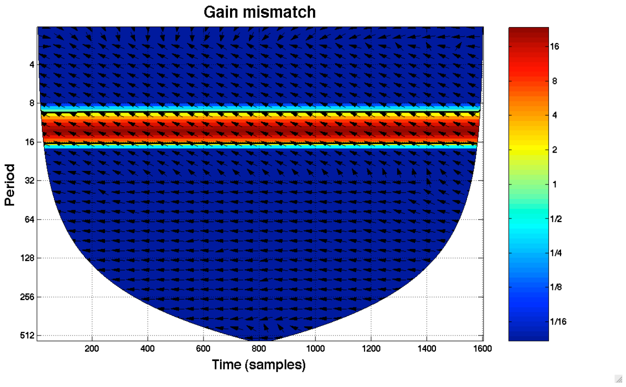


Fig. 11. Cross wavelet transform plot between u and y_p when the oscillation is gain mismatch

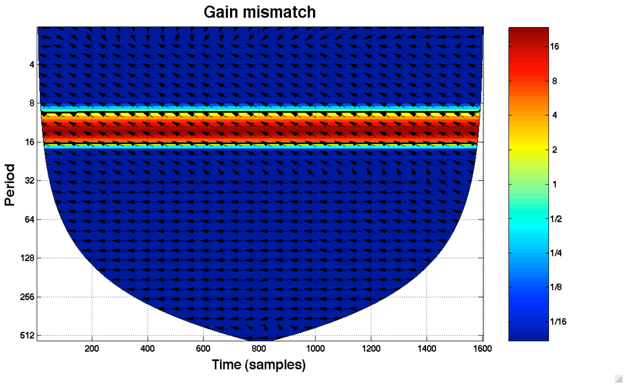


Fig. 12. Cross wavelet transform plot between u and y_m when the oscillation is gain mismatch

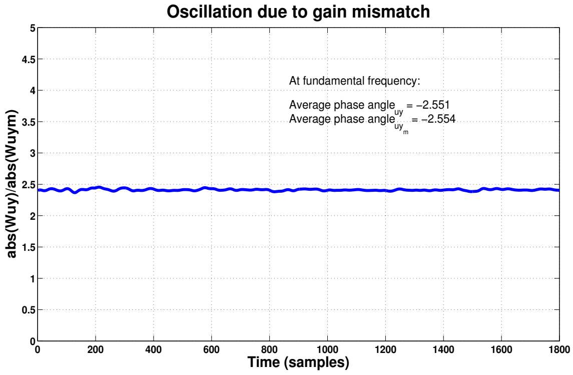


Fig. 13. A plot of ratio of cross wavelet transforms when the oscillation is due to gain mismatch.

4.4 Oscillation due to delay mismatch

The control loop whose outputs are given in Figure 20 is analyzed for diagnosing the source(s) of oscillations. The cross wavelet transforms, $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$

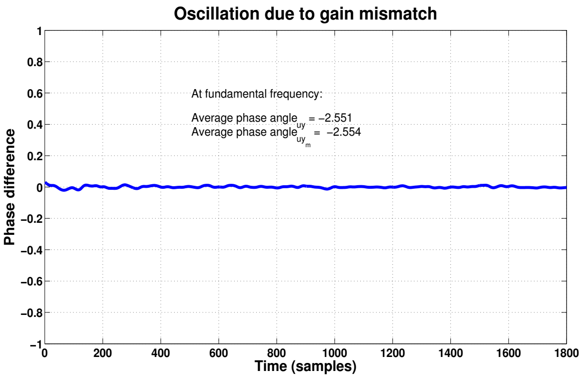


Fig. 14. A plot of phase difference when the oscillation is due to gain mismatch

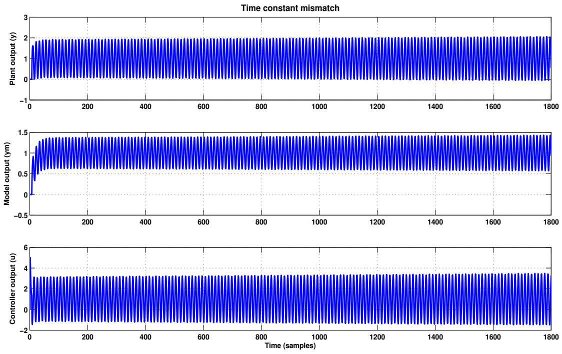


Fig. 15. Time domain plots of plant, model and controller outputs for the case of time constant mismatch as the source of oscillation

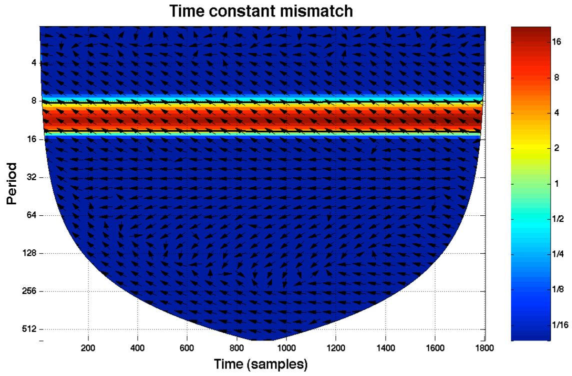


Fig. 16. Cross wavelet transform plot between u and y_p when the oscillation is due to time constant mismatch

are estimated from where the quantities, absolute cross wavelet transform ratio $\frac{|W_{uy}(f,\tau)|}{|W_{uy_m}(f,\tau)|}$, the phase difference, $\phi_{uy}(f,\tau) - \phi_{uy_m}(f,\tau)$ and average phase angles of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ at the frequency of oscillation are obtained. The value of $\frac{|W_{uy}(f,\tau)|}{|W_{uy_m}(f,\tau)|}$ is found unity (Figure 23), the phase difference is non-zero (Figure 24) at the frequency of oscillation and the average phase angles are -2.9498 and 1.751. These observations show the presence of delay mismatch. Further, the plots of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ (Figures 21 & 22) indicate that the arrows are

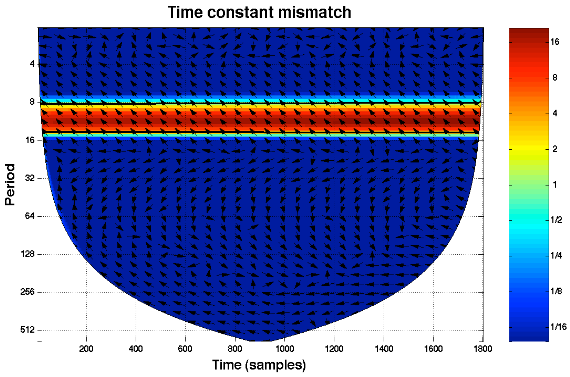


Fig. 17. Cross wavelet transform plot between u and y_m when the oscillation is due to time constant mismatch

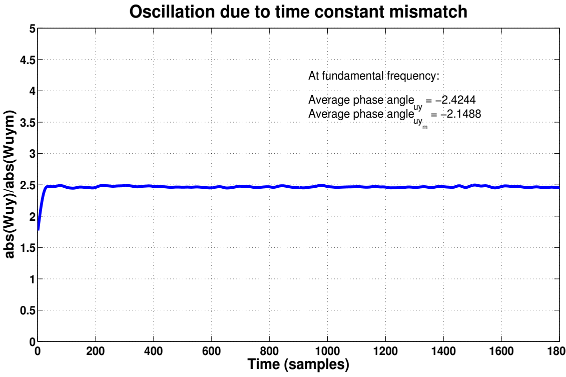


Fig. 18. A plot of ratio of cross wavelet transforms when the oscillation is due to time constant mismatch

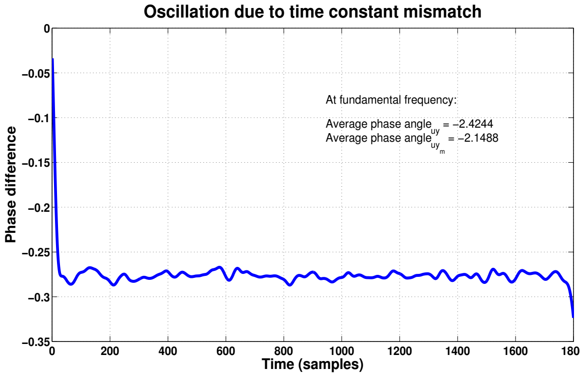


Fig. 19. A plot of phase difference when the oscillation is due to time constant mismatch

in opposite direction strengthening the presence of delay mismatch as the source of oscillation.

5. CONCLUSIONS

A pattern recognition technique combined with two key measures namely, absolute cross wavelet transform ratio and wavelet phase difference for the diagnosis of control loop oscillations in internal model control systems due to multiple sources has been developed. A diagnostic study of oscillation due to either one of valve stiction, model plant mismatch, oscillatory disturbance or combination of these has been presented. The oscillations due to valve stiction manifest as harmonics as well as discontinuities

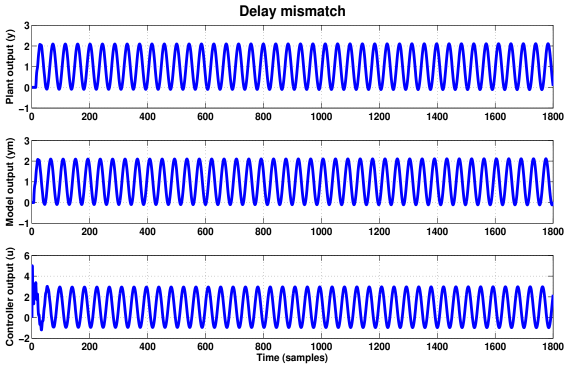


Fig. 20. Time domain plots of plant, model and controller outputs for the case of delay mismatch as the source of oscillation.

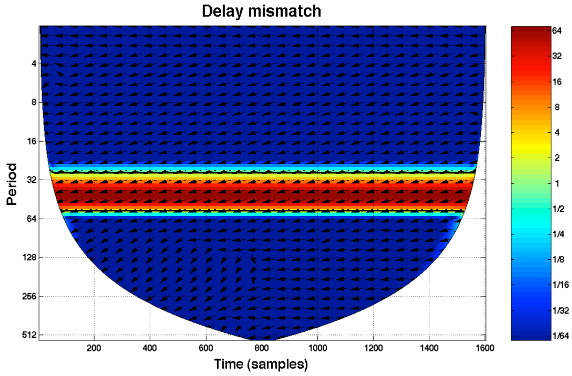


Fig. 21. Cross wavelet transform plot between u and y_p when the oscillation is due delay mismatch..

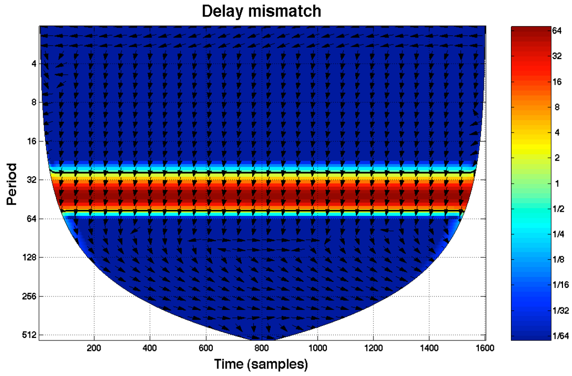


Fig. 22. Cross wavelet transform plot between u and y_m when the oscillation is due delay mismatch..

in the cross wavelet transform plots whereas oscillation due to model plant mismatch leaves distinct signatures in the phase information (arrows). If the oscillations are due to gain mismatch, the absolute cross wavelet transform is constant at non-zero value at the frequency of oscillation and the wavelet phase difference is zero. Further, the plots of $W_{uy}(f,\tau)$ and $W_{uy_m}(f,\tau)$ show the arrows are in same direction which strengthens the finding of the gain mismatch as the source of oscillation. If the oscillation is due to time constant mismatch, both the quantities, the absolute cross wavelet transform and the the wavelet phase difference are affected. On the other hand, oscillation due to delay mismatch results in a directional change in the phase difference while the absolute cross wavelet transform ratio is unity at the frequency of oscillation.

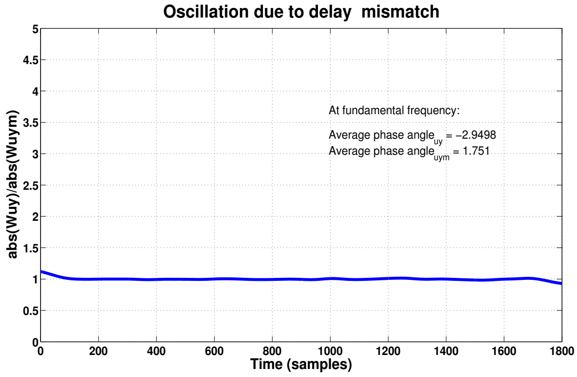


Fig. 23. A plot of ratio of cross wavelet transforms when the oscillation is due to delay constant mismatch

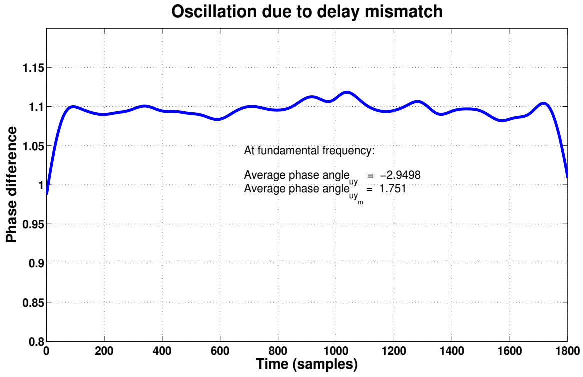


Fig. 24. A plot of phase difference when the oscillation is due to delay mismatch

REFERENCES

Bloomfield, D.S., R.T.J. McAteer, B.W. Lites and P.G. Judge (2004). Wavelet phase coherence analysis: Application to a quiet-sun magnetic element. *The Astrophysical Journal* **617**, 623–632.

Choudhury, M.A.A.S., S.L. Shah and N.F. Thornhill (2004). Detection and quantification of control valve stiction. In: *DYCOPS*. Boston, USA.

Forsman, K. and A. Stattin (1999). A new criterion for detecting oscillations in control loops. In: *CP8-3*. European control conference. Karlsruhe, Germany.

Hägglund, T. (1995). A control-loop performance monitor. *Control Engineering Practice* **3**, 1543–1551.

Horch, A. (1999). A simple method for the detection of stiction in control valves. *Control Engineering Practice* **7**, 1221–1231.

Horch, A. and A.J. Isaksson (1998). A method for detection of stiction in control valves. In: *Online fault detection and supervision in the chemical process industry*. IFAC Workshop. Lyon, France. p. 4B.

Rengaswamy, R., T. Hägglund and V. Venkatasubramanian (2001). A qualitative shape analysis formalism for monitoring control loop performance. *Engineering Applications of Artificial Intelligence* **14**, 23–33.

Tangirala, A.K., J. Kanodia and S.L. Shah (2007). Non negative matrix factorization for detection and diagnosis of plant wide oscillations. *Industrial & Engineering Chemistry Research* **46**, 801–817.

Thornhill, N.F. and T. Hägglund (1997). Detection and diagnosis of oscillation in control loops. *Control Engineering Practice* **5**, 1343–1354.

Thornhill, N.F., B. Huang and H. Zhang (2003). Detection of multiple oscillations in control loops. *Journal of Process Control* **13**, 91–100.

Torrence, C. and G.P. Compo (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society* **79**, 61–78.

Relay Identification Analyzing Non-symmetrical Oscillations for Optical Plant

Mikuláš Huba*,and Peter Ťapák***

** Institute of Control and Industrial Informatics, FEI STU, Ilkovičova 3, SK-812 19 Bratislava*
Slovakia (Tel: +421-2 -60291 771; e-mail: mikulas.huba@ stuba.sk).
*** MI/PRT, FernUniversität in Hagen, Universitätsstr. 27, D-58084 Hagen*
Germany (e-mail: mikulas.huba@fernuni-hagen.de)

Abstract: The paper deals with approximation of systems with the dominant first order dynamics by the Integrator Plus Dead Time (IPDT) model based on the analysis of the nonsymmetrical oscillations with possible offset arising typically under relay control. The analytical derivation is illustrated by results achieved by identification of optical plant. The results are experimentally verified by PI controller tuned using the identification results. Process parameters in various operating points are analyzed and the robutst controller tuning based on performance portrait analysis is employed.

Keywords: relay identification, integrator plant, nonsymmetrical oscillations.

1. INTRODUCTION

The relay feedback test is very popular approach used in several commercial autotuners. The current research in this area was closely analyzed in (Tao Liu, Furong Gao 2009). There are two types of relay tests, unbiased and biased. When using the unbiased test the process gain can be highly deflorated by a load disturbance. Many relay feedback methods have been proposed to reject static disturbances (Hang, Åström, & Ho, 1993; Park, Sung, & Lee, 1997, 1998; Shen, Wu, & Yu, 1996). Their approaches bias the reference value of the relay on–off as much as a static disturbance (that must be known in advance), in order to achieve the same accuracy as in the case of no disturbance. Nevertheless none of these approaches can be applied to large static disturbance, of which the magnitude is bigger than that of the relay. By inserting a proportional integral (PI) controller behind the relay for the test, (Sung and Lee, 2006) proposed an identification method for application against large static disturbance, larger than the magnitude of the relay. The drawback of the method is given be necessity to tune an additional controller.

Another important question is related to the models used for approximating the plant dynamics. Almost 70 years ago, Ziegler and Nichols (Ziegler & Nichols, 1942) proposed to use the sustained oscillations for process dynamics characterization giving finally PID controller tuning, whereby the process dynamics approximation was equivalent to the use of the IPDT model. It is, however, well known that the method is appropriate also for dealing with many systems with more complicated and typically static dynamics. Several papers investigate the transition point when the designer should choose to use more complex models - the First Order Plus Dead Time (FOPDT) representing the first possible extension (Skogestad, 2003; Jones and Tham, 2004). Also Huba (2003) shows that for the relatively low ratio of the dead time and the plant time constant T_d / T_p it is enough to use the to Integrator Plus Dead Time (IPDT) approximations also for dealing with the FOPDT processes used in this paper. However, when using the IPDT approximation for the FOPDT process, the plant feedback that is around an operating point equivalent to a load disturbance will lead to assymmetrical behavior also in the case with symmetrical relay without additional load. So, in the relay identification this oscillation asymmetry is playing an important issue with respect to the precision of the whole approximation. For a noncompensated disturbance (including also the internal plant feedback around the operating point), the deformation of oscillations leads to increased influence of higher harmonics and to decreased precision of the identification both by using the describing functions method and the Fast Fourier Transform (FFT). The main advantage of constraining the plant approximation to the IPDT model

$$S(s) = K_s e^{-T_d s} / s \tag{1}$$

is that both the experiment setup and the corresponding formulas remain relatively simple and more robust against measurement noise than when using the FOPDT model. There is no need to tune the PI controller before the identification, or to use the PI controller with an additional anti windup circuitry.

Let us consider oscillations in the control loop with a relay with the output $u_r = \pm M$ and a piecewise constant input disturbance $v = const$. Then, the actual plant input will be given as a piecewise constant signal $u_A = \pm M + v$. Possible transients are shown in Fig. 2.

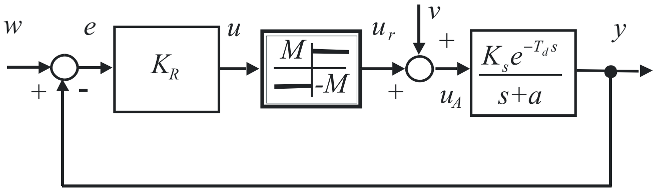


Fig. 1 Relay identification with nonsymmetrical plant input

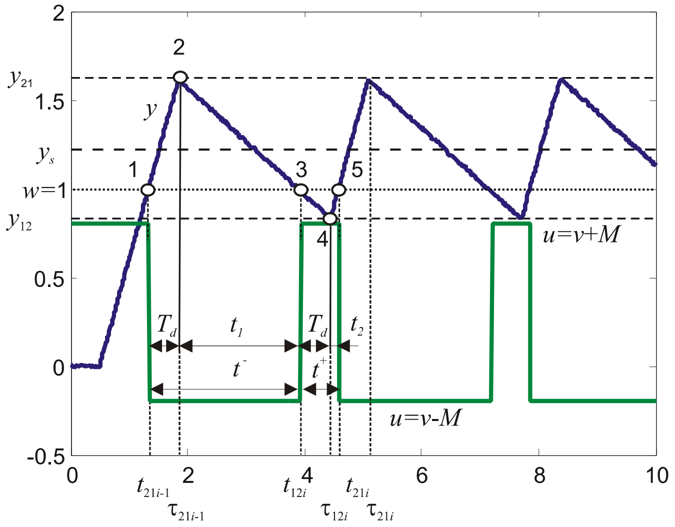


Fig. 2 Transients of basic variables of the loop in Fig. 1

By assuming relay switching from the positive relay output $u = M$ to the negative value $u = -M$ (point 1) at the time moment t_{21i-1}, due to the dead time the influence of the positive plant input $U_2 = (v+M)K_s$ will keep over interval with the length equal to the dead time value T_d.. Then, after reaching output value y_{21} at the time moment τ_{21i-1} (point 2) due to the effective plant input $U_1 = (v-M)K_s$ the output starts to decrease. After the time interval t_1 it reaches the reference value *w* (point 3). Even though at this moment the relay switches to the positive value $u = M$ the plant output continues to fall the time T_d longer and reaches the value y_{12} (point 4). The total length of the interval with negative relay output will be denoted as t^-. Under virtue of the positive relay output the plant output starts to rise and reaches the reference value after the time t_2 (point 5). The total duration of the positive relay output may be denoted as

$$t^+ = t_2 + T_d \tag{2}$$

As a result of the time delay, the plant output turnover time instants τ_{21i} are shifted with respect to the relay reversal moments t_{21i} by T_d. Similar time shift exists among time instants τ_{12i} and t_{12i}, i.e.

$$\tau_{21i} = t_{21i} + T_d \quad ; \quad \tau_{12i} = t_{12i} + T_d \tag{3}$$

For a single integrator it is possible to formulate relations

$$\begin{aligned} y_{21} - w = U_2 T_d \ ; \ t_1 = (w - y_{21})/U_1 \\ y_{12} - w = U_1 T_d \ ; \ t_2 = (w - y_{12})/U_2 \end{aligned} \tag{4}$$

Period of one cycle may be denoted as

$$P_u = t^+ + t^- = 2T_d + t_1 + t_2 = \frac{4T_d M^2}{M^2 - v^2} \tag{5}$$

For a known value of the relay amplitude M and a known ratio of the positive and negative relay output duration over one cycle

$$\varepsilon = \frac{t^+}{t^-} = \frac{t_2 + T_d}{t_1 + T_d} = -\frac{v - M}{v + M} \tag{6}$$

it is possible to express the identified disturbance as

$$v = u_0 + v_n \tag{7}$$

This may be composed of the known intentionally set offset at the relay output u_0 and an unknown external disturbance v_n that may be identified as

$$v = M\frac{1-\varepsilon}{1+\varepsilon} \tag{8}$$

From (5) it then follows

$$T_d = \frac{P_u}{4}\left[1 - \left(\frac{v}{M}\right)^2\right] = P_u \frac{\varepsilon}{(1+\varepsilon)^2} \tag{9}$$

The output mean value over one cycle period may be expressed as

$$\begin{aligned} y_s = \frac{1}{P_u}\Bigg[w + \int_0^{T_d} U_2 t dt + \int_{T_d}^{2T_d+t_1} (y_{21} + U_1(t - T_d))dt + \\ + \int_{2T_d+t_1}^{P_u} (y_{12} + U_2(t - 2T_d - t_1))dt \Bigg] \end{aligned} \tag{10}$$

$$\begin{aligned} y_s = \frac{1}{P_u}\Bigg[w + \int_0^{T_d} K_s(v+M)t dt + \\ + \int_{T_d}^{2T_d+t_1} (y_{21} - K_s(v-M)(t - T_d))dt + \\ + \int_{2T_d+t_1}^{P_u} (y_{12} + K_s(v+M)(t - 2T_d - t_1))dt \Bigg] \end{aligned} \tag{11}$$

Finally, one gets formula for the plant gain

$$K_s = \frac{(1+\varepsilon)^2}{\varepsilon P_u}\frac{(y_s - w)}{v} = \frac{(1+\varepsilon)^3}{\varepsilon(1-\varepsilon)P_u}\frac{(y_s - w)}{M} \tag{12a}$$

It is also possible to calculate the plant gain by using the area A limited by $y(t)$ around w over one period (5), when

$$K_s = \frac{(1+\varepsilon)^4}{\varepsilon\left(1+\varepsilon^2\right)}\frac{A}{MP_u^2} \tag{12b}$$

In difference to (12a), this may also be used in the symmetrical case with $v = 0$ and $y_s = w$. So, to get the model parameters (1) it is enough to calculate the mean plant output value over one cycle of relay switching (11), or the equivalent area A, the period of oscillation (5) and the ratio of time slots with positive and negative relay output (6). The approximation should remain valid also in the case of constant input disturbance $v = const$. This may be considered to be composed of the intentionally introduced disturbance u_0 and of the external disturbance v_n

$$v = u_0 + v_n \tag{13}$$

In this way it is possible to introduce an additional free parameter for tuning enabling to work in any working point with arbitrarily low relay module *M*.

After carrying out the above procedure at least for two different reference signal values w_1 and w_2 and by evaluating changes of the identified disturbance values v_1 and v_2 in dependence on the mean output values y_{s1} and y_{s2} it is then possible to approximate the dependence

$$v = f(y_s) \tag{14}$$

If it has a negligible slope with respect to changes in y_s, the system is sufficiently well approximated by the IPDT model.

2. REAL EXPERIMENT

The thermo-optical plant laboratory model (Fig.3) offers measurement of 8 process variables: controlled temperature, its filtered value, ambient temperature, controlled light intensity, its derivative and filtered value, the fan speed of rotation and current. The temperature and the light intensity control channels are interconnected by 3 manipulated voltage variables influencing the bulb (heat & light source), the light-diode (the light source) and the fan (the system cooling). Besides these, it is possible to adjust two parameters of the light intensity derivator. Within Matlab/Simulink or Scilab/Scicos schemes [10] the plant is represented as a single block and so limiting needs on costly and complicated software packages for real time control. The (supported) external converter cards are necessary just for sampling periods below 50ms. Currently, more than 40 such plants are used in labs of several EU universities.

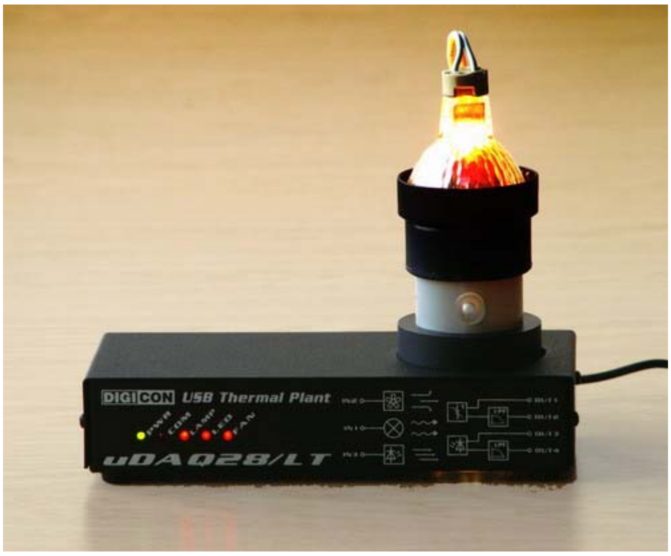


Fig. 3 Thermo-optical plant

The thermal plant consists of a halogen bulb 12V DC/20W, of a plastic pipe wall, of its internal air column containing the temperature sensor PT100, and of a fan 12V DC/0,6W (can be used for producing disturbances, but also for control).

The optical channel has two outputs. The non filtered light intensity measured by a photodiode and the filtered one, where the signal is filtered by an analogue low pass filter with time constant at about 20s.

The non-filtered light channel represents a very fast process which can be approximated as memoryless plant. In an ideal case static feedforward control with inverse process gain should be sufficient for such process. However the filtered optical channel was used for the experiments, where the analogue first order filter is used to filter the non-filtered light channel output. We analyzed the system parameters in several working points. The input of the system is the bulb voltage which is limited to 5V.

The following table shows the system parameters in all working points. Relay magnitude ranges from 3 to 5V and the setpoint (light intensity) ranges from 10 to 35.

Table 1. Average system parameters

w	M	Ks	Td	v
35	5	0,581038	0,5505	-2,89034
30	5	0,564979	0,57954	-2,52964
20	5	0,536201	0,582476	-1,77778
10	5	0,475311	0,562213	-0,98358
30	4	0,57706	0,472195	-2,64252
20	4	0,522255	0,508156	-1,86067
10	4	0,468589	0,522159	-1,01576
30	3	0,602765	0,389834	-2,78204
20	3	0,56064	0,442128	-2,03024
10	3	0,536333	0,448577	-1,2376

In Fig. 4, the measured and the approximated system output in one working point is compared.

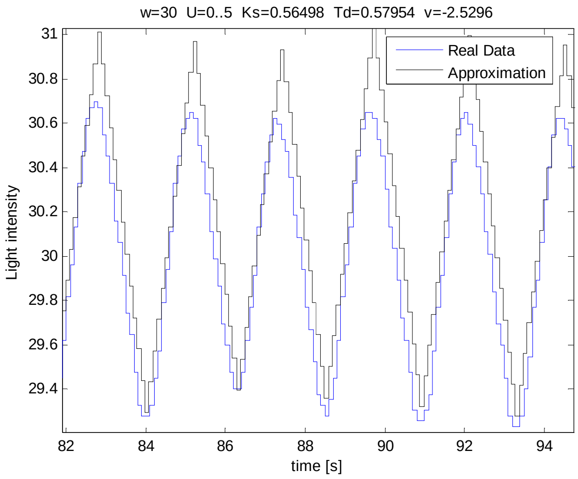


Fig. 4 Measurement and simulation comparison

3. PI CONTROLLER TUNING

PI controller was employed to control the plant to verify the identification results. To improve control performance the PI-controller structure from Fig. 5 was used.

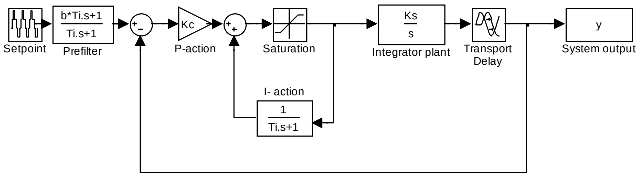


Fig. 5 PI1-controller

Performance portrait analysis was used to tune the controller. Upper left portrait shows the amount of an overshooting, the red area corresponds to controller tuning which yields overshooting up to 0.01%, the amount of overshooting grows to 5% in the blue area. The upper right portrait shows the control signal deviation from the defined shape. Lower portraits show the borders of the areas above.

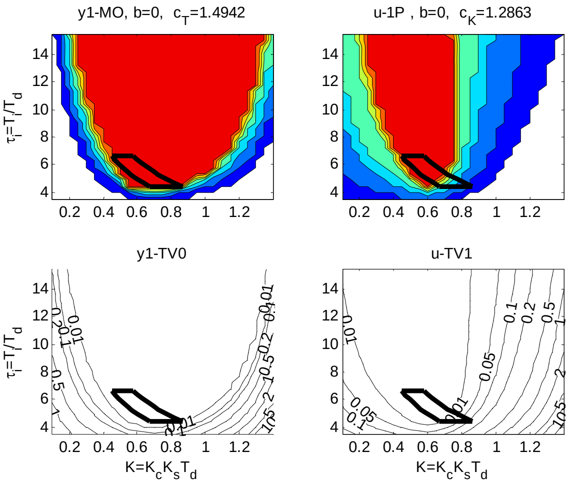


Fig. 6 Performance portrait

The following figures show the real experiment results for various setpoint changes. The controller was tuned to yield up to 0.1% overshooting.

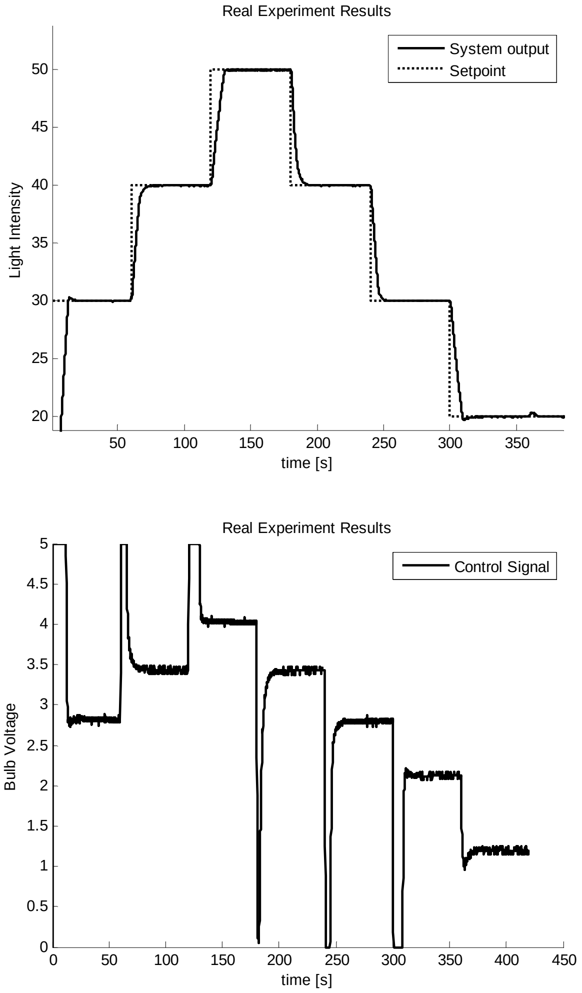


Fig.6 Control results

The control results in Fig. 6 show fast transients without overshooting except the last downward setpoitnt step, where a small overshoot occurs which results from the approximation imperfection. The control signal consists of two control phases: one can observe an interval at the saturation followed by the control signal's monotonic transition to the new steady state value.

4. CONCLUSION

New relay experiment identification method has been proposed for the IPDT plant. Stable optical plant with the first order dominant dynamics was used for illustrating and verifying the method by the real experiment. The method benefits from obtaining the load disturbance value without need of tuning a PI controller firstly. Sensitivity to the measurement noise that may lead to more complicated relay output than the considered period consisting of two pulses, can be at least partially eliminated by sampled-data relay control using longer sampling periods.

In applying the proposed method to controlling optical plant, the relay test yields results depending on the working point that obviously points out on nonlinear plant behaviour. In this paper, the nonlinear properties were treated by a robust controller tuning. One of the strong advantages of the proposed method, however, is its possible extension to identifying FOPDT model, or a nonlinear model with dominant first order dynamics + dead time. Nevertheless, due to the simple analytical formulas the proposed algorithm is easy to implement online.

ACKNOWLEDGEMENTS

This work was partially supported by the Project VEGA 1/0656/09: Integration and development of nonlinear and robust control methods and their application in controlling flying vehicles, by the project KEGA 3/7245/09 Building virtual and remote experiments for network of online laboratories. It was also supported by the grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Hang, C. C., Aström, K. J., & Ho, W. K. (1993). Relay auto-tuning in the present of the static load disturbance. *Automatica*, 29, 563.

Huba, M.(2003). *Constrained systems design. Vol.1 Basic controllers*. STU Bratislava (in Slovak).

Huba, M. (2006). Constrained pole assignment control, *In: Current Trends in Nonlinear Systems and Control*, L. Menini, L. Zaccarian, Ch. T. Abdallah, Edts., Boston: Birkhäuser, 163-183.

Huba, M., Šimunek, M. (2007). Modular Approach to Teaching PID Control. *IEEE Trans. Ind. Electr.*, 54, 6, 3112-3121.

Huba, M. (2010). Robust Constrained PID Control. *Int. Conference Cybernetics and Informatics*, Vysna Boca, Slovakia.

Jones, R. W., Tham, M. T. (2004).: Gain and Phase Margin Controller Tuning: FOPTD or IPDT moel-based methods?, *In.: Annual Conference in Sapporo*, August 4-6, 2004

Park, J. H., Sung, S. W., & Lee, I. (1997). Improved relay auto-tuning with static load disturbance. *Automatica*, 33, 711.

Shen, S., Wu, J., & Yu, C. (1996). Autotune identification under load disturbance. Industrial Engineering and Chemical Research, 35, 1642.

Skogestad, S. (2003).: Simple Analytic Rules for Model Reduction and PID Controller tuning, *Journal of process Control*, Vol 13, pp. 291-309

Sung, S.W., & Lee, J. (2006) Relay feedback method under large static disturbances, Automatica, 42, 353-356

Tao Liu, Furong Gao (2009).: A generalized relay identification method for time delay and non-minimum phase processes. *In. Automatica 45* (2009) 1072_1079

Ziegler, J.G., Nichols, N.B. , (1942) Optimum settings for automatic controllers, *Trans. ASME 64* 759–768.

Real-time Air/Fuel Ratio Model Predictive Control of a Spark Ignition Engine

Matúš Kopačka, Peter Šimončič, Jozef Csambál, Marek Honek, Sławomir Wojnar, Tomáš Polóni, Boris Rohaľ-Ilkiv

*Faculty of Mechanical Engineering, Slovak University of Technology,
Bratislava, Slovakia
(e-mail: matus.kopacka@stuba.sk)*

Abstract: The following paper describes the control of air/fuel ratio (AFR) of a spark ignition engine utilizing the analytical model predictive controller based on the multi-model approach. The multi-model approach employs the autoregressive model (ARX) network, using the weighting of local models, coming from the sugeno-type fuzzy logic. The weighted ARX models are identified in the particular working points and are creating a global engine model, covering its nonlinearity. Awaited improvement of a proper air/fuel mixture combusted in a cylinder is mostly gained in the transient working regimes of an engine. In these regimes, the traditional control approach looses its quality, compared to steady state working regimes of an engine. This leads to higher fuel consumption and level of emissions from an engine. Presented results of the air/fuel ratio control are acquired from the real-time control of the VW Polo $1390cm^3$ engine, at which the original electronic control unit (ECU) has been replaced by a dSpace system executing the model predictive controller. It has been proven, that the proposed controller is suitable for the air/fuel ratio control giving sufficiently good and steady system output.

Keywords: model predictive control, analytical solution, air/fuel ratio, SI engine, ARX models

1. INTRODUCTION

A run of a spark ignition engine (SI) is highly dependent on the mixture of the sucked air and injected fuel present in the cylinder, waiting to be ignited by the spark. Incorrect ratio of this two components may lead to the poor engine power, ineffective functionality of the catalytic converter resulting in higher level of emissions polluting the environment and in the extreme case this can lead to the engine stoppage. Due to this reason it is crucial to keep the air/fuel ratio (AFR) at the stoichiometric level, which means, that both, the air and the fuel are completely combusted. Due to above mentioned reasons and all the time tightening emission standards the car producers are improving the control of the air/fuel ratio.

Traditional control of air/fuel ratio is based on a feedforward control using predefined tables determining how much fuel has to be injected into a cylinder, based on the information from the mass air flow meter. This fuel amount is subsequently corrected using the information from the lambda probe, so the stoichiometric mixture can be reached. Due to a lambda probe position (at the engine exhaust) a delay arises, causing an improper feedback correction at the unstable engine regimes, like acceleration, or deceleration. On the other side, this kind of control guarantees stability and robustness at all conditions and therefore is still preferred by car producers, despite its disadvantages in control.

The academic field have started to publish other kinds of air/fuel control, mostly model-based ones. The model-based approaches are bringing good quality of control, but are also more sensitive to the model precision and issues with stability and robustness appear. A survey through popular "mean value engine modeling" is described in Bengtsson et al. (2007). This analytical way of engine modeling is very clear, but requires exact knowledge of the system and the model error has to be taken into account explicitly. Other ways of a model acquisition are based on the experimental identification (black box modeling). Works of Zhai et al. (2010), Zhai and Yu (2009) and Hou (2007) are specialized in employment of neural networks, while Mao et al. (2009) uses for engine modeling CARIMA models.

In the engine control itself became popular fuzzy logic (Hou (2007)), neural network control (Arsie et al. (2008)) and model predictive control (MPC) approaches (Lorini et al. (2006) and Muske and Jones (2006)). General topics on an issue of stability and robustness in MPC can be found in Mayne et al. (2000), or Zeman and Rohal-Ilkiv (2003).

Our approach, introduced in Polóni et al. (2007) is utilizing an analytical model predictive controller with a penalization of a terminal state. It uses a multi-model approach using a weighted net (sugeno-type fuzzy logic) of autoregressive models (ARX) as a system model. The ARX models were identified in the particular working points of the engine as black box models. This method of engine modeling offers an easy way of "global nonlinear system model" acquisition with subsequent utilization in the model based system control. The preliminary real-time predictive control results presented in this paper indicate that the proposed controller could be suitable alternative toward the air/fuel ratio control through the look-up tables.

2. AIR/FUEL RATIO

The model of the air/fuel ratio dynamics λ of a spark ignition engine is based on the mixture, defined as a mass ratio of the air and fuel present in a cylinder at a time instance k. Due to the fact, that the air mass flow is measured as an absolute value, it was necessary to integrate this amount during the particular time and express the air and fuel quantity as relative mass densities $(\frac{grams/cylinder}{grams/cylinder})$. Hence, the air/fuel ratio is defined, as:

$$\lambda(k) = \frac{m_a(k)}{m_f(k)} \frac{1}{L_{th}} \tag{1}$$

Where $m_a(k)$ and $m_f(k)$ are relative mass amounts of air and fuel in a cylinder and $L_{th} \approx 14.64$ is the theoretical amount of air necessary for the ideal combustion of a unit amount of fuel. The L_{th} constant normalizes the ideal value of λ to be 1.0.

3. SI ENGINE MODELING USING ARX MODELS

The engine modeling is based on the weighted linear local model with single input single output (SISO) structure (Polóni et al., 2008). The parameters of local linear ARX models with weighted validity (Murray-Smith and Johanssen, 1997) are identified to model the nonlinear dynamics of the AFR. The principle of this nonlinear modeling technique is in partitioning of the engine's working range into smaller working points.

A net of local ARX models weighted for a particular working point ϕ is defined, as:

$$\begin{aligned}&\sum_{h=1}^{n_M} \rho_h(\phi(k)) A_h(q) y(k) = \\ &\sum_{h=1}^{n_M} \rho_h(\phi(k)) B_h(q) u(k) + \sum_{h=1}^{n_M} \rho_h(\phi(k)) c_h + e(k)\end{aligned} \tag{2}$$

defined by polynomials A_h and B_h:

$$\begin{aligned} A_h(q) &= 1 + a_{h,1} q^{-1} + \ldots + a_{h,n_y} q^{-ny} \\ B_h(q) &= b_{h,1+d_h} q^{-1-d_h} + \ldots + b_{h,n_u+d_h} q^{-nu-d_h} \end{aligned} \tag{3}$$

where symbolics q^{-i} denotes a sample delay, e.x. $q^{-i}y(k) = y(k-i)$, $a_{h,i}$ and $b_{h,(j+d_h)}$ are parameters of hth local function and d_h is its delay. Parameter n_M represents the number of local models.

The ρ_h denotes a weighting function of a particular ARX model (see Sec. 3.1) and the $e(k)$ is a stochastic term with a white noise properties. The engine working point itself is defined by engine revolutions n_{en} and the throttle valve position t_r, hence: $\phi(k) = [n_{en}(k), t_r(k)]^T$. The absolute term $\hat{c}_h$ of the equation is computed from the steady state values of the system output $y_{e,h}$ and the system input $u_{e,h}$, as:

$$\hat{c}_h = y_{e,h} + y_{e,h} \sum_{i=1}^{ny} \hat{a}_{h,i} - u_{e,h} \sum_{j=1}^{nu} \hat{b}_{h,j} \tag{4}$$

The model output is computed from the equation:

$$\begin{aligned} y_s(k) = &\sum_{h=1}^{n_M} \rho_h(\phi(k)) \\ &\cdot \left(\sum_{i=1}^{ny} \hat{a}_{h,i} q^{-i} y_s(k) + \sum_{j=1}^{nu} \hat{b}_{h,(j+d_h)} q^{-j-d_h} u(k) + \hat{c}_h \right) \end{aligned} \tag{5}$$

which after the introduction of the estimated parameter vector $\hat{\theta}_h$ and the regression vector $\gamma(k)$, becomes:

$$y_s(k) = \gamma^T(k) \sum_{h=1}^{n_M} \rho_h(\phi(k)) \hat{\theta}_h + \sum_{h=1}^{n_M} \rho_h(\phi(k)) \hat{c}_h \tag{6}$$

3.1 Weighting functions

The full working range of the engine has been covered by a discrete amount of local linear models (LLMs), identified at particular working points. The LLMs are being weighted by a weighting functions defining validity of each local model according to an instantaneous working point of the engine. Due to a request of a smooth and continuous global engine model, design of those weighting functions was crucial.

There were designed particular interpolation functions for every LLM, assigning it 100% validity exactly at the belonging working point with a decreasing tendency in the directions of the deviation of the throttle valve opening Δt_r and the engine revolutions Δn_{en} from the particular working point. The "three dimensional" Gaussian functions:

$$\begin{aligned}&\tilde{\rho}_h(\phi(k)) = \\ &\exp\left[-\left[\, \Delta n_{en}(k) \;\; \Delta t_r(k) \,\right] \begin{bmatrix} \frac{1}{\sigma_{h,1}^2} & 0 \\ 0 & \frac{1}{\sigma_{h,2}^2} \end{bmatrix} \begin{bmatrix} \Delta n_{en}(k) \\ \Delta t_r(k) \end{bmatrix} \right]\end{aligned} \tag{7}$$

were used as the local weighting functions, due to their suitable shape fulfilling the approximation properties. The choice of tuning parameters $\sigma_{h,1} = 250$ and $\sigma_{h,2} = 0.8$ used in the weighting functions has been chosen experimentally, awaiting continuous and smooth output of the modeled system. At the same time the experiments have shown, that there can be used identical weighting functions for weighting of the air and fuel path parameters.

All the weighting functions were at the end normalized by creating normalized weighting functions:

$$\rho_h(\phi(k)) = \frac{\tilde{\rho_h}(\phi(k))}{\sum_{h=1}^{n_M} \tilde{\rho_h}(\phi(k))} \tag{8}$$

so the sum of values of all weighting functions belonging to a particular working point (Fig. 1), equals exactly one: $\sum_{h=1}^{n_M} \rho_h(\phi(k)) = 1$.

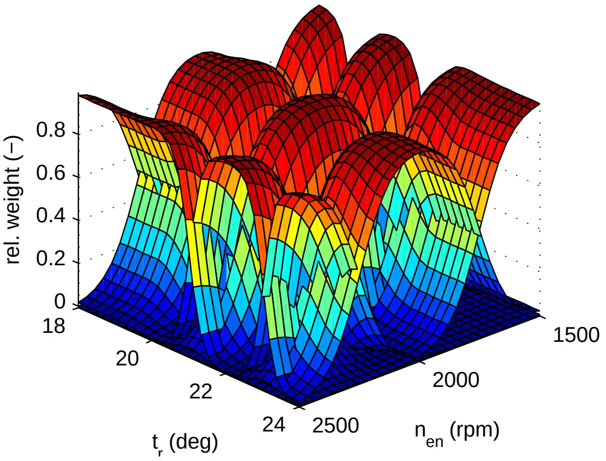


Fig. 1. Relative weighting Gaussian functions

3.2 Model identification

Considering the $\lambda(k)$ modeling, the engine has been divided into two subsystems with independent inputs, namely into:

air path with the air throttle position as the disturbance input, and
fuel path with the input of fuel injector opening time.

Another disturbance-like acting quantity in the air path were engine revolutions, implicitly included in the engine model, particularly for each working point.

Parameters of the local ARX models have been estimated from the data acquired from the exhaust gas oxygen sensor and an air flow sensor. The identification has been designed so, that the dynamics of the air path and fuel path stayed uncoupled, hence the dynamics of both paths were measured indirectly.

Air path identification The first experiment started at the stoichiometric value of λ_a in the operation point ϕ. To excite the air path dynamics, the throttle valve position was oscillating around its steady position according to a pseudo-random binary signal (PRBS), while the fuel injectors were delivering constant fuel mass $m_{f,e}$. The change in λ_a value has been recorded. During the experiment the engine had been braked at constant revolutions.

Fuel path identification The identification of the fuel path dynamics has been done similarly, but with the fixed throttle valve delivering a constant air mass $m_{a,e}$. The PRBS was varying the fuel injectors' opening time and the value of λ_f had been measured again.

In both experiments it was necessary to wisely propose a PRBS, so that the air/fuel mixture is always ignitable. The local ARX models can be subsequently determined from the measured values of instantaneous $\lambda_a(k)$ and $\lambda_f(k)$ belonging to the air path and fuel path, utilizing relative air and fuel mass densities:

$$m_a(k) = m_{a,e}(\phi)\lambda_a(k) \tag{9}$$

and

$$m_f(k) = \frac{m_{f,e}(\phi)}{\lambda_f(k)} \tag{10}$$

The final formula describing the aif/fuel ratio dynamics is built up of local linear ARX models of the air and fuel paths is in the form:

$$\lambda_s(k) = \frac{1}{L_{th}} \cdot \left[\frac{\gamma_a^T(k) \sum_{h=1}^{n_A} \rho_{a,h}(\phi(k))\hat{\theta}_{a,h} + \sum_{h=1}^{n_A} \rho_{a,h}(\phi(k))\hat{c}_{a,h}}{\gamma_f^T(k) \sum_{h=1}^{n_F} \rho_{f,h}(\phi(k))\hat{\theta}_{f,h} + \sum_{h=1}^{n_F} \rho_{f,h}(\phi(k))\hat{c}_{f,h}} \right] \tag{11}$$

Where:

γ	is the regression vector of system inputs and outputs
n_A	is the amount of working points
ρ	is the interpolation function
ϕ	is the vector of a working point
θ	is the vector of ARX parameters
c	is the absolute term of an ARX model

In accordance with the general model structure presented, the key variables are defined in the Table 1.

Table 1. Symbol connection between the general expression and the model

general symbol	air-path model	fuel-path model	operating point
$y(k)$	$m_a(k)$	$m_f(k)$	
$u(k)$	$t_r(k)$	$u_f(k)$	
$\gamma(k)$	$\gamma_a(k)$	$\gamma_f(k)$	
$\hat{\theta}_h$	$\hat{\theta}_{a,h}$	$\hat{\theta}_{f,h}$	
$\rho_h(\phi(k))$	$\rho_{a,h}(\phi(k))$	$\rho_{f,h}(\phi(k))$	
$\hat{c}_h$	$\hat{c}_{a,h}$	$\hat{c}_{f,h}$	
$\phi(k)$			$[n_e(k), t_r(k-\delta)]^T$

4. PREDICTIVE CONTROL

The strategy of an "exceeding oxygen amount" control using a predictive controller is based on a prediction of a controlled quantity λ and subsequent minimization of a chosen cost function on the horizon N_p expressed in a standard quadratic form. The value of λ is predicted by utilization of partially linear models of the air and fuel path. Through the independent air path model the proper amount of fuel is predicted and enters the cost function J. Hence, the target of the cost function minimization is to determine such a control law, that the measured system output λ is stoichiometric. The second modeled subsystem, the fuel-path, is an explicit component of the objective function where the amount of the fuel is the function of optimized control action (Polóni et al. (2008)).

4.1 Predictive model

The applied control strategy is based on the knowledge of the internal model (IM) of air-path, predicting the change of air flow through the exhaust pipe, and consequently, setting the profile of desired values of the objective function on the control horizon. In this case we will consider the state space (SS) formulation of the system and therefore it is necessary to express linear local ARX models in the SS structure with time varying parameters:

$$\begin{aligned} x_{(a,f)}(k+1) &= A_{(a,f)}(\phi)x_{(a,f)}(k) + B_{(a,f)}(\phi)u_{(a,f)}(k) \\ m_{s,(a,f)}(k) &= C_{(a,f)}x_{(a,f)}(k) \end{aligned} \tag{12}$$

The weighted parameters of multi-ARX models are displayed in matrices $A_{a,f}$ and $B_{a,f}$ for both subsystems. This is a non-minimal SS representation whose advantage is, that no state observer is needed. The "fuel pulse width control" is tracking the air mass changing on a prediction horizon from IM of the air-path, by changing the amount of injected fuel mass. Due to tracking offset elimination, the SS model of the fuel-path (12) (index f), with its state space vector x_f, is written in augmented SS model form to incorporate the integral action:

$$\tilde{x}_f(k+1) = \tilde{A}_f(\phi)\tilde{x}_f(k) + \tilde{B}_f(\phi)\Delta u_f(k) \tag{13}$$

or

$$\begin{bmatrix} x_f(k+1) \\ u_f(k) \end{bmatrix} = \begin{bmatrix} A_f(\phi) & B_f(\phi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_f(k) \\ u_f(k-1) \end{bmatrix} + \\ + \begin{bmatrix} B_f(\phi) \\ 1 \end{bmatrix} \Delta u_f(k)$$

$$m_{s,f}(k) = \tilde{C}_f\tilde{x}_f(k) + D_f\Delta u_f(k) \tag{14}$$

or

$$m_{s,f}(k) = [\, C_f \;\; D_f \,]\, \tilde{x}_f(k) + D_f\Delta u_f(k)$$

The prediction of the air mass ($\underrightarrow{m}_a$) on the prediction horizon (N_p) is dependent on the throttle position ($\underrightarrow{t}_r$) and is computed as

$$\underrightarrow{m}_a(k) = \Gamma_a(\phi)x_a(k) + \Omega_a(\phi)\underrightarrow{t}_r(k-1) \tag{15}$$

where the x_a denotes the state space vector of the air path.

Due to the unprecise modeling (IM strategy), the biased predictions of the air mass future trajectory and consequently biased fuel mass might occur. This error is compensated incorporation the term $L[\hat{m}_f(k) - m_{s,f}(k)]$ into the fuel mass prediction equation:

$$\begin{aligned} \underrightarrow{m}_f(k) \;&=\; \Gamma_f(\phi)\tilde{x}_f(k) + \Omega_f(\phi)\Delta\underrightarrow{u}_f(k-1) + \\ &+\; L[\hat{m}_f(k) - m_{s,f}(k)] \end{aligned} \tag{16}$$

The matrices of free response Γ_a, Γ_f and forced response Ω_a, Ω_f are computed from the SS model (12), respectively (Maciejowski, 2000). Since there is only $\lambda(k)$ measurable in equation (1), the value of $m_a(k)$ needs to be substituted using IM of the air-path, then:

$$\hat{m}_f(k) = \frac{1}{L_{th}}\frac{m_{s,a}(k)}{\lambda(k)} \tag{17}$$

The estimate $\hat{m}_f(k)$ is used to compensate for possible bias errors of predicted $\underrightarrow{m}_f(k)$ in (16).

4.2 Analytical solution

The analytical solution is based on the cost function (18), encompassing deviations of predicted fuel mass amounts between the air and fuel path (based on (1)); a penalization of control increments r; and a penalization p of a deviation between a predicted and desired end state.

$$\begin{aligned} J_\lambda = &\left\| \frac{\underrightarrow{m}_a(k)}{L_{th}} - \underrightarrow{m}_f(k) \right\|_2^2 + r\|\Delta\underrightarrow{u}_f(k-1)\|_2^2 \\ &+ p\|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2 \end{aligned} \tag{18}$$

The chosen MPC approach utilizes the state space representation with an integral control for the correction of the prediction.

Due to a disturbance $d(k)$, the steady state values of u and x have to be adapted so, that the assumption $J = 0$ is valid. This problem solves an explicit inclusion of the disturbance into the model.

The fuel injectors are controlled by a fuel pulse width, what is at the same time the control u_f. The optimal injection time can be computed by minimization of a cost function (18), which has after expansion by the fuel path prediction equation, form:

$$\begin{aligned} J_\lambda = &\left\| \frac{\underrightarrow{m}_a}{L_{th}} - \Gamma_f\tilde{x}_f(k) + \Omega_f\Delta\underrightarrow{u}_f(k-1) \right\|_2^2 \\ &+ r\left\|\Delta\underrightarrow{u}_f(k-1)\right\|_2^2 + p\|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2 \end{aligned} \tag{19}$$

An analytical solution of $\frac{dJ_\lambda}{\Delta\underrightarrow{u}} = 0$ of (19) without constraints leads to an expression determining the change of "fuel injector opening time" in a step (k), as:

$$\begin{aligned} &\Delta u = \left(\Omega^T\Omega + Ir + p\Omega_{xN}^T\Omega_{xN}\right)^{-1} \\ &\cdot \left[\Omega^T[w(k) - \Gamma\tilde{x}(k)] - p\Omega_{xN}^T A^N\tilde{x}(k) + p\Omega_{xN}^T\tilde{x}_{f,r}(N)\right] \end{aligned} \tag{20}$$

Hence, the absolute value of the control action in a step k is given by a sum of a newly computed increment in a control (20) and an absolute value of the control in a step $(k-1)$:

$$u_f(k) = u_f(k-1) + \Delta u_f(k) \tag{21}$$

5. RAPID CONTROL PROTOTYPING SYSTEM

The computational unit necessary for the real-time implementation of the MPC control is based on a powerful and freely programmable control system based on *dSpace* and *RapidPro* units; or "Rapid Control Prototyping System" (RCP), (Fig. 2, dSPACE GmbH. (2009)). It is built-up on the processor board *ds1005* and hardware-in-loop platform *dS2202 HIL*. The RCP ensures sufficient headroom for the real-time execution of complex algorithms (Arsie et al. (2008)) and lets all engine tasks to be controlled directly. Also, the customized variants of the controller can be performed immediately.

Typical RCP system consists of:

- A math modeling program (prepared in Simulink)
- Symbolic input/output blocks
- A real-time target computer (embedded computer with an analog and digital I/O)

- A host PC with communication links to target computer
- A graphical user interface (GUI) which enables to control the real time process

The RCP system enables to use a support in the form of embedded functions which make the preparation of algorithms easy and fast. It is a great help, because one can then concentrate on significant problems (development and debugging of algorithms) without the spending time on not so important tasks (how to handle features of RCP system at low level programming).

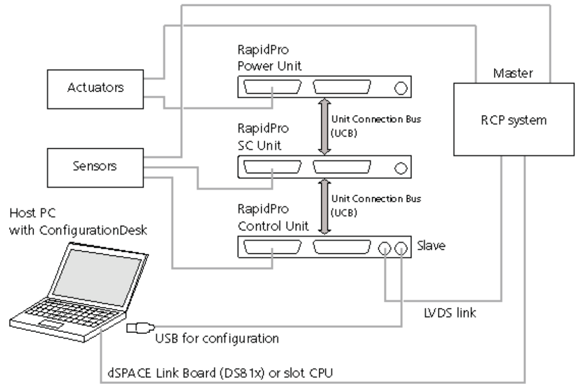


Fig. 2. Rapid control prototyping scheme

6. REAL-TIME APPLICATION OF A PREDICTIVE CONTROL

The ability to control the mixture concentration at stoichiometric level using MPC is demonstrated through the real-time SI engine control (Fig. 3). This has been performed using the AFR predictive control strategy described in the previous section, designed in *Matlab/Simulink* environment and compiled as a real-time application for a *dSpace* platform. It has been applied to the VW Polo engine (Fig. 4), 1390 cm^3 with 55kW@5000 rpm, not equipped with a turbocharger or an exhaust gas recirculation system. The control period was 0.2s. The result of an identification are 9 local linear models (LLM) for each, air and fuel path, dependent on a throttle valve opening and engine revolutions.

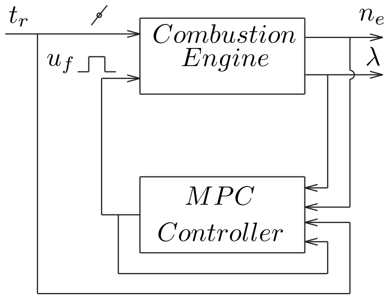


Fig. 3. Control scheme

The primary target of a control (Fig. 5) was to hold the air/fuel ratio in a stoichiometric region ($\lambda = 1$), in the worst case to keep the mixture ignitable ($0.7 \leq \lambda \leq 1.2$). During the experiment, the change in throttle valve opening, between 21 and 22 degrees (Fig. 5, variable t_r) and the change of engine revolutions (Fig. 5, variable n_{en}), has been performed several times. These changes simulate varying working regimes of an engine, which is adapting its run to a daily traffic. Changes in t_r and n_{en} quantities are determining the engine load, at the same time, ensuring, that the engine passes through several working points during its operation. As mentioned in Section 3, the engine revolutions are not included among explicit variables of local models, but they build together with a delayed throttle valve position a vector of an working point $\phi(k)$.

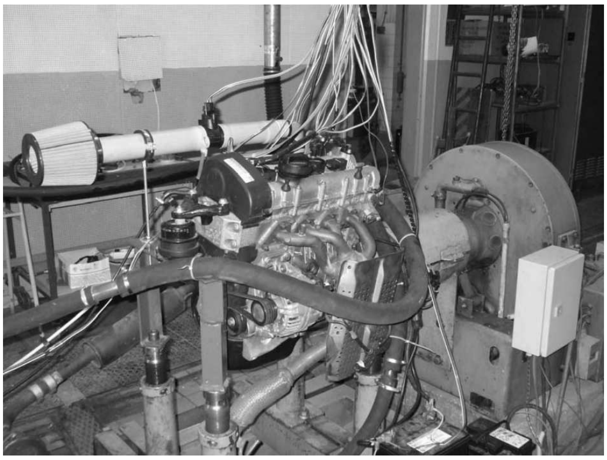


Fig. 4. Spark ignition engine VW Polo 1.4

The quality of control is sufficient (Fig. 5, variable λ), with exceptional acceptable overshoots in both directions. These overshoots of the controlled variable λ have been caused by smaller model precision, due to its distance from the working point, at which the system identification has been performed. This effect is caused by the approximation of a particular model from the other working points' models.
The corresponding control (fuel injection time) computed by the controller is shown in (Fig. 5, variable t_{inj}).

The initial engine warm-up (to 80 °C) eliminated model-plant mismatch caused by temperature dependent behavior of the engine.

The control has been performed by choosing the penalization $r = 0.1$. Utilizing the member $p\|\tilde{x}_f(N) - \tilde{x}_{f,r}(N)\|_2^2$ of a cost function by setting $p = 1.0$ allowed us to shorten the control horizon to $N_p = 20$ what significantly unloaded the computational unit and stabilized the controlled output of the engine on this shortened horizon, as well. The best control has been achieved in the neighborhood of working points, what is logically connected to the most precise engine model at those points. In other working points the control is still good enough, with small deviations from the stoichiometric mixture.

7. CONCLUSION

Considering the preliminary results from the real-time experiments at the engine, it can be concluded, that the idea of the AFR model predictive control based on local ARX models is suitable and applicable for the SI engine control. The proposed flexible design of a predictive controller offers easy tuning possibilities and a potential for

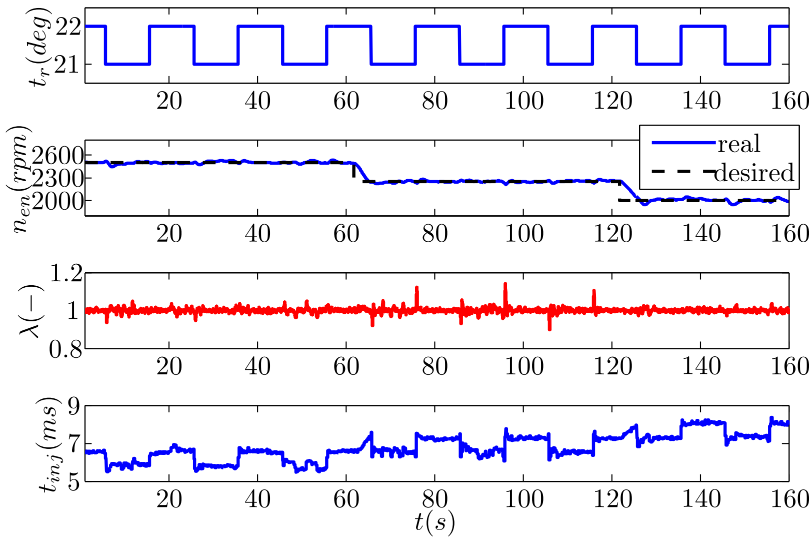


Fig. 5. Results of an AFR SI engine control

the model accuracy improvement by the extension of the global engine model to other working regimes of the engine. The next project step shall be the overshoot elimination in the λ - control by the identification of wider net of "local linear engine models" and implementation of constraints. Another task which has to be done is a comparison of the quality of control gained by the MPC controller with a baseline electronic control unit. This goal has been not yet achieved, as the original ECU has been replaced by the dSpace system running our controller.

ACKNOWLEDGMENTS

The work has been supported by the Slovak Research and Development Agency under grant LPP-0075-09, LPP-0118-09 and LPP-0096-07. This research is also supported by the grant from Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic. This support is very gratefully acknowledged.

REFERENCES

Arsie, I., Iorio, S.D., Noschese, G., Pianese, C., and Sorrentino, M. (2008). Optimal air-fuel ratio. *dSpace Magazine*, (1), 20–23.

Bengtsson, J., Strandh, P., Johansson, R., Tunestal, P., and Johansson, B. (2007). Hybrid modeling of homogenous charge compression ignition (HCCI) engine dynamics -a survey. *International journal of control*, 80(11), 1814–1847.

dSPACE GmbH. (2009). *HelpDesk Application*.

Hou, Z. (2007). Air fuel ratio control for gasoline engine using neural network multi-step predictive model. 3rd international conference on intelligent computing, Qingdao, China.

Lorini, G., Miotti, A., and Scattolini, R. (2006). Modeling, simulation and predictive control of a spark ignition engine. In Predimot (ed.), *Predictive control of combustion engines*, 39–55. TRAUNER Druck GmbH & CoKG.

Maciejowski, J.M. (2000). *Predictive control with constraints*. University of Cambridge.

Mao, X., Wang, D., Xiao, W., Liu, Z., Wang, J., and Tang, H. (2009). Lean limit and emissions improvement for a spark-ignited natural gas engine using a generalized predictive control (GPC)-based air/fuel ratio controller. *Energy & Fuels*, (23), 6026–6032.

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, (36), 789–814.

Murray-Smith, R. and Johanssen, T.A. (1997). *Multiple model approaches to modelling and control.* Taylor & Francis.

Muske, K.R. and Jones, J.C.P. (2006). A model-based SI engine air fuel ratio controller. American Control Conference, Minneapolis, USA.

Polóni, T., Johansen, T.A., and Rohaľ-Ilkiv, B. (2008). Identification and modeling of air-fuel ratio dynamics of a gasoline combustion engine with weighted arx model network. *Transaction of the ASME (Journal of Dynamic Systems, Measurement, and Control)*, 130(6). 061009.

Polóni, T., Rohaľ-Ilkiv, B., and Johansen, T.A. (2007). Multiple ARX model-based air-fuel ratio predictive control for SI engines. In *IFAC Workshop on advanced fuzzy and neural control.* Valenciennes, France. Conference paper MO5-3.

Zeman, J. and Rohal-Ilkiv, B. (2003). Robust minmax model predictive control of linear systems with constraints. 930 – 935. IEEE International Conference on Industrial Technology.

Zhai, Y.J., Ding-WenYu, Hong-YuGuo, and D.L.Yu (2010). Robust air/fuel ratio control with adaptive DRNN model and AD tuning. *Engineering Applications of Artificial Intelligence*, (23), 283–289.

Zhai, Y.J. and Yu, D.L. (2009). Neural network model-based automotive engine air/fuel ratio control and robustness evaluation. *Engineering Applications of Artificial Intelligence*, (22), 171–180.

Wind turbine power control for coordinated control of wind farms

Vedrana Spudić Mate Jelavić Mato Baotić

Faculty of Electrical Engineering and Computing
University of Zagreb, Croatia
e-mail: vedrana.spudic@fer.hr, mate.jelavic@fer.hr, mato.baotic@fer.hr

Abstract: The new grid regulations require that a grid-connected wind farm acts as a single controllable power producer. To meet this requirement a traditional wind farm control structure, which allowed individual wind turbines to internally define their power production, needs to be modified. In this paper the opportunity for wind turbine load reduction that arises from dynamic power control of wind turbines is studied. The wind farm controller design is proposed that utilizes coordinated power control of all wind turbines to achieve the wind farm regulation requirements and to minimize the wind turbine loads.

Keywords: Wind turbine control, Wind farm control, Model predictive control, Structural Loads

1. INTRODUCTION

With the increasing exploitation of wind energy, wind farms are growing both in number and in size and quickly becoming significant contributors in production of electrical energy. Consequently, the requirement emerges for large wind farms to function as a single controllable entity on the power grid, much like conventional power plants, see e.g. Elkraft System and Eltra (2004). For example, wind farm may be required to track the power reference provided by the Transmission System Operator (TSO) or to reduce the power production in order to contribute to the grid frequency regulation.

Traditionally, wind farm is operated as a collection of individually controlled wind turbines. Due to the new control requirements, however, the wind farm controller needs to take into account the interaction of wind turbines. The wind farm controller receives the wind farm power reference (or the wind farm regulation requirement, which can be readily expressed as the wind farm power reference, see e.g. Hansen et al. (2006)) from the TSO and distributes the individual wind turbine power references, see Figure 1. The wind farm controller uses the measurements from the wind farm as feedback. The sampling time for the wind farm controller has the order of 1 second.

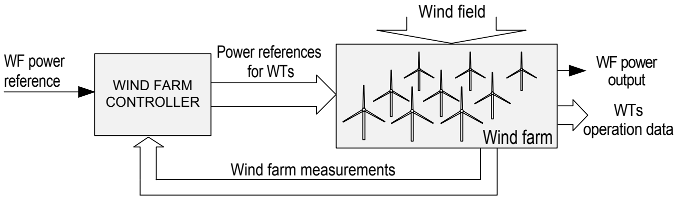


Fig. 1. Wind farm control system setup

A modern variable-speed wind turbine needs to be actively controlled to be operable. The state-of-the-art wind turbine control system has the ability to receive an external power reference. In this paper we study the behavior of the wind turbine with respect to the provided wind turbine power reference. The aim is to assess the potential for improving wind turbine operation by the appropriate wind farm controller design. The interest for this issue is spurred by the new wind farm control requirements. Namely, if the wind farm is to track a wind farm power reference then that power reference must be lower than the power available from the wind (the estimation of available wind farm power is used to determine the wind farm power reference, see Sørensen et al. (2005)). Therefore, the wind turbines are not necessarily producing all the available power (as it is typically the case in the wind farms). In this paper we study the idea of utilizing this power surplus to improve wind turbine dynamic operation. To the best of the authors knowledge this problem has not been tackled in wind energy literature.

The wind turbine considered in this paper is a conventional horizontal-axis three-bladed upwind variable-speed wind turbine with a blade-pitch-to-feather control system. This control system uses the increase in pitch angle to reduce the angle of attack of the blade and thus reduce lift force and the rotor torque. This is the current state-of-the-art in wind turbine technology. For simulations we use the MATLAB implementation (Soltani et al. (2010)) of a 5MW reference wind turbine model for offshore system development developed at National Renewable Energy Laboratory and described in Jonkman et al. (2009).

The paper is structured as follows:
Section 2 tackles the problem of defining a practical (but also justified) cost function for validation of wind turbine operation. Section 3 demonstrates and discusses the possibility for improvement of wind turbine dynamic behavior by adapting the power reference. In Section 4 the wind farm control system that utilizes the demonstrated benefits is proposed. Section 5 concludes the paper. For a brief overview of wind turbine operation and the description of

wind turbine control design model used in the paper the reader is referred to Spudić et al. (2010).

2. WIND FARM CONTROL OBJECTIVES

The primary wind farm control objective is that the wind farm electrical power output tracks the provided wind farm power reference.

As discussed in Section 1, the reserve in the wind power that occurs while tracking the provided power reference can be utilized for improvement of wind turbine operation. Here, we are interested in reducing the loads experienced by the wind turbines. Note that in this paper the term loads refers to the forces and moments experienced by the wind turbine structure. To define the control objective one needs to resolve how to relatively compare two different load histories.

2.1 Comparing the load histories

The main driver of the wind turbine damage is the dynamic stress experienced by the structure. The cyclic stress causes material fatigue, which reduces the wind turbine operational life. The standard fatigue analysis is based on the Palmgren-Miner rule, see e.g. Sutherland (1999). This rule defines the total damage of the wind turbine component as:

$$D_{\mathrm{t}} = \sum_{j=1}^{M} \frac{n_j}{N_j}, \tag{1}$$

where n_j is the number of cycles that the structure undergoes at stress level σ_j, and the N_j is the number of cycles at the stress level σ_j that leads to component failure. The different stress levels are denoted by indices $j = 1, \ldots, M$. The Palmgren-Miner rule states that the component breaks when the total damage equals one.

The relation between the stress levels σ_j and the maximum number of cycles at that level N_j is described by the S-N curve, which can be well approximated with:

$$\sigma = CN^{-\frac{1}{m}}, \tag{2}$$

where m is denoted as Wöhler coefficient, the empirically determined parameter that characterizes the material, and C is the maximal static stress that the material can withstand.

The notion of total damage is typically used for lifetime calculations that aim at determining when will the total damage reach one. The lifetime calculation requires extensive simulations of different operating scenarios to be viable. For estimation of control benefits it is more common to use the damage equivalent loads. The damage equivalent load (DEL) is the amplitude of a sinusoidal load of constant frequency f which produces the same damage as the original signal. It is determined by (Bossanyi (2003a)):

$$DEL = \left(\sum_{j=1}^{M} \frac{\sigma_j^m n_j}{Tf} \right)^{\frac{1}{m}}, \tag{3}$$

where T is the duration of the load history.

The question remains how to extract the individual cycles from the complex signal. The method that is commonly used in fatigue analysis is the rainflow counting procedure described in Sutherland (1999).

The wind turbine simulation model at hand, Soltani et al. (2010), can provide the tower bending moment and the torsional torque of the shaft. In this work we use the damage equivalent loads computed from those load histories. This is a typical procedure for comparison between control strategies, see e.g. Bossanyi (2003b) and Bossanyi (2005). The DEL computation is performed by the MCrunch code (see Buhl (2010)) with $C = 1$, $Tf = 1$, $m = 4$ for the tower bending moment and $m = 8$ for the shaft moment.

2.2 Control design cost function

According to the previously described DEL is not a suitable load measure for use in the control design cost function. The rainflow counting algorithm is not analytic and the function (3) is nonlinear. Therefore, the aim is to find the cost function that is simpler, but which consequents in the reduction of DEL. The DELs will be computed a-posteriori to evaluate the control effects.

According to (3), the stress amplitudes enter the Palmgren-Miner sum linearly, while the number of stress cycles enters with the exponent $\frac{1}{m}$. This means that the contribution of the large cycles to the DEL is exponentially larger than that of the small cycles (e.g. one cycle of the shaft moment with the amplitude A contributes equally to DEL as 10^8 cycles of the amplitude $A/10$). Also, it should be noticed that the frequency of the cycles does not influence the damage equivalent loads.

Typically the oscillations of the wind turbine structures comprise of high frequency components (contributed to structure eigen-oscillations) and low frequency components (contributed to external excitation of the wind turbine subsystems). The low frequency components introduce larger cycles, while eigen-oscillations are smaller (especially if the wind turbine controller is well-designed, see Spudić et al. (2010)). The aim of the wind farm controller design is to reduce the excitation of these modes. Thus the largest cycles of the load histories can be reduced, which would in turn reduce DEL.

The wind farm controller design presented in this paper assumes that the 10-minute mean wind speed at each of the turbines is known (estimated) and that an initial distribution of wind turbine power references is known, i.e., a mean wind speed V^0 and the constant power reference P_{ref}^0 is attributed to every wind turbine. The distribution of constant power references can be obtained by some simple distribution (e.g. $P_{\mathrm{ref}}^0 = \frac{P_{\mathrm{ref}}^{\mathrm{WF}}}{N_{\mathrm{WT}}}$, where $P_{\mathrm{ref}}^{\mathrm{WF}}$ is the wind farm power reference and N_{WT} is the number of wind turbines in the wind farm) or this distribution can also be optimised by taking into account the quasi-stationary aerodynamics of the wind farm (interaction of wind farms through wakes), see e.g. Spruce (1993). The mean wind speed and the constant power reference determine the wind turbine operating point. The cost function penalizes the deviations from this operating point.

The chosen control design cost function is:

$$\begin{aligned} J(P_{\mathrm{ref}}(t),\ F_{\mathrm{T}}(t), T_{\mathrm{shaft}}(t)) :=& \\ := rP_{\mathrm{e}}(t)^2 + qT_{\mathrm{shaft}}(t)^2 + q_d \left(\frac{F_{\mathrm{T}}(t)}{dt} \right)^2, \end{aligned} \tag{4}$$

where r, q and q_d are the weighing coefficients, P_{e} denotes the deviations in produced power, T_{shaft} denotes the low-frequency shaft torque deviations, and F_{T} denotes the deviations of the thrust force (which is the excitation for the tower bending). The thrust force is penalized by its derivation to prevent the drifting of the power reference due to changes of the wind speed. Namely, the steady-state thrust force is dependant on the wind speed (disturbance). On the other hand, the steady state shaft torque depends only on the power reference. Therefore, the shaft torque deviation can be penalized by its absolute value.

3. CASE STUDIES

In this section the benefits of controlling the wind turbine via power reference are assessed. The following question is considered: can the wind turbine loads be reduced by introducing the power reference deviations, P_{ref}, via a closed loop optimal controller? To answer this question first a wind turbine is exposed to an artificial deterministic disturbance and then to a disturbance characteristic for wind turbine operation. The system response is compared to the case when the constant reference is provided to the system (i.e., the power reference deviations are zero).

To state an optimization problem the wind turbine model is required. Here we use a discrete linear state-space model of the wind turbine developed in Spudić et al. (2010):

$$\begin{aligned} x[t+1] &= Ax[t] + Bu[t] + B_d d[t], \\ y[t] &= Cx[t] + Du[t] + D_d d[t], \end{aligned} \tag{5}$$

where $x := \left[\beta, \omega_{\mathrm{g}}, \omega_{\mathrm{g}}^{\mathrm{filt}}\right]'$ (β is the pitch angle, ω_{g} is the generator speed and $\omega_{\mathrm{g}}^{\mathrm{filt}}$ is the filtered generator speed), $u = [P_{\mathrm{ref}}]'$, $d = [v]$, $y = [F_{\mathrm{T}}, T_{\mathrm{shaft}}]'$ and t denotes the discrete time instant.

Based on the discretized cost function (4) and wind turbine model, the wind turbine control problem is defined as a Constrained Finite-Time Optimal Control (CFTOC) problem (Borrelli et al. (2005)):

$$\begin{aligned} &\min_{U} \; U'\mathcal{R}U + Y'\mathcal{Q}Y + Y_d'\mathcal{Q}_d Y_d \\ &\text{subject to} \begin{cases} \mathcal{Y} = \mathcal{C}x_0 + \mathcal{D}U + \mathcal{D}_d D, \\ \mathcal{E}_U U \le \mathcal{F}_U, \end{cases} \end{aligned} \tag{6}$$

where: x_0 is the initial state of the system; N is the prediction horizon; U is the optimization variable, $U := [u_1', \ldots, u_{N-1}']'$; D is the vector of predicted disturbances, $D := [d_0', d_1', \ldots, d_{N-1}']'$; Y is the vector of predicted outputs, $Y := [y_0', \ldots, y_{N-1}']'$; Y_d is the vector of predicted output differences, $Y_d := [y_0' - y_{-1}', \ldots, y_{N-1}' - y_{N-2}']'$. The matrices $\mathcal{E}_U, \mathcal{F}_U$ define system constraints and $\mathcal{C}$, $\mathcal{D}$, $\mathcal{D}_d$ describe the system evolution that can be obtained from the wind turbine state-space model, see e.g., Maciejowski (2002). In this paper only the constraints on the control variable are defined. The minimal power reference is defined by generator properties, while the maximum is defined by the nominal generator power or, at lower wind speeds, by the available power.

The control weighing matrices are, according to (4), defined as: $\mathcal{R} := \mathrm{diag}\,(R, \ldots, R)$, $R \in \mathbb{R}$, $R > 0$ is the control weight matrix; $\mathcal{Q} := \mathrm{diag}\left(\begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix}, \ldots, \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix}\right)$, $Q \in \mathbb{R}$, $Q \ge 0$, is the output weight matrix; and $\mathcal{Q}_d := \mathrm{diag}\left(\begin{bmatrix} Q_d & 0 \\ 0 & 0 \end{bmatrix}, \ldots, \begin{bmatrix} Q_d & 0 \\ 0 & 0 \end{bmatrix}\right)$, $Q_d \in \mathbb{R}$, $Q_d \ge 0$, is the output difference weight matrix.

The wind turbine states are not weighted in this control problem because, as will be shown in the simulations, the action of the controller designed according to (7) stabilizes and improves the behavior of the overall system. Further penalization of states therefore only complicates the weight tuning.

The controller is designed as an on-line Model Predictive Controller (MPC) that uses a sampling time of 1 second. Every time instant the controller is fed with the current state vector, x_0, and, due to delta formulation, the output (thrust force) from a previous time instant, y_{-1}. All states used in the model, as well as the thrust force, are measurable or easily estimated.

In the following the case studies will be presented that demonstrate the potentials of wind turbine control via a wind farm controller. This case studies are for demonstration purpose, while the design of the wind farm controller based on this will be demonstrated in the next section.

All case studies are performed on the full-scale nonlinear wind turbine model from Soltani et al. (2010).

3.1 Deterministic input

The first case study tests the controller performance in the case of a positive and negative step change of 2 m/s in wind speed. The aim of this case study is to determine the full potential of this type of the controller. Therefore, the prediction horizon $N = 10$ is used, to make sure that the entire transient is predicted, and the perfect disturbance prediction is used, meaning that the controller has the exact information about the wind speed in the next 10 seconds.

In the following experiments different weight settings are used to demonstrate the trade-offs between the competing objectives.

Reducing tower loads In this experiment $\mathcal{Q}$ is set to zero in order to estimate the potential for minimizing tower loads. The results of the experiments are depicted in Figure 2. The first glimpse reveals that the controller has a substantial ability to reduce the tower bending, however at an extremely high control cost.

For weight ratio $Q_d/R = 1000$ the tower deflection amplitude during the positive wind step is reduced by more than 50%. This is achieved by the change in power of more than 2 MW. This large change in power is naturally followed by a large increase in shaft torque. During the positive wind step the control input ran into the constraint. This kind of system behavior is not acceptable.

When the weight ratio is reduced to $Q_d/R = 100$ the reduction in tower bending is around 10 %, which is achieved by the maximal power deviation of around 750 kW. This power deviation is still large and the shaft oscillations are still much increased.

Introducing the power controller also improved behavior of wind turbine states. There is less pitch action (with weighting $Q_d/R = 1000$ the pitch response is aperiodic, while weighting $Q_d/R = 100$ significantly reduces the response overshoot). The overshoot of the rotor speed is also reduced, the transient is less oscillatory and the nominal speed is restored faster.

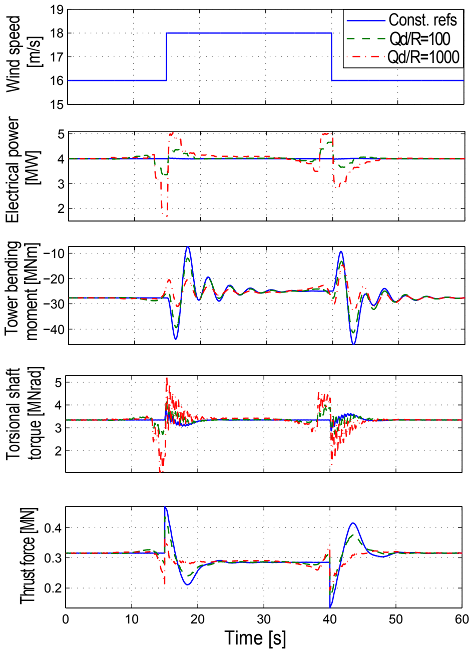


Fig. 2. Deterministic disturbance - Reducing tower loads

One should notice that this controller relies very much on the feed-forward control action (the large drop in control variable before the positive step and the large increase before the negative step). This is problematic because it indicates that the inaccuracy in disturbance prediction might lead to poor performance. The assumptions on the perfect prediction will be weakened in the Section 3.2 where the realistic wind disturbance will be considered.

To conclude, this experiment reveals the potential for alleviating the thrust-induced loads, however, the weight that penalizes the thrust needs to be kept small to prevent violent control and increase in shaft loads. It has to be kept in mind that this type of disturbance is artificial and the typical wind disturbance is less violent, so the behavior of the controller can be expected to improve for different scenarios.

Reducing shaft loads In this experiment $\mathcal{Q}_d$ is set to zero in order to estimate the potential for minimizing shaft loads. The results of the experiments are depicted in Figure 3. The simulation outputs demonstrate the potential for shaft load reduction at a much smaller control cost. The system response for weight ratio $Q/R = 2$ is very satisfactory, the maximal power deviation is 200 kW, while the amplitude of the slow frequency load cycles has reduced significantly. The high frequency oscillations are not additionally excited. The tower loads remain much the same as in the case of constant reference. For the higher weight ratio $Q/R = 20$ the response of the shaft torque deteriorates because, due to more violent control actions, the high frequency oscillations increase in amplitude. In this case the low-frequency component of the shaft torque (the only one modeled in the control design model) is still reduced, however the overall response deteriorated due to increased high-frequency oscillations.

Also in this case the response of the wind turbine states is improved, the speed tracking is improved and the pitch action is reduced. Also, there is no significant feed-forward control action.

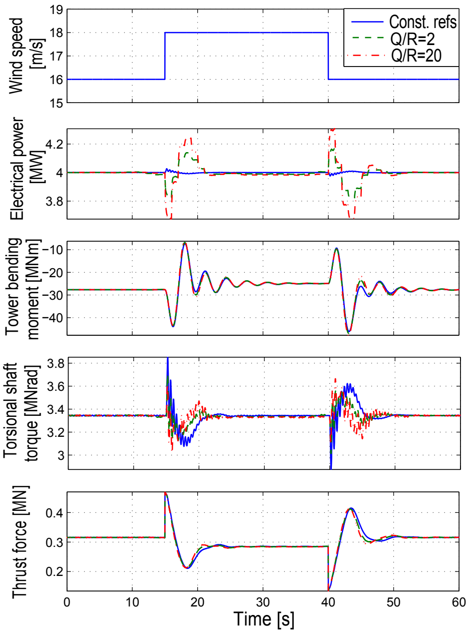


Fig. 3. Deterministic disturbance - Reducing shaft loads

To conclude, this experiment demonstrates that there exist an opportunity to improve the shaft loading at a relatively small control effort. However, to asses the benefits correctly it is necessary to apply the realistic disturbance and compute the damage equivalent loads.

3.2 Turbulent wind

In reality the wind turbine is exposed to turbulent wind. Turbulence can be described as a stochastic signal, by its turbulence intensity and its spectrum. To properly simulate the turbulence one needs to take into account the frequency characteristics of the point-wise wind speed, the spatial correlation of the wind, and the wind field propagation that renders the time-wise correlation. In order to obtain a realistic excitation of the wind turbine, the turbulent wind speed for this case study is simulated according to the turbulence model implemented in Soltani et al. (2010). The turbulence intensity used in simulations is 6%.

From the experiments with the deterministic disturbance the weights $Q/R = 2$ and $Q_d/R = 30$ are found satisfactory and will be used in further simulations. In the first simulation the assumption of perfect prediction of disturbances is kept and the prediction horizon is $N = 10$.

The results of this simulation are given in Figure 4. The Figure 5 shows the magnification of the response in order to depict the fast scale dynamics. The simulation outputs suggest that the variance of the shaft torque has been reduced, while the high frequency shaft oscillation have not been enhanced (apparent from the response detail in Figure 5). The control action is in the acceptable range (±150 kW) and there are no large jumps in the control variable. The effects on the tower bending can not be clearly assessed from the graphical depiction of the responses.

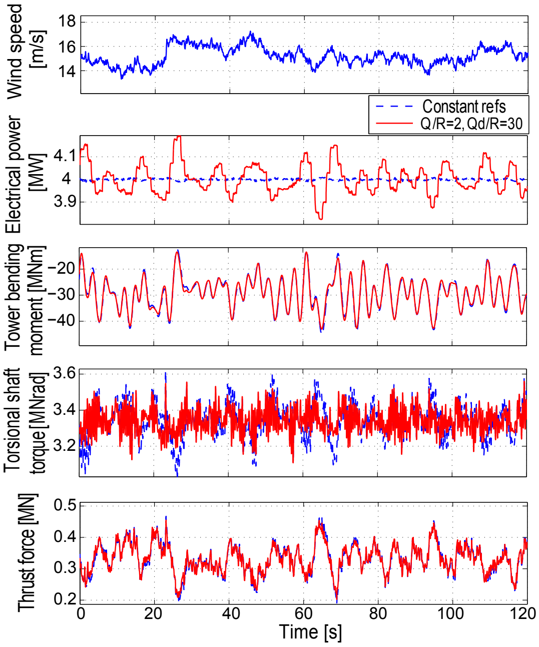


Fig. 4. Turbulent wind scenario

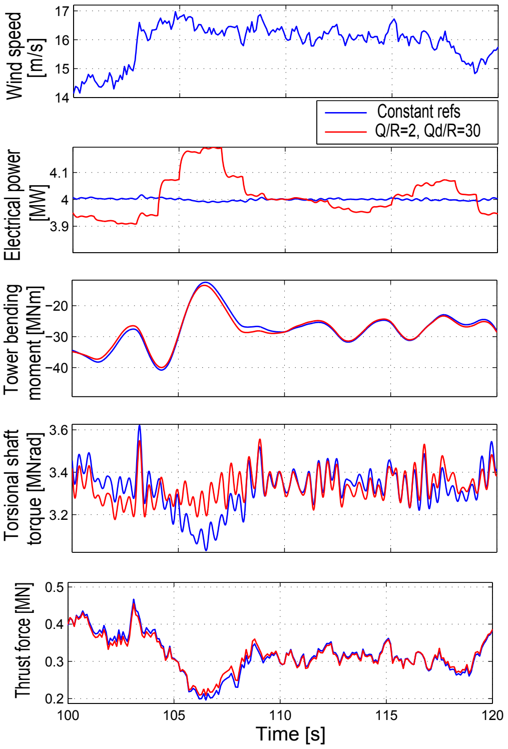


Fig. 5. Turbulent wind scenario (detail)

To asses the benefits of this control design one needs to perform the damage equivalent load analysis, which is reasonable since the applied disturbance (unlike the deterministic one) actuates all the representative system modes. The statistics (tower and shaft DELs and standard deviations (STDs) of the pitch rate, rotor speed and electrical power) of the simulation responses are given in the second column (denoted Perfect prediction) of the Table 2. The statistics are performed on the 500 second simulation run. The statistics show that the shaft DEL has reduced by 18%, while the tower DEL reduced by 4%. The standard deviation of electrical power increased to 67 kW, which is a reasonable value. This results demonstrate a good trade-off between the increase in control effort and decrease in the turbine loads. It is also important to notice that the pitch angle activity is reduced and speed tracking is improved. This shows that the added controller does not compete with the wind turbine controller, but improves the overall wind turbine behavior.

Table 1. Turbulent wind scenario statistics

	Constant reference	Perfect prediction	Persistence assumption
T_{shaft} DEL [Nrad]	$7.6203 \cdot 10^5$	$6.2449 \cdot 10^5$	$6.7610 \cdot 10^5$
M_{tow} DEL [Nm]	$6.5793 \cdot 10^7$	$6.3153 \cdot 10^7$	$6.4097 \cdot 10^7$
$d\beta/dt$ STD [°/s]	0.8085	0.8027	0.7935
ω_{r} STD [rad/s]	0.0162	0.0158	0.0156
P_{e} STD [kW]	4.2817	67.1112	45.3780

However, the assumption of the perfect wind prediction in the horizon of 10 seconds is unrealistic. For the next experiment this assumption is dropped and replaced by the assumption that the wind speed estimated wind speed at given time (d_0) will be constant during the prediction horizon. When this assumption is introduced it is not sensible to keep such long prediction horizon. Namely, due to relatively low frequency content of the turbulent wind such assumption (commonly referred to as *persistence* assumption) is valid for short horizons, however the validity severely deteriorates with increase of the prediction horizon. By performing several simulations the prediction horizon $N = 3$ was shown to provide the best results. The statistics of the results are given in the third column of the Table 1, denoted Persistence assumption.

The statistics show the expected decrease in performance in comparison to the assumption of perfect prediction. However, in comparison to simulation in which the power reference is kept constant there is still significant improvement, 11% improvement in shaft DEL and 3% reduction in tower DEL. The reduction in tower damage is very small, which can be contributed to the lack of feed-forward action since the disturbances are not predicted. However, in several simulation that were performed with different excitations a small improvement in tower loads proved to be consistent. The improvements in the shaft load are significant and also consistent. The support to speed control is evident in reduction of pitch action and improvement of speed tracking.

4. WIND FARM CONTROL FOR LOAD MINIMIZATION

In the previous section the case studies were shown that demonstrate the potential for improvement in wind turbine operation by controlling the power reference. Such control of an individual turbine is doubtfully beneficial, since the power production of the wind turbine is significantly deteriorated. However, this type of control can be used to control the clusters of wind turbines (i.e., wind farms). The costs of the individual wind turbine control problems (7) are summed together and the constraint is added that has to ensure that the wind farm will deliver the required power.

To formulate the control problem we assume that the stationary power references, P_{ref}^{j0} (where j is an index that denotes an individual wind turbine in the cluster), are attributed to the wind turbines and that they add-up to the exact amount of the wind farm power reference, $\sum_{j=1}^{N_{\mathrm{WT}}} P_{\mathrm{ref}}^{j0} = P_{\mathrm{WF}}^{\mathrm{ref}}$, where N_{WT} denotes the number of turbines in the wind farm and $P_{\mathrm{WF}}^{\mathrm{ref}}$ is the wind farm power reference.

Then, we can define the simple wind farm optimal control problem as:

$$\min_{U^1,\ldots,U^{N_{WT}}} \sum_{j=1}^{N_{\mathrm{WT}}} U^{j'}\mathcal{R}U^j + Y^{j'}\mathcal{Q}Y^j + Y_d^{j'}\mathcal{Q}_dY_d^j$$
$$\text{subject to} \begin{cases} \mathcal{Y}^j = \mathcal{C}^jx_0^j + \mathcal{D}^jU^j + \mathcal{D}_d^jD^j, \\ \mathcal{E}_U^jU^j \leq \mathcal{F}_U^j, \\ \sum_{j=1}^{N_{\mathrm{WT}}} [\,1\ 0\ \ldots\ 0\,]\,U^j = 0 \end{cases} \tag{7}$$

where j denotes the variables and parameters attributed to the j-th wind turbine.

Essentially, this formulation allows only the control moves that add-up to zero. This seems rather conservative, however, one has to consider the fact that wind turbines in wind farms are relatively far apart and the turbulence that they experience at a certain moment are not significantly correlated. Therefore, the larger the controlled cluster gets the turbulence effects tend to level out (i.e., loosely put, there is a larger chance that there exists the turbine which requires the complementary control).

Here, we present the results of the simulation of a small wind farm consisting of only two wind turbines (statistically the worst case). The generated wind histories are not correlated. The statistics of the run are given in Table 2.

Table 2. Wind farm controller statistics

	Wind turbine 1	
	Const. ref.	WF control
T_{shaft} DEL [Nrad]	$7.6108 \cdot 10^5$	$7.2495 \cdot 10^5$
M_{tow} DEL [Nm]	$6.5696 \cdot 10^7$	$6.5012 \cdot 10^7$
$d\beta/dt$ STD [°/s]	0.8095	0.8035
ω_{r} STD [rad/s]	0.0162	0.0158
P_{e} STD [kW]	4.2803	32.1285
	Wind turbine 2	
	Const. ref.	WF control
T_{shaft} DEL [Nrad]	$8.1920 \cdot 10^5$	$7.5618 \cdot 10^5$
M_{tow} DEL [Nm]	$7.5716 \cdot 10^7$	$7.4977 \cdot 10^7$
$d\beta/dt$ STD [°/s]	0.7394	0.7300
ω_{r} STD [rad/s]	0.0150	0.0148
P_{e} STD [kW]	4.6279	30.8181
	Wind farm	
	Const. ref.	WF control
P_{WF} STD [kW]	6.4017	6.4193

The shaft DELs were reduced by 5% on the first wind turbine and by 8% on the second wind turbine. The tower DELs were reduced by 1% on both wind turbines. The increase in standard deviation of the wind farm power is negligible. The improvement in speed control is still present. The overall (cumulative) percentage reduction of loads in the wind farm is around the same level as for the single controlled wind turbine.

5. CONCLUSION

The paper analyses the wind farm control problem and gives an assessment of the potential for reduction of wind turbine loads via power control of wind turbines. It is shown that the significant reduction of shaft loads can be obtained, while the potential for reduction of thrust induced loads is smaller.

Most importantly, it is demonstrated that it is possible to achieve reduction in loads without deteriorating any of the operating conditions – the wind farm power is maintained while all considered loads are reduced, the speed control is improved and the pitch action is reduced. Therefore, the wind farm can benefit from coordinated wind turbine control.

ACKNOWLEDGMENTS

This research was supported by the EU FP7 project Aeolus, Grant Agreement No 224548; by the Ministry of Science, Education and Sports of the Republic of Croatia under grant No 036-0361621-3012; and by the Croatian Science Foundation and Končar - Electrical Engineering Institute. This support is gratefully acknowledged.

REFERENCES

Borrelli, F., Baotić, M., Bemporad, A., and Morari, M. (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10), 1709–1721.

Bossanyi, E.A. (2003a). *GH Bladed Theory Manual*. Garrad Hassan and Partners Ltd, 11 edition.

Bossanyi, E.A. (2003b). Individual blade pitch control for load reduction. *Wind Energy*, 6, 119 – 128.

Bossanyi, E.A. (2005). Further load reductions with individual pitch control. *Wind Energy*, 8, 481 – 485.

Buhl, M. (2010). NWTC Design Codes (MCrunch). `http://wind.nrel.gov/designcodes/postprocessors/mcrunch/`. Last modified 25-October-2010; Accessed 02-November-2010.

Elkraft System and Eltra (2004). Wind turbines Connected to Grids with Voltages above 100 kV – Technical regulation for the properties and the regulation of wind turbines. Regulation TF 3.2.5.

Hansen, A.D., Sørensen, P., Iov, F., and Blaabjerg, F. (2006). Centralized power control of wind farm with doubly fed induction generators. *Renewable Energy*, 31, 935 – 951.

Jonkman, J., Butterfield, S., Musial, W., and Scott, G. (2009). Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical report, National Renewable Energy Laboratory.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Pearson Education Limited, Essex.

Soltani, M., Knudsen, T., Grunnet, J., and Bak, T. (2010). Aeolus toolbox for dynamic wind farm model, simulation, and control. In *European Wind Energy Conference*. Krakow, Poland.

Sørensen, P., Hansen, A.D., Thomsen, K., Buhl, T., Morhorst, P.E., Nielsen, L.H., Iov, F., Blaabjerg, F., Nielsen, H.A., Madsen, H., and Donovan, M.H. (2005). Operation and control of large wind turbines and wind farms - Final report. Technical Report Risø-R-1532(EN), Risø National Laboratory.

Spruce, C.J. (1993). *Simulation and Control of Windfarms*. Phd thesis, University of Oxford.

Spudić, V., Jelavić, M., Baotić, M., Vašak, M., and Perić, N. (2010). Aeolus Deliverable 3.3. Reconfigurable Control Extension. Project report, University of Zagreb, Faculty of Electrical Engineering and Computing.

Sutherland, J. (1999). On the fatigue analysis of wind turbines. Technical report, Sandia National Laboratories, Albuquerque, New Mexico, USA.

Advanced Process Control BGHT7 Desulphurization Unit

M. Čižniar* D. Puna**

** Honeywell Process Solutions, Honeywell s.r.o., Mlynské nivy 71, P.O.BOX 75, 820 07 Bratislava 27, Slovakia
(Tel: +421 2 322 622 70; e-mail:michal.cizniar@honeywell.com)
** Automation and Control Solutions, Honeywell s.r.o., V Parku 2326/18, 148 00 Praha 4, Czech Republic
(Tel: +420 242 442 305; e-mail:dalibor.puna@honeywell.com)*

The contribution deals with the design and practical implementation of an advanced process control (APC) on the BGHT7 Desulphurization Unit at Slovnaft, Bratislava Refinery.

First, the process and operation of the BGHT7 Desulfurization Unit is briefly described, then the control objectives are introduced, and finally, design and implementation of multivariable predictive control solution is presented.

Laboratory for Renewable Energy Sources and Identification of the Laboratory Wind Turbine Model

V. Bobanac* M. Brekalo M. Vašak*** N. Perić******

Faculty of Electrical Engineering and Computing, University of Zagreb, Unska 3, HR-10000 Zagreb, Croatia
** (Tel: +385-1-6129805; e-mail: vedran.bobanac@fer.hr)*
*** (Tel: +385-34-263248; e-mail: brekalomarijan@gmail.com)*
**** (Tel: +385-1-6129821; e-mail: mario.vasak@fer.hr)*
***** (Tel: +385-1-6129863; e-mail:nedjeljko.peric@fer.hr)*

Abstract: This paper presents Laboratory for Renewable Energy Sources (LARES) at the Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. Laboratory consists of experimental setups for wind energy, solar energy and hydrogen fuel. The aim of LARES is development and experimental research of the advanced control strategies, in order to improve the energy conversion efficiency and thus increase the cost effectiveness of renewable energy sources. Focus of this paper is placed on the wind part of LARES and especially on mathematical model identification of the laboratory wind turbine. Obtained model is a basis for subsequent development of the wind turbine control algorithms.

1. INTRODUCTION

One of the main concerns of today's civilization is electrical energy production. The society, the more developed it is, consumes more energy. Therefore, growth of human population and development of societies imply increased energy demand. This trend will surely be continued in the future. Current world energy production is still based on burning of fossil fuels which becomes more and more unacceptable, mainly because of the current levels of related CO_2 emissions which cause ecological problems. On the other hand fossil fuel reserves will sooner or later be exhausted. In last two decades numerous international agreements and protocols have been signed, which oblige the signing countries to reduce the CO_2 emissions. Thus, many governments are forced to subsidize development and building of the ecologically acceptable power plants. These are all reasons why investment in renewable energy sources have been growing rapidly in the last decade. EU members have set a goal that by the year of 2020 20% of total electrical energy production should come from renewable energy sources. According to the latest information from European Wind Energy Association (http://www.ewea.org/ 2011), this goal will be exceeded. Renewable energy source with the highest growth rate and potential for further development is the wind energy.

However, renewable energy is still not rentable like energy obtained from classical sources (e.g. thermal power plants). With the development of many new power plants based on renewable energy sources, one might rightfully raise a question about the energy conversion efficiency. The aim of the Laboratory for Renewable Energy Sources (LARES) on Faculty of Electrical Engineering, University of Zagreb (UNIZG-FER) is development and experimental research of the advanced control strategies, in order to improve the energy conversion efficiency and thus increase the cost effectiveness of the renewable energy sources.

The paper is organized as follows. Section 2 describes LARES in whole and in Section 3 wind part of LARES is described in more detail. In Section 4 insight into the LARES wind turbine control system is given. Section 5 describes the wind turbine identification experiment and the identification results are given in Section 6. Conclusions are given in Section 7.

2. DESCRIPTION OF THE LABORATORY

LARES is placed on the top floor of the UNIZG-FER skyscraper and it consists of the following experimental units:

- Wind turbine setup placed in the air chamber and driven by a fan;
- An array of solar panels placed on the roof of the skyscraper (under construction);
- Hydrogen fuel cells stack with metal hydride storage supplied by an electrolyser.

Principle scheme of LARES is shown in Figure 1. From the scheme it is visible that the three main parts of LARES are actually connected in a microgrid. This microgrid conceptually operates as follows. Energy obtained from wind and sun is either transferred to the grid or it is used in an electrolyser in order to produce hydrogen which is then deposited in a metal hydride storage. In cases when energy from wind and sun is not sufficient, energy stored in hydrogen can be converted into electrical energy by using the fuel cells stack. The electrical energy is fed into the grid through properly controlled power converters.

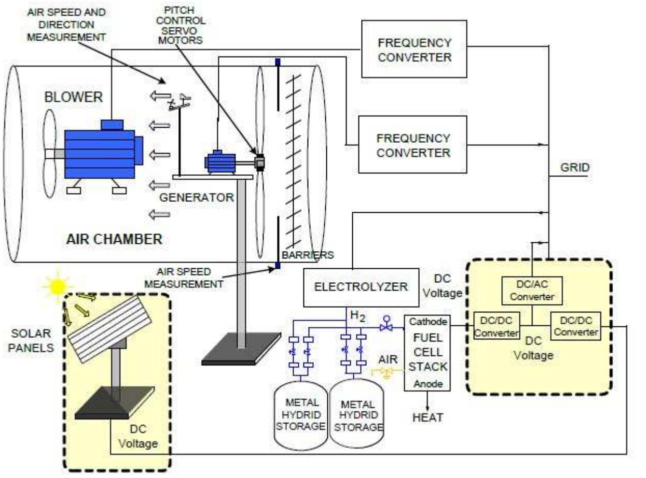


Fig. 1. Principle scheme of LARES.

The LARES configuration enables development and experimental research of control algorithms for each particular setup (wind, sun or hydrogen). The laboratory as a whole enables experimental research related to control of microgrids.

Layout of LARES is shown in Figure 2, where solar panels are not shown, as they are placed on the top of the skyscraper (just above the laboratory which is on the top floor).

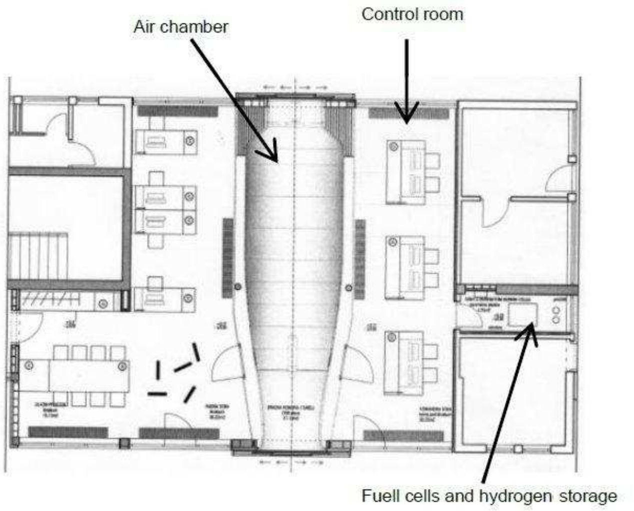


Fig. 2. LARES layout.

Focus of this paper is placed on the wind part of the laboratory, such that the rest of it relates only to the wind energy. In the following section a more detailed description of the wind turbine setup and of the air chamber is given.

3. WIND TURBINE AND WIND CHAMBER

Development and construction of the wind turbine setup was the most challenging task during the construction of the laboratory. The most important goal was the preservation of the aerodynamic relations present at MW-class wind turbines. The wind turbine setup design was initiated by extensive analysis and simulations in professional tools, where the basic requirements were (Perić et al. 2010):

- Preservation of Betz assumption regarding energy transformation;
- Fulfilling of dynamic, kinematic and geometry conditions;
- Respecting of optimal tip-speed ratio for three bladed rotor.

Laboratory wind turbine has rated power of 300 W and rated rotational speed of 240 rpm. Wind turbine setup is placed inside the wind chamber as shown in Figure 3.

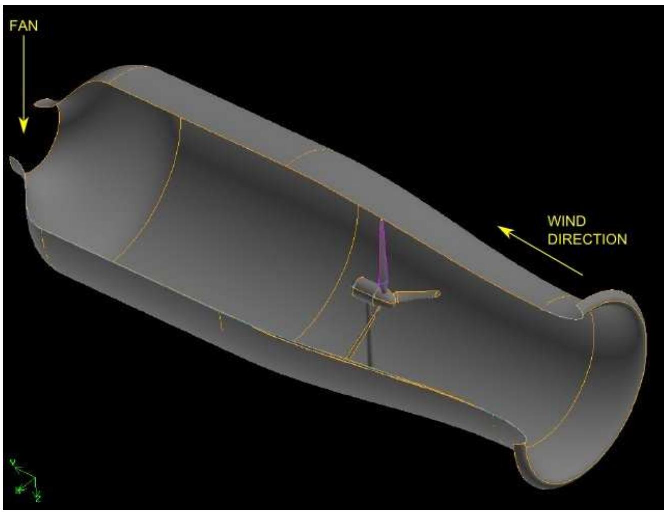


Fig. 3. Wind chamber layout.

The fan is placed at the far end of the chamber such that, when the wind is produced, the particles of air first pass across the turbine and then across the fan. Fan is controlled over the frequency converter which allows various rotational speeds of the fan and thus various wind speeds inside the chamber. This enables research of the wind turbine operation in wind conditions which are close to natural, where wind is stochastic.

Majority of the modern MW-class wind turbines are controlled by varying the generator torque and by pitching the rotor blades. This control structure was implemented on the laboratory wind turbine also. Since no commercially available small wind turbine could be used, completely new design had to be developed. This design included:

- 4-quadrant (4Q) frequency converter which is used for connecting the turbine synchronous generator to the grid;
- Actuators for pitching the rotor blades around their longitudinal axis.

4Q frequency converter allows electromagnetic torque of the turbine generator to be varied in a wide range which is used for control of the turbine rotational speed. Each of the three rotor blades has a DC servo drive which is used for pitching the blade around its longitudinal axis also for rotor speed control purposes. It is worth mentioning that laboratory turbine hub has a very limited space, so significant effort was needed to fit all of the components (servo motors, gearboxes, pitch controllers and blade position sensors). The issue of control objectives and realization of the laboratory turbine control system is addressed in Section 4.

MW-class wind turbines due to their large dimensions have a flexible tower with low modal frequencies. In order to obtain plausible laboratory turbine model, flexible tower had to be constructed, which was not an easy task, since this demand comprises structural stability of the laboratory turbine. Solution was found in the form of the stiff tower which is mounted on an oscillatory bed and connected with the rigid structure through the system of springs and dampers. There are various types of springs and dampers and they can be replaced in order to obtain different tower characteristics.

Wind turbine and the wind chamber are also equipped with numerous sensors and instruments like: an array of 15 anemometers, torque sensor, accelerometers, strain gauges, web camera, stroboscope and weather station.

Realization of the wind turbine, the fan and the wind chamber is shown in Figure 4.

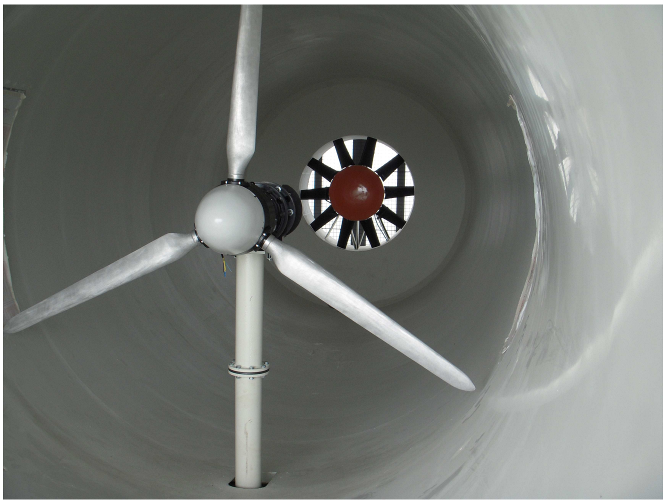


Fig. 4. Wind turbine setup in LARES.

Since aerodynamic relations present on the MW-class wind turbines were successfully transferred to the laboratory wind turbine and since all the means for controlling MW-class turbines were implemented on the laboratory turbine, it can be concluded that the design was successful and that the wind turbine setup can serve its primary purpose: development and research of the advanced control algorithms which can subsequently be implemented on the real, MW-class wind turbines.

4. WIND TURBINE CONTROL SYSTEM

Nonlinear relation between power in the wind and the wind speed (Jelavić et al. 2009) has resulted in modern wind turbines having two main operating regions: below rated wind speed and above rated wind speed. Rated wind speed is defined as the lowest wind speed at which wind turbine is operating at its rated power.

Below rated wind speed power that can be extracted from the wind is smaller than the rated power of the wind turbine and the control objective in this region is to maximize energy conversion efficiency. Control is performed by adjusting the generator torque, while the pitch angle is kept around minimum value of 0^o, such that the system operates along the maximum power coefficient curve (Jelavić et al. 2009). Controlling the generator torque is possible because modern generators are not connected to the grid directly, but over a frequency converter which allows generator speed to be varied in a wide range. Wind turbines with synchronous generator that is connected to the frequency converter of rated generator power are particularly suitable for such control strategy. The wind turbine setup in LARES is constructed such that it satisfies these characteristics.

Above rated wind speed power that can be extracted from the wind grows rapidly with the wind speed and it is greater than the rated power of the wind turbine. Control objective in this region is to limit the turbine rotational speed on its rated value which implies that the wind turbine is operating at the rated power. This is the case because generator torque is kept at the rated value above rated wind speed. Control in this region is performed by pitching the rotor blades which changes aerodynamic characteristics of the blades in order to lower the energy conversion efficiency. This degradation is necessary as the wind turbine generator should not operate above its rated power.

Explanation given above describes basic principles of classical control system for modern wind turbines, structure of which is shown in Figure 5.

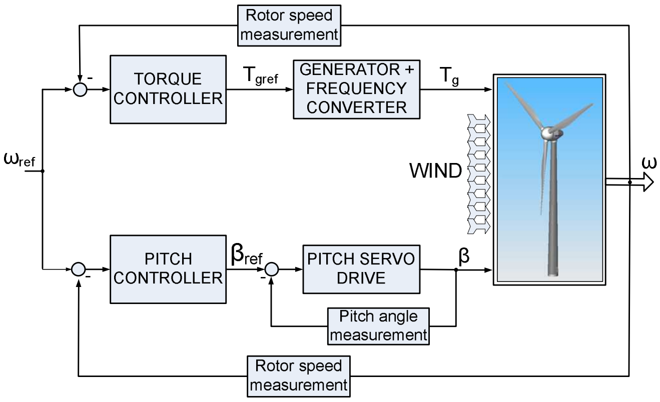


Fig. 5. Classical wind turbine control system.

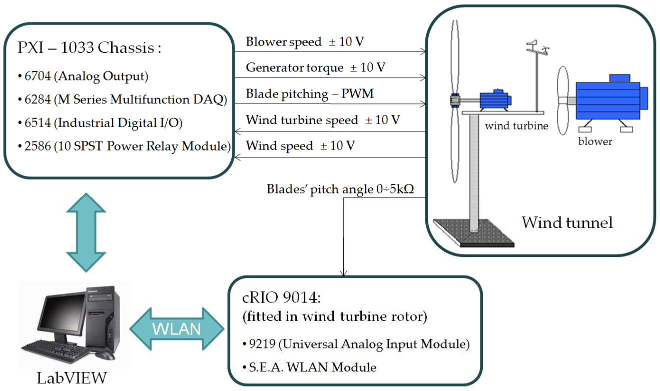

PXI – 1033 Chassis :
• 6704 (Analog Output)
• 6284 (M Series Multifunction DAQ)
• 6514 (Industrial Digital I/O)
• 2586 (10 SPST Power Relay Module)
Blower speed ± 10 V
Generator torque ± 10 V
Blade pitching – PWM
Wind turbine speed ± 10 V
Wind speed ± 10 V
wind turbine
blower
Wind tunnel
Blades' pitch angle 0÷5kΩ
WLAN
LabVIEW
cRIO 9014:
(fitted in wind turbine rotor)
• 9219 (Universal Analog Input Module)
• S.E.A. WLAN Module

Fig. 6. Principle scheme of the wind turbine control system in LARES.

The laboratory wind turbine control system is based on platform LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) (National Instruments 2007), and it is implemented on a PC. All signal processing and control computational tasks are performed on a PC, while data acquisition and signal generation are performed on specialized input-output circuitry, produced also by National Instruments. Principle scheme of the laboratory wind turbine control system is shown in Figure 6.

The equipment for signal generation and acquisition can be divided into two groups:

- PXI-1033 chassis which is used for communication between the PC and the input-output modules;
- cRIO-9014 controller used for blade pitch angle measurement.

PXI-1033 chassis contains 4 modules over which various analog and digital signals are measured and generated. Some of the most important signals are: analog output signals for defining the fan speed and generator electromagnetic torque, analog input signals of measured generator speed and rotor position, output PWM signals for pitching of each of the turbine blades, digital output signals for starting the fan and the generator, digital input signals which indicate state of the plant (fan works, generator ready…) etc.

cRIO-9014 is mounted in the turbine rotor together with 3 potentiometer sensors for measuring blade pitch angles. cRIO consists of modules which are programmed to periodically collect the measured data and send it to PC by a WLAN protocol.

Configuration of the laboratory wind turbine control system allows both classical and advanced control strategies to be implemented and tested. Wind turbine and wind farm control has been one of the main research interests at UNIZG-FER for past ten years. In this period many advanced wind turbine control and estimation algorithms have been developed or upgraded. For instance, one such algorithm is individual blade pitch control which can significantly reduce the mechanical loads on the turbine construction (Jelavić et al. 2008). So far these algorithms have been tested using professional simulation tools such as GH Bladed (Jelavić et al. 2008, 2009). Now these algorithms can be tested on a laboratory turbine where all the problems present in real systems can be considered. This is one step closer to the final goal which is implementation of the advanced control algorithms on real MW-class wind turbines.

5. IDENTIFICATION PROCEDURE

Basis for designing a control system is a reliable mathematical model of a process that is to be controlled. One way of obtaining a mathematical model is theoretical analysis and setting the equations which describe behavior of the process (e.g. equations of energy and momentum balance). Such a model of a laboratory wind turbine has already been obtained (Jelavić et al. 2009) and it has been used for design of various control algorithms and for testing of these algorithms in simulations. However, when it comes to real, practical systems a more reliable mathematical model may be obtained by process identification. Process identification is another way of obtaining a mathematical model, which is based on processing and analysis of input/output data obtained on the real process. Identification of the laboratory wind turbine mathematical model is described in the sequel.

It has been decided to use a parametric method of identification and to search for a mathematical model in a state-space form, since this form is appropriate for subsequent advanced control system design. Also, the state-space model form is a good choice for identification of processes with unknown structure, because it requires only two parameters: model order and input delays (MathWorks 2010). Since the process is nonlinear (Jelavić et al. 2009) it cannot be described with a single linear mathematical model. Therefore, identification is performed in 4 operating points which are defined by the wind speed. For each operating point a linear mathematical model in state-space form is obtained. Inputs to the system are wind speed (kept constant for a certain operating point), generator torque and pitch angle, while the output is turbine rotational speed, as shown in Figure 5.

Identification is performed in 2 operating points below rated wind speed and in 2 operating points above rated wind speed. Transfer function (1) is obtained below rated and transfer function (2) above rated. This is reasonable since in classical control configuration generator torque and pitch angle are solely used for control in operating regions below and above rated wind speed, respectively (see Section 4).

$$G_T(z) = \frac{\omega(z)}{T_{gref}(z)} \tag{1}$$

$$G_\beta(z) = \frac{\omega(z)}{\beta_{ref}(z)} \tag{2}$$

Identification procedure includes the following steps:

1. Gathering of a-priori knowledge and information about the process;
2. Selection of the input test signal;
3. Imposing the test signal to the system and measurement of the input/output signals;
4. Obtaining mathematical model by processing of the recorded input/output data with the identification software;
5. Validation of the obtained mathematical model.

In the sequel these steps are described in more detail.

Step 1.

A-priori knowledge about the wind turbine process is gathered from a theoretically obtained nonlinear mathematical model. This model was previously obtained for the MW-class wind turbine (Jelavić et al. 2009) and it was subsequently adjusted to suit the laboratory turbine. The model is implemented both in Matlab and LabVIEW. By simulations of this model important information are obtained which are primarily used for selecting the test signal parameters. This is described in the following step.

Step 2.

Parametric identification is based on processing of the measured input/output data of the observed system. Therefore, selection of the proper input test signal is very important. In theory white noise is a very good test signal because it excites all the frequency modes of the system. This is important, because the resulting model should contain all of the system's important dynamics. Since white noise cannot be physically realized, for linear model identification PRBS (*Pseudo Random Binary Sequence*) is used instead. PRBS is a signal which has similar characteristics to the white noise. It is a rectangular signal of various width. It can take on only two values (*binary*) which change pseudo-randomly (*pseudo random*) at multiples of the sampling time. The pseudo random signal sequence repeats several times. PRBS signal was realized in LabVIEW by using shift register and an "exclusive OR" function.

Selection of PRBS parameters is vital for identification. Parameters that are to be selected are: c – amplitude, Δt – clock time (minimum length between signal changes) and N – number of clock times in a single period. Example of a PRBS signal is shown in Figure 7.

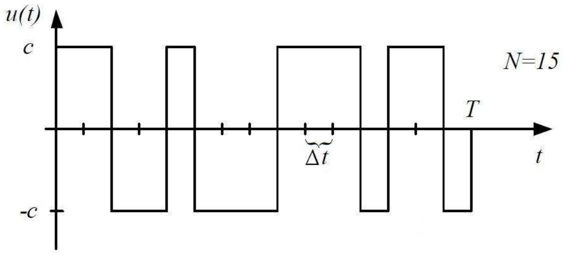


Fig. 7. Example of a PRBS signal – single period.

Following guidelines for determining parameters of a PRBS signal were taken from (Perić et al. 2005). Amplitude c is chosen depending on the process characteristics. It must be neither too small (because signal/noise ratio would then be too small) nor too large (because system can drift away from its operating point or nonlinear effects may be pronounced in responses). Clock time is chosen as approximately 1/5 of the smallest dominant system time-constant. N is chosen so that PRBS period T defined in (3) has a value about 50% larger than the system's impulse response settling time (t_{95}). Approximate relation is given in (4).

$$T = N \cdot \Delta t \tag{3}$$

$$T \approx 1.5 \cdot t_{95} \tag{4}$$

Number of repetitions of a PRBS signal is bounded by the allowed measurement time which is given by relation:

$$T_M = M \cdot T\,, \tag{5}$$

where M is the number of PRBS signal periods used in identification. Larger M is required for better signal/noise ratio.

Another important parameter for identification is sample time T_S. Sample time can be determined from the following empiric relation:

$$T_S = \left(\frac{1}{6} \div \frac{1}{10}\right) \cdot t_{63}\,, \tag{6}$$

where t_{63} relates to the system's step response rise time. When determining sample time one should bear in mind that Nyquist-Shannon sampling theorem must be satisfied for higher resonant frequencies. This can be controlled by determining frequency characteristics of the system, as described in (Brestovec 2005). It should be pointed out that the just mentioned frequency characteristics, as well as the impulse and step responses (for information about t_{95} and t_{63}) are determined from a theoretical mathematical model (this is the required a-priori knowledge about the system, which was mentioned in *Step 1*).

Step 3.

System must be brought to the desired operating point in closed-loop control. Once the system reaches the steady-state controllers are turned off and the test PRBS signal is imposed to the system input (torque or pitch reference depending on the operating point). During the experiment system output (turbine rotational speed) is recorded. Note that the identification is performed with control loop open (controllers are turned off during the identification data collection).

Step 4.

After the experiment is finished sets of the recorded input/output data are forwarded to the identification software. In this case Matlab's SIT (*System Identification Toolbox*) was used. SIT was configured to use parametric identification procedure shown in Figure 8.

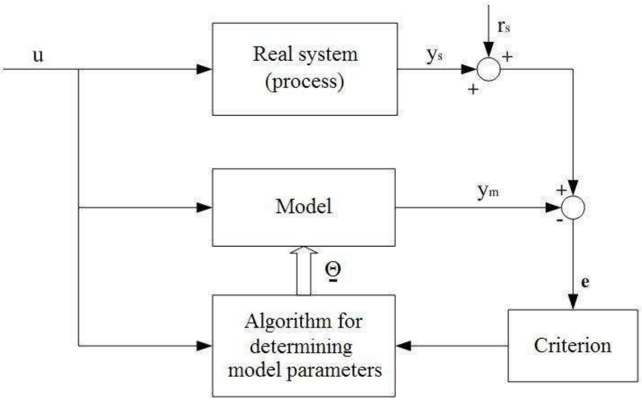


Fig. 8. Principle scheme of parametric identification procedure.

In the parametric identification procedure optimization methods are used to vary the set of model parameters, in order to minimize the difference between the model output and the measured output. Model order (and thus the number of model parameters) can be defined by the user.

Step 5.

After the model has been obtained it is validated on the set of data which is different from the one used for identification. Usually a step response is used for validation of the identified models. However, since our laboratory wind turbine setup is extremely sensitive to disturbances in the form of outer wind, it was impossible to obtain reliable, repeatable step responses on a real system. Therefore, data obtained during identification experiment was simply split in half, so that one half was used for identification (*Step 4*) and the other half for validation.

6. RESULTS

Before conducting the experiment on a real system, identification was carried out on a theoretically obtained mathematical model implemented on a LabVIEW platform. This was very useful to get acquainted with the steps of the identification procedure and the obtained results were very good (not shown in this paper).

After successful identification of the LabVIEW model, experiments on the real system could begin. It is worth mentioning that although transfer function from wind speed to turbine rotational speed, given by (7), would be very useful for predictive control system design, it could not be identified because of the fan protection system. Namely, a ramp filter has been put on the fan speed reference, so that the fan motor is protected from sudden speed changes. Therefore, it is not possible to produce the wind speed in the form of a PRBS signal.

$$G_v(z) = \frac{\omega(z)}{v_{wind}(z)} \tag{7}$$

During the experiment significant effort was needed to bring the system to the desired operating point and especially to achieve that the system does not drift away from the operating point once the controllers are turned off and the PRBS signal is imposed on the system inputs. The reason for this lies in nonlinearity of the system and also in a fact that the system is very sensitive to outer wind speed (wind that normally blows in the surroundings) and on rare occasions was the weather completely steady. Solution has been found in experimenting with PRBS parameters until the system was able to stay around the desired operating point long enough to record enough data, i.e. to achieve at least M=5 in relation (5). Although PRBS parameters were found by trial-and-error method, their initial value was determined from the recommendations given in Section 5 (*Step 2*). Specific parameters of the used PRBS signals will be given in the sequel for each particular operating point.

Sample time has also been determined along the guidelines from Section 5. It is fixed, i.e. it does not change with the operating point and it amounts 20 ms. It is also worth noting that all the identification experiments were conducted with stiff laboratory turbine tower, because the system of springs and dampers, which allows tower oscillations, was not yet fully functional at the time of experiments.

All identified linear state-space models are given in the following discrete-time form:

$$\begin{aligned} x(k+1) &= A \cdot x(k) + B \cdot u(k) + K \cdot e(k), \\ y(k) &= C \cdot x(k) + D \cdot u(k) + e(k). \end{aligned} \tag{8}$$

Here *x(k)* is a state vector. Dimension of *x(k)* corresponds to the model order. *u(k)* and *y(k)* are system input and output respectively. *K* is a matrix which models influence of the noise to the system and *e(k)* is process noise (white noise which is generated and used by the identification software). *A*, *B*, *C* and *D* are standard matrices used in the state-space form.

Free parameterization of the state-space matrices was used, which means that any elements in the matrices are adjustable by identification algorithm. Therefore, a basis for the state-space realization is automatically selected to give well-conditioned calculations (MathWorks 2010).

Based on the obtained input-output data, models were also identified in parametric forms other than state-space, e.g. ARX, ARMAX, BJ and OE. Identified models of different forms were mutually compared by validation and they all gave similar results (not shown in this paper).

Identification results for each of the 4 operating points are presented in the sequel. Since state-space models were identified results are presented in the form of state-space matrices, but also in the form of corresponding transfer functions. Models presented in Subsections 6.1 and 6.2 correspond to the transfer function (1), while models in 6.3 and 6.4 correspond to (2) (see Figure 5). For every operating point two figures are given. First one displays recorded input-output data, while the second one displays validation results. In all the figures mean values of the signals have been removed, but they are given in tables along with PRBS parameters used. Models of different orders were obtained and the ones that gave the lowest deviation from the real system are displayed. It is worth noting that responses of three identified models in Figure 10 overlap. By plotting pole/zero maps (not shown), it has been found that all the identified models are stable. It can be concluded that satisfactory results have been obtained, despite all the problems that are always present on real systems, e.g. nonlinearities, measurement noise, sensitivity to disturbances etc.

Additionally, pitch servo drive identification has been performed and the results are given in Subsection 6.5. In Figure 18 responses of the real pitch servo drive and the identified model are compared. The model describes the real system satisfactory and matching of the responses is good.

It should be pointed out that the identified models from Subsections 6.3 and 6.4 include the dynamics of the pitch servo drive. Since model (15) has order 3 and the pitch servo drive (17) has order 2, it can be argued how can the remaining part of the system (from β to ω, see Figure 5) be modeled with only first order. From (Jelavić et al. 2009) it can be seen that this is reasonable because the identification has been performed with the stiff tower.

6.1 Identification of $G_T(z)$ with v_{wind} = 7 m/s

Values of the process variables		PRBS parameters	
T_{mean} =	3.5 Nm	*c* =	0.8 Nm
β_{mean} =	0° (const)	*N* =	16
ω_{mean} =	183 rpm	*Δt* =	1 s

Table 1. Identification parameters for v_{wind} = 7 m/s.

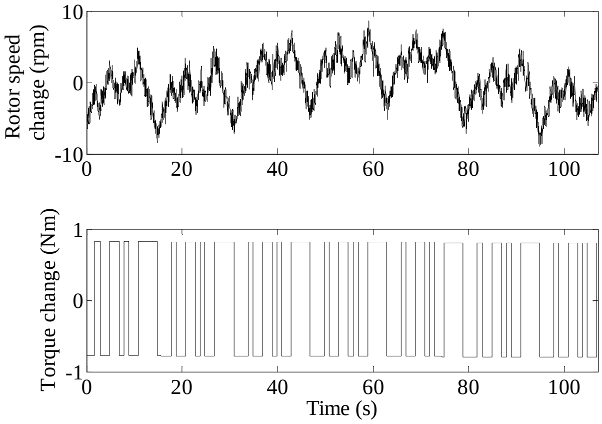


Fig. 9. Output and input signals for v_{wind} = 7 m/s.

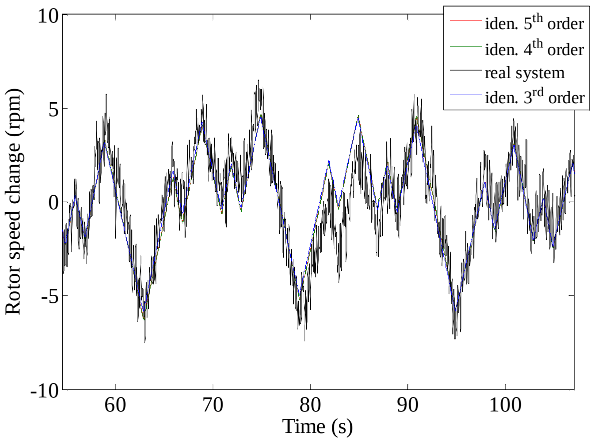


Fig. 10. Turbine rotational speed responses for v_{wind} = 7 m/s.

Identified 3rd order state-space model of structure (8) reads:

$$
\begin{aligned}
A &= \begin{bmatrix} 0.99559 & 0.046808 & -0.065886 \\ -0.054331 & 0.63367 & -0.64621 \\ 0.012755 & 0.72281 & 0.7296 \end{bmatrix}, \\
B &= \begin{bmatrix} -0.00021549 \\ 0.0018215 \\ -0.0017577 \end{bmatrix}, \quad K = \begin{bmatrix} 0.003357 \\ 0.016101 \\ -0.00023999 \end{bmatrix}, \\
C &= \begin{bmatrix} 371.12 & 6.5785 & -11.542 \end{bmatrix}, \\
D &= \begin{bmatrix} 0 \end{bmatrix}.
\end{aligned}
\tag{9}
$$

Corresponding transfer function reads:

$$
G_{T1}(z) = \frac{-0.0477z^{-1} + 0.1223z^{-2} - 0.1081z^{-3}}{1 - 2.359z^{-1} + 2.29z^{-2} - 0.9299z^{-3}}. \tag{10}
$$

6.2 Identification of $G_T(z)$ with v_{wind} = 9 m/s

Values of the process variables		PRBS parameters	
T_{mean} =	4.9 Nm	*c* =	0.8 Nm
β_{mean} =	0° (const)	*N* =	16
ω_{mean} =	213 rpm	*Δt* =	1 s

Table 2. Identification parameters for v_{wind} = 9 m/s.

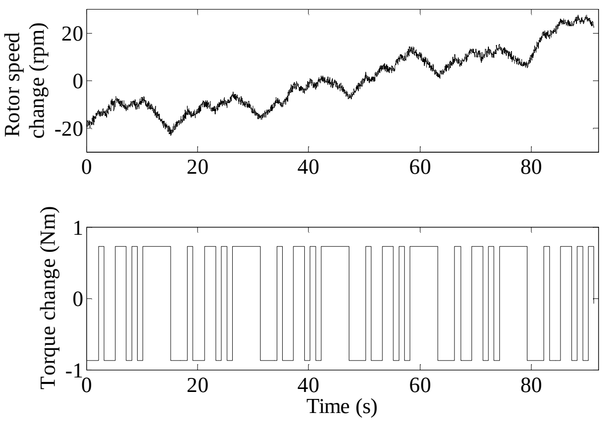


Fig. 11. Output and input signals for v_{wind} = 9 m/s.

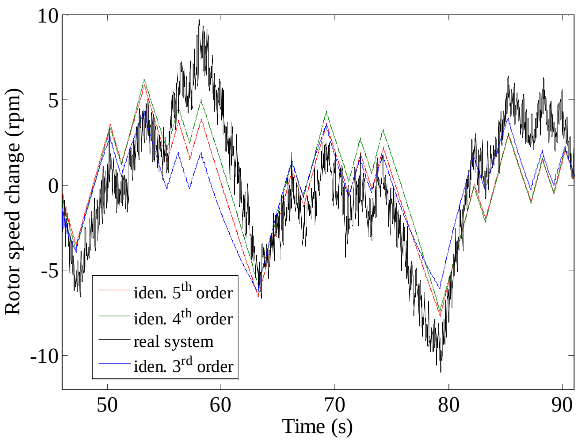


Fig. 12. Turbine rotational speed responses for v_{wind} = 9 m/s.

Identified 4th order state-space model of structure (8) reads:

$$
\begin{aligned}
A &= \begin{bmatrix} 0.98208 & -0.16176 & -0.061514 & -0.068064 \\ 0.13113 & 0.51558 & 0.80396 & -0.29476 \\ -0.069255 & -0.73422 & 0.61965 & 0.25423 \\ -0.034816 & 0.13938 & 0.12753 & -0.19436 \end{bmatrix}, \\
B &= \begin{bmatrix} 0.00028043 \\ 0.00051794 \\ 0.0014344 \\ 0.002622 \end{bmatrix}, \quad K = \begin{bmatrix} 0.0073324 \\ -0.016026 \\ 0.010136 \\ 0.019012 \end{bmatrix}, \\
C &= \begin{bmatrix} 181.66 & -12.455 & -5.863 & -2.6417 \end{bmatrix}, \\
D &= \begin{bmatrix} 0 \end{bmatrix}.
\end{aligned}
\tag{11}
$$

Corresponding transfer function reads:

$$
G_{T2}(z) = \frac{0.02916z^{-1} - 0.0843z^{-2} + 0.04038z^{-3} - 0.03675z^{-4}}{1 - 1.923z^{-1} + 1.636z^{-2} - 0.5914z^{-3} - 0.1215z^{-4}}. \tag{12}
$$

6.3 Identification of $G_\beta(z)$ with v_{wind} = 11 m/s

Values of the process variables		PRBS parameters	
T_{mean} =	6.8 Nm (const)	c =	2^o
β_{mean} =	5.6^o	N =	32
ω_{mean} =	240 rpm	Δt =	2 s

Table 3. Identification parameters for v_{wind} = 11 m/s.

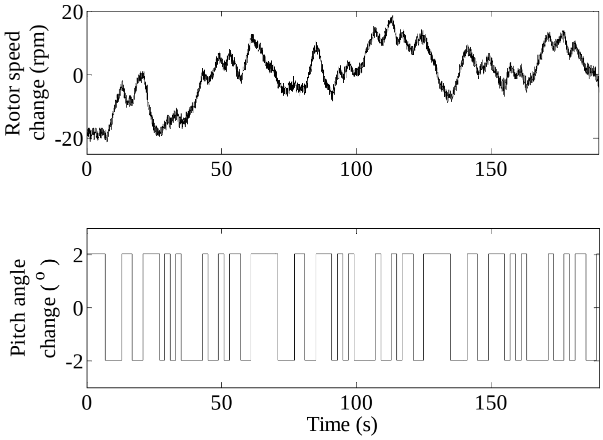


Fig. 13. Output and input signals for v_{wind} = 11 m/s.

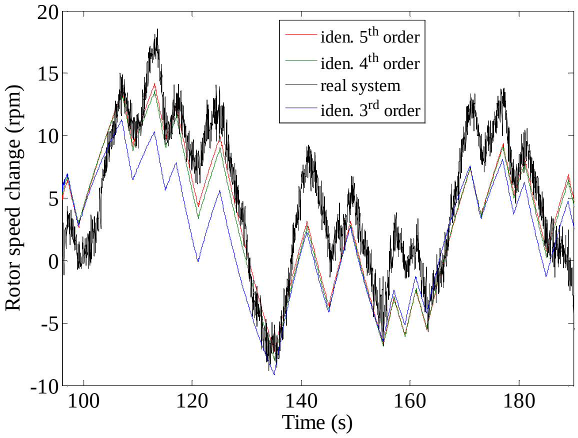


Fig. 14. Turbine rotational speed responses for v_{wind} = 11 m/s.

Identified 4th order state-space model of structure (8) reads:

$$A = \begin{bmatrix} 0.99592 & -0.068805 & 0.0048254 & 0.048758 \\ 0.045489 & 0.40068 & -0.90725 & -0.097681 \\ 0.026732 & 0.79103 & 0.30208 & 0.53587 \\ 0.036769 & 0.12396 & 0.36363 & -0.3416 \end{bmatrix},$$
$$B = \begin{bmatrix} 3.6982e-05 \\ -0.00014468 \\ 0.0013955 \\ -0.0024389 \end{bmatrix}, \quad K = \begin{bmatrix} 0.0018945 \\ -0.0060099 \\ -0.005948 \\ -0.010702 \end{bmatrix}, \qquad (13)$$
$$C = \begin{bmatrix} 659.29 & -17.365 & 6.3086 & 12.757 \end{bmatrix},$$
$$D = [0].$$

Corresponding transfer function reads:

$$G_{\beta 1}(z) = \frac{0.004584z^{-1} - 0.01992z^{-2} + 0.04688z^{-3} - 0.05696z^{-4}}{1 - 1.357z^{-1} + 0.7768z^{-2} + 0.03354z^{-3} - 0.4519z^{-4}}. \qquad (14)$$

6.4 Identification of $G_\beta(z)$ with v_{wind} = 13 m/s

Values of the process variables		PRBS parameters	
T_{mean} =	6.8 Nm (const)	c =	2^o
β_{mean} =	10.7^o	N =	32
ω_{mean} =	240 rpm	Δt =	2 s

Table 4. Identification parameters for v_{wind} = 13 m/s.

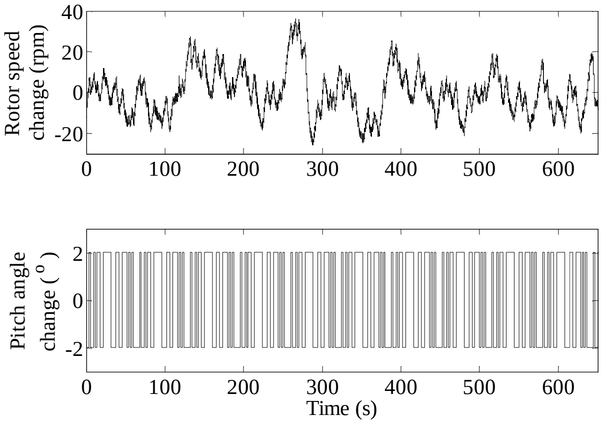


Fig. 15. Output and input signals for v_{wind} = 13 m/s.

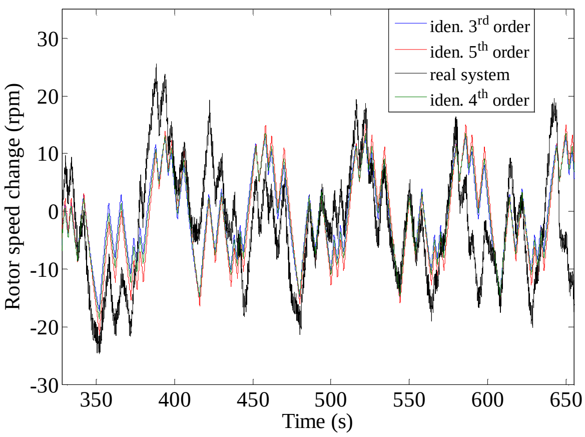


Fig. 16. Turbine rotational speed responses for v_{wind} = 13 m/s.

Identified 3rd order state-space model of structure (8) reads:

$$A = \begin{bmatrix} 0.99693 & -0.060215 & -0.026464 \\ 0.043779 & 0.45433 & 0.83324 \\ -0.0253 & -0.79544 & 0.55406 \end{bmatrix},$$
$$B = \begin{bmatrix} -9.1179e-006 \\ -0.00012523 \\ 6.5033e-005 \end{bmatrix}, \quad K = \begin{bmatrix} 0.00067204 \\ -0.0030013 \\ 0.0028048 \end{bmatrix}, \qquad (15)$$
$$C = \begin{bmatrix} 1969.8 & -45.602 & -28.418 \end{bmatrix},$$
$$D = [0].$$

Corresponding transfer function reads:

$$G_{\beta 2}(z) = \frac{-0.0141z^{-1} + 0.01811z^{-2} - 0.02716z^{-3}}{1 - 2.005z^{-1} + 1.922z^{-2} - 0.9151z^{-3}}. \qquad (16)$$

6.5 Identification of the pitch servo drive

Task of the pitch servo drive is to position the rotor blades into the angular position required by the outer loop (see Figure 5). Dynamics of the pitch servo drive does not depend on the operating point (wind speed) and it turns out that it can be described by a second-order transfer function. Input-output data are shown in Figure 17. These data were recorded during identification experiment described in Subsection 6.4. Therefore all the parameters of the experiment are given in Table 4. above.

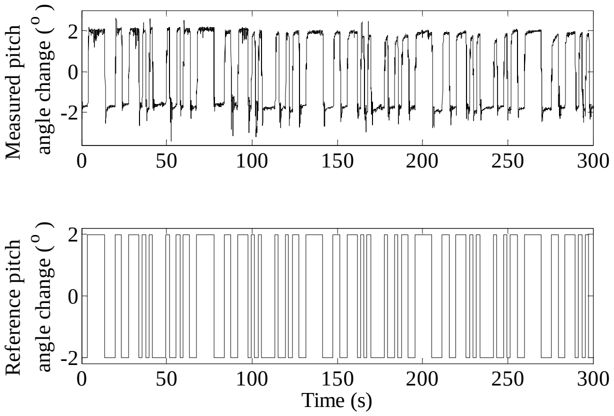


Fig. 17. Output and input signals of the pitch servo drive.

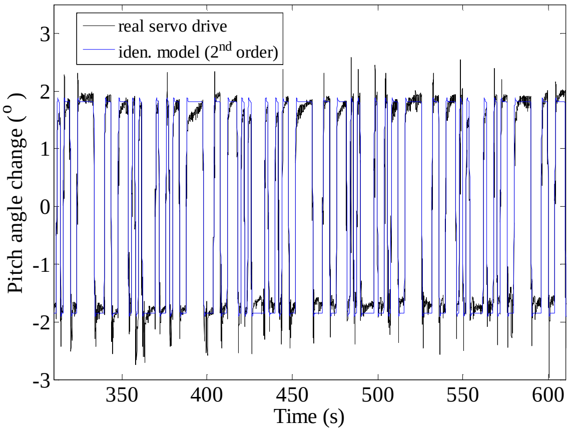


Fig. 18. Pitch servo drive responses.

Identified transfer function of the pitch servo drive is:

$$G_{servo}(z)=\frac{\beta(z)}{\beta_{ref}(z)}=\frac{0.006363z^{-1}+0.004433z^{-2}}{1-1.876z^{-1}+0.8881z^{-2}}. \quad (17)$$

7. CONCLUSION

Laboratory for Renewable Energy Sources on Faculty of Electrical Engineering and Computing, University of Zagreb enables experimental research and development of various advanced control algorithms for renewable energy sources. These algorithms can be used to enhance the electrical energy production from wind, sun and hydrogen, but also to reduce losses in a process of producing hydrogen which is used for storing energy. Laboratory wind turbine setup, which has all the important characteristics of the modern MW-class wind turbines, is described. Identification of the laboratory wind turbine mathematical model has been performed and satisfactory results have been obtained. Reliable mathematical model of a process is a basis for successful control system design.

ACKNOWLEDGMENTS

This work has been financially supported by *The National Foundation for Science, Higher Education and Technological Development of the Republic of Croatia* and *Končar – Electrical Engineering Institute*.

REFERENCES

Bobanac V., M. Jelavić and N. Perić (2010). Linear Parameter Varying Approach to Wind Turbine Control, *Proceedings of the 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010.*, 8 pages, Ohrid, Republic of Macedonia.

Brekalo M. (2010). *Mathematical Model Identification of the Laboratory Wind Turbine*, Master thesis, Faculty of Electrical Engineering and Computing, Zagreb, in Croatian.

Brestovec B. (2005). *Modeling Wind Turbines for the Purpose of Designing a Control System*, Master thesis, Faculty of Electrical Engineering and Computing, Zagreb, in Croatian.

Burton T., D. Sharpe, N. Jenkins and E. Bossanyi (2001). *Wind Energy Handbook*, John Wiley & sons, Chichester, UK.

European Wind Energy Association (2011). http://www.ewea.org/.

Jelavić M. and N. Perić (2009). Wind Turbine Control for Highly Turbulent Winds, *Automatika*, **Vol. 50**, No. 3-4, pp. 135-151.

Jelavić, M., V. Petrović and N. Perić (2008). Individual pitch control of wind turbine based on loads estimation, *Proceedings of the 34th Annual Conference of the IEEE Industrial Electronics Society (IECON 2008)*, pp. 228-234, Orlando, USA.

MathWorks (2010). *System identification Toolbox 7*, User's Guide.

National Instruments (2007). *Getting Started with LabVIEW*, User’s Guide.

Perić N. and I. Petrović (2005). *Process identification*, Lecture notes, Faculty of Electrical Engineering and Computing, Zagreb, in Croatian.

Perić, N, Ž. Ban, M. Jelavić, B. Matijašević, S. Mikac, M. Kostelac and H. Domitrović (2010). *Wind turbine control research in Laboratory for renewable energy sources*, 7 pages, European Wind Energy Conference and Exhibition, EWEC, Warsaw, Poland.

Petrović, V, N. Hure and M. Baotić (2010). *Appliance of LabVIEW software for development of HIL wind turbine speed control structure*, 6 pages, MIPRO, Opatija, Croatia, in Croatian.

Skogestad S. and I. Postlethwaite (2005). *Multivariable Feedback Control*, John Wiley & Sons Ltd, Chichester, UK.

Mathematical Modeling and Implementation of the Airship Navigation

F. Jelenčiak*

** FernUniversität in Hagen, MI / PRT*
(Tel: +49 2331 / 987 1119.; e-mail: Frantisek.Jelenciak@fernuni-hagen.de)

Abstract: The main source of errors for airship navigation is that the airship body is not solid. For this reason a standard fixed calibration for a navigation system is not the best solution. This article provides an overview of the proposed navigation system for airships with compensation of errors due to resilience.

1. INTRODUCTION

Navigation systems and flight control are basic equipments of unmanned aerial systems (UAS). For modeling of the navigation system (see references) it is necessary to select appropriate coordinate systems and to select description methods for body orientation. The next step is to apply transformations between coordinate frames, and to relate them to kinematics and dynamics theory. Stability and reliability of the navigation system can be achieved by combining the on-board IMU (Inertial Measurement Unit) with external observation units such as GPS (Global Positioning System), barometric and magneto-compass units.

The proposed model of airship navigation uses coordinate systems like ECEF (Earth-Centered-Earth-Fixed) and NED (North-East-Down). For the description of the airship's orientation Euler angles as well as roll, pitch, and yaw are used. All measured data are joined together by a special type of direct Kalman filter. This filter and the strapdown modification are designed in such a way that together with other measurement units it is possible to compensate the problem, which is caused by the non-solid body of airships. Navigation systems implemented on airships show characteristic errors due to this fact. The following facts can be considered as sources of these errors: an airship body changes its shape depending on several parameters, e.g. the helium pressure in the hull; the ambient temperature; relative position of inner and outer hull; aerodynamic forces. The shape changes are in general nonlinear. For these reasons a fixed calibration for navigation system is not the optimal solution. This article provides an overview of the proposed navigation system for the airship which compensates the aforementioned errors.

The development of the airship navigation algorithm was carried out in the programming environment Matlab-Simulink. The toolboxes used are "Target Support Package" and "Code Composer Studio" was designed Simulink-scheme, which is algorithm interpretation of the airship model navigation. This Simulink-scheme can be transferred as program code into the microprocessor TMS320F28335, which together with sensors of the IMU (Inertial Measurement Unit), barometric unit, GPS, and magneto-compass are core of the navigation system (http://prt.fernuni-hagen.de/ARCHIV/2010/fernsehen_2010.html, http://www.derwesten.de/staedte/hagen/FernUni-Luftschiff-auf-Minensuche-id2359263.html).

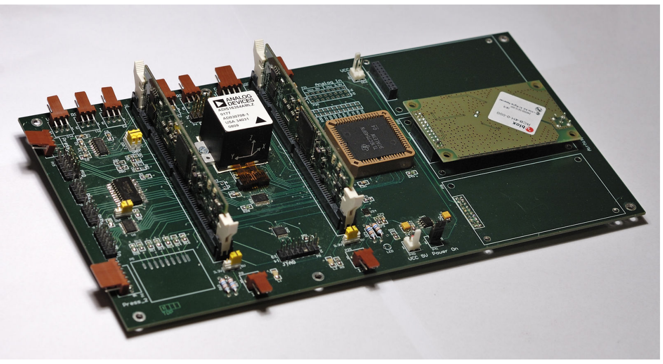


Fig. 1. Electronic board of the navigation system (design by Dr.-Ing. Ivan Masár*).

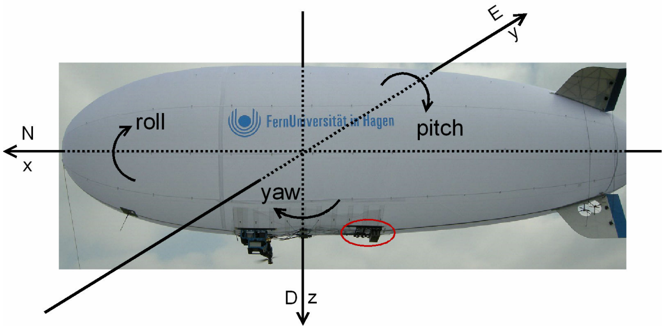


Fig. 2. Airship photo with selected area for storing of the navigation system (two hulls, length 9m, diameter 2,5m, volume 27m^3).

The article is structured as follows: In chapter 2, the basic scheme of the navigation algorithm is shown. In chapters 3-7, the essential parts of the navigation algorithm are described in detail. In the final chapter 8, conclusions and results are given.

2. BASIC SCHEME

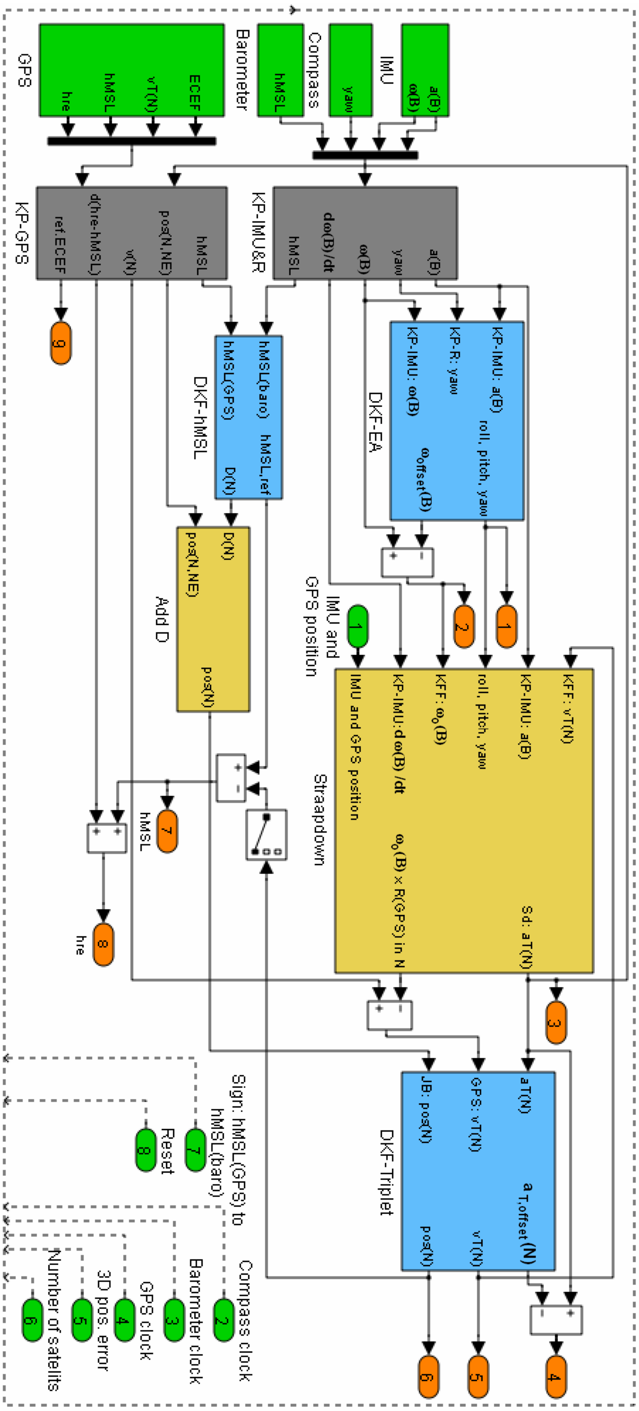


Fig. 3. Basic scheme of the navigation algorithm.

At the beginning it is necessary to divide the input navigation data into two groups. The first group represents data from the IMU. The second group represents data from other sensors.

From the IMU (strapdown platform), it is possible to load data such as acceleration and angular velocity of rotation. The advantage of these data is that they may be available at high sampling rates with relatively high sensitivity and accuracy. The disadvantage of these data is, that velocity, position or orientation determined by direct time integration are loaded with error, which in time always increases.

A second group is represented by data from GPS, barometer, and magneto-compass. The advantage of these data is, that the error with which they are loaded is not dependent on time. The disadvantage of these data is that they often show less accuracy, sensitivity, and their availability with a smaller sampling rate. Advantages of both groups can be obtained by some form of Kalman filter.

3. SIGNALS AND BLOCKS

In Fig. 3, all inputs into the navigation algorithm are marked green. These are signals, which are directly necessary for the navigation algorithm. Output signals are marked in orange. Algorithms which use pre-filter based on Kalman filter are marked in gray. Algorithms which use direct Kalman filters are marked in blue. Other blocks in which partial calculations are carried out are marked with yellow or white. The argument B represents the body frame and the argument N represents the navigation frame. Both frames have orientation as NED coordination system.

4. ALGORITHMS WITH KALMAN PRE-FILTER

In Fig. 3, two blocks are using Kalman pre-filters: “KP-IMU&R” and “KP-GPS”. Block “KP-IMU&R” processes two types of data: direct data from the IMU, which are represented by the body acceleration $\mathbf{a}(\mathrm{B})$ and the body angular velocity $\boldsymbol{\omega}(\mathrm{B})$, and data from the magneto-compass (angle yaw γ_{R}) and barometer (h_{MSL} – height above mean sea level). Task of this block is to make Kalman signal pre-filtering. By this pre-filter also the value of angular acceleration $\dfrac{d\boldsymbol{\omega}(\mathrm{B})}{dt}$ is calculated.

Block “KP-GPS” processes data from GPS and strapdown algorithm. This block calculates the following values: h_{MSL}, position and velocity in the navigation frame N, the difference between the height of the reference ellipsoid h_{RE} and h_{MSL}, and the ECEF reference point which defines zero of the navigation frame N. These parameters are calculated using Kalman filtering. The ratio between the covariance matrix of process noise and the covariance matrix of measurement noise is dependent on the total translational acceleration of the body in the navigation frame N $\mathrm{Sd}:\mathbf{a}_{\mathrm{T}}(\mathrm{N})$ determined by strapdown algorithm.

5. ALGORITHMS WITH DIRECT KALMAN FILTER

In Fig. 3, three blocks are applying direct Kalman filters: “DKF-EA”, “DKF-hMSL”, and “DKF-Triplet”.

Block “DKF-EA” processes two types of data:

- data from the IMU as total acceleration $\mathbf{a}(\mathrm{B})$ and angular velocity of rotation $\boldsymbol{\omega}(\mathrm{B})$;

- reference data about the azimuth γ_{R} from the magneto-compass.

In this block, reference values of Euler angles are computed based on the azimuth from the magneto-compass and acceleration values from the accelerometer. These reference values of Euler angles are used for estimation of the real Euler angles using a direct Kalman filter. The mathematical model of the direct Kalman filter is divided into two parts:

- direct computation of Euler angles using the RM (rotation matrix) with respect to the sequence of rotation roll, pitch, and yaw;

- computation of Euler angles using quaternions.

For smaller values of Euler angles, direct computation using the rotation matrix is considered. For larger values of Euler angles, computing over quaternions is applied.

The direct Kalman filter works as follows: if reference data in the actual sample time interval are not known, the filter performs angular velocity integration, resulting in Euler angles or quaternions. If reference data in the actual sample time interval are known, the filter performs estimation and computed offset of angular velocity. This offset of angular velocity is then subtracted of the angular velocity

$$\boldsymbol{\omega}_{\mathbf{O}}(\mathrm{B}) = \boldsymbol{\omega}(\mathrm{B}) - \boldsymbol{\omega}_{\mathbf{offset}}(\mathrm{B}) .$$

Block “DKF-hMSL” processes two types of data:

- height above mean sea level from barometer $h_{\mathrm{MSL}}(\mathrm{baro})$;

- height above mean sea level from GPS $h_{\mathrm{MSL}}(\mathrm{GPS})$

In this case, $h_{\mathrm{MSL}}(\mathrm{GPS})$ is the reference signal. The reason is, that GPS can operate in several modes (for example differential GPS). The accuracy of $h_{\mathrm{MSL}}(\mathrm{GPS})$ is generally dependent on many factors. For this reason auxiliary variables as $3D_{\mathrm{ERR}}$ (3D position error), N_{S} (number of the satellites) and S_{GPS2B} ($h_{\mathrm{MSL}}(\mathrm{baro})$ value is replaced by $h_{\mathrm{MSL}}(\mathrm{GPS})$ value with respect to weight of both values) are used. According to the value of the logical variable S_{GPS2B} the direct Kalman filter can perform the estimation. If the estimation, then the direct Kalman filter carried out determines also the offset $h_{\mathrm{MSL,offset}}$, which is subtracted of the $h_{\mathrm{MSL}}(\mathrm{baro})$. The result $d = h_{\mathrm{MSL}}(\mathrm{baro}) - h_{\mathrm{MSL,offset}}$ is then subsequently subtracted from the reference $h_{\mathrm{MSL,ref}}$. $h_{\mathrm{MSL,ref}}$ corresponds with the ECEF reference point.

The final result from this block is then an estimated value $D = h_{\mathrm{MSL,ref}} - d = h_{\mathrm{MSL,ref}} - h_{\mathrm{MSL}}(\mathrm{baro}) + h_{\mathrm{MSL,offset}}$.

Under certain assumptions, this value of D may replace the D value in navigation frame N computed by GPS. The advantage of this method is that the value of D is known from barometric measurements even if no GPS signal is available. This method divides one position vector in the navigation frame N into two independent parts: part NE and part D.

Block “DKF-Triplet” processes three types of data:

- translational acceleration $\mathbf{a_T}(\mathrm{N})$ computed from strapdown algorithm;

- velocity in navigation frame N $\mathbf{v}(\mathrm{N})$ from GPS;

- position in the navigation frame N obtained as combination of elements NE of the position vector N from GPS with element D from “DKF-hMSL” block.

This block uses a direct Kalman filter as estimator in the following cases:

- in the current sampling period a reference vector NED or only NE of the position in frame N is known, or

- in the current sampling period a reference vector of the translational velocity from GPS $\mathbf{v_T}(\mathrm{N})$ is known, or

- in the current sampling period only a reference value D as one element from position vector in navigation frame N is known.

If in the current sampling period no reference value of the position in frame N as N or E or D, or $\mathbf{v_T}(\mathrm{N})$ is known, the direct Kalman filter works as integrator for translational acceleration $\mathrm{Sd}:\mathbf{a_T}(\mathrm{N})$ from the strapdown algorithm. If the direct Kalman filter works as estimator, then it computes also the acceleration offset $\mathbf{a_{T,offset}}(\mathrm{N})$, with

$$\mathbf{a_{TO}}(\mathrm{N}) = \mathbf{a_T}(\mathrm{N}) - \mathbf{a_{T,offset}}(\mathrm{N}) .$$

6. STRAPDOWN ALGORITHM

The mathematical model of the strapdown algorithm, which is used for airship navigation purpose, expresses the following relationship

$$\mathbf{a}(\mathrm{B}) = \frac{d\boldsymbol{\omega}(\mathrm{B})}{dt} \times \mathbf{R_{IMU}} + \boldsymbol{\omega}(\mathrm{B}) \times \mathbf{v_T}(\mathrm{B}) + \\ + \boldsymbol{\omega}(\mathrm{B}) \times \left(\boldsymbol{\omega}(\mathrm{B}) \times \mathbf{R_{IMU}}\right) - \mathbf{g}(\mathrm{B}) + \mathbf{a_T}(\mathrm{B}) + \mathbf{a_R}(\mathrm{B})$$

where

$\mathbf{a}(\mathrm{B})$ is the measured acceleration vector; $\mathbf{v_T}(\mathrm{B})$ is the translational velocity vector of the body; $\mathbf{R_{IMU}}$ is the IMU position on the body of the turn-point, $\mathbf{g}(\mathrm{B})$ is the gravitation vector in the body frame, $\mathbf{a_T}(\mathrm{B})$ is the translational vector of the body; $\mathbf{a_R}(\mathrm{B})$ is the residual acceleration vector of others forces in the body frame which are not include in the strapdown algorithm.

In this case in the strapdown algorithm is $\mathbf{a_R}(\mathrm{B}) = 0$, because errors of unmodeling acceleration are solving by $\boldsymbol{\omega}_{\mathbf{offset}}(\mathrm{B})$, $h_{\mathrm{MSL,offset}}$ and $\mathbf{a_{T,offset}}(\mathrm{N})$. The reason why it is possible to accept this assumption is,that all errors caused by this assumption are small in relation to errors caused by the behaviour of the airship body hull.

In case of $\mathbf{a_R}(\mathrm{B})=0$, the strapdown equation is the equation of the „Flat Earth Navigator“.

In the strapdown block there is also the computation of $\left(\boldsymbol{\omega}_\mathbf{O}(\mathrm{B})\times\mathbf{R}(\mathrm{GPS})\right)_\mathrm{N}$, which defines the transformation to the navigation frame N with $\mathbf{R}(\mathrm{GPS})$ as GPS antenna position on the airship body of the airship turn-point. This part is necessary to subtract of the $\mathbf{v_T}(\mathrm{N})$ obtained from GPS reciever.

Main output from the strapdown algorithm is the translational acceleration in the navigation frame N: $\mathbf{a_T}(\mathrm{N})$.

7. OTHER BLOCKS

In the ‘other’ blocks of the algorithm (see figure 3), auxiliary inputs and characteristic outputs of the navigation algorithm are calculated.

The clocks for magneto-compass, barometer, and GPS are used as auxiliary inputs. The logical value of these signals is ONE if magneto-compass, barometer, or GPS have an actual value in the actual sample time. This is important for decision of the Kalman filtering.

The following parameters are outputs from the navigation algorithm:

- roll, pitch, and yaw angles;
- the angular velocity $\boldsymbol{\omega}_\mathbf{O}(\mathrm{B})$ or $\boldsymbol{\omega}_\mathbf{O}(\mathrm{N})$, adjusted by the angular velocity offset $\boldsymbol{\omega}_\mathbf{offset}(\mathrm{B})$;
- the translational acceleration from the strapdown algorithm $\mathrm{Sd}:\mathbf{a_T}(\mathrm{N})$;
- the translational acceleration $\mathbf{a_T}(\mathrm{N})$ which is result with respect to all measurement units and which is adjusted by offset $\mathbf{a}_\mathbf{T,offset}(\mathrm{N})$;
- the translational velocity $\mathbf{v_T}(\mathrm{N})$;
- the position $\mathbf{pos}(\mathrm{N})$;
- the height above mean sea level h_MSL ;
- the height above the reference ellipsoid h_RE ;
- the ECEF reference point

8. CONCLUSIONS AND RESULTS

To show the results, a comparison of a commercial navigation system with a system that was developed at our department is accomplished. The latter is used as one part this algorithm (the full mathematical model of the navigation algorithm can not be presented here due to limited space). For this comparison we used a commercial navigation system in the price range of 4000 €. For a comparison, 25 journeys by car were made. The mounting of both navigation systems were done in a way that the conditions were as similar as possible as on an airship hull (see figure 4).

Both navigation systems are mounted on a special fixture which is placed in the trunk of a car. This fixture consists of a top platform, a lower platform, and four foam springs. The navigation systems are mounted on the top platform. The lower platform is based in the trunk. When the car is running, the springs cause movements of the top platform contrary to the movement of the car. This is the main objective of the fixture. This mechanism emulates an airship with implemented navigation system. The airship hull is a non-solid body with spring effects. Another reason why the comparison was done on the car is, that at the time of article writing our navigation system did not offer a DGPS mode. Driving a car and matching with Google maps is an alternative strategy.

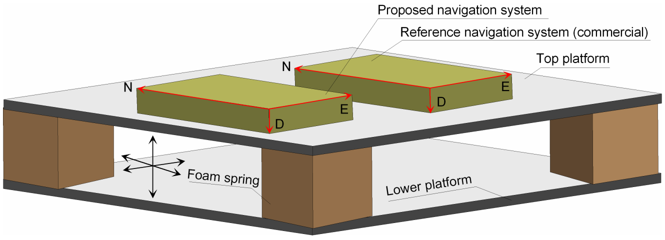

Proposed navigation system
Reference navigation system (commercial)
Top platform
N
N
E
E
D
D
Foam spring
Lower platform

Fig. 4. Fixture of the navigation systems in the car.

Figures 5 and 6 show Google maps with plotted averaged paths (from 25 journeys by car) obtained from the navigation system. The real paths are driven with respect to the traffic rules. In figures 5 and 6, two characteristic points are marked. Point 1 is located at traffic lights. Point 2 is a parking position. From figures 5 and 6, it is easy to see that if the car stopped the navigation algorithm begins "walking". This is caused by the GPS data, which are obtained in normal user mode. If the body is in motion, it is possible to eliminate this “walking” of the GPS by taking into account the orientation of the body (car).

The precision of the proposed navigation system on the defined path is better than the commercial navigation system (4000 €) about from 0,5 to 2,5m.

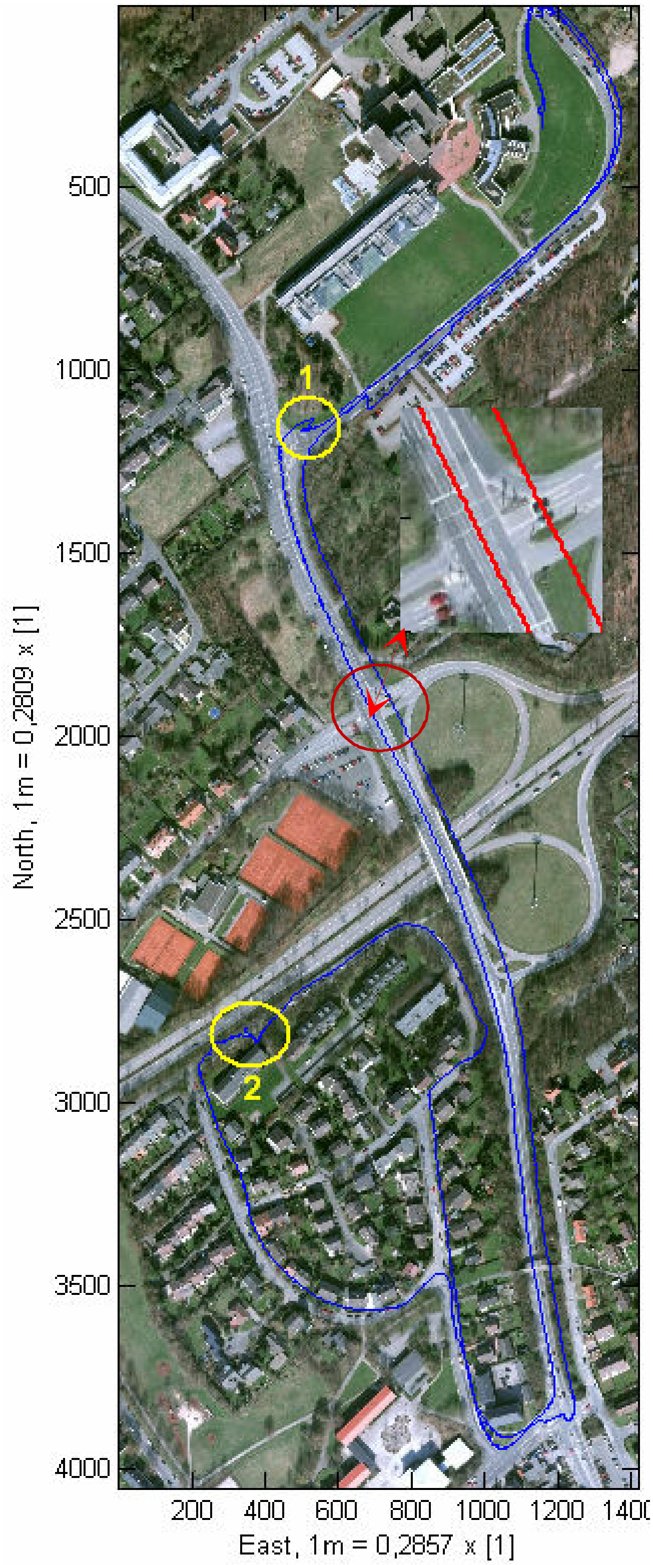


Fig. 5. Reference navigation system (commercial).

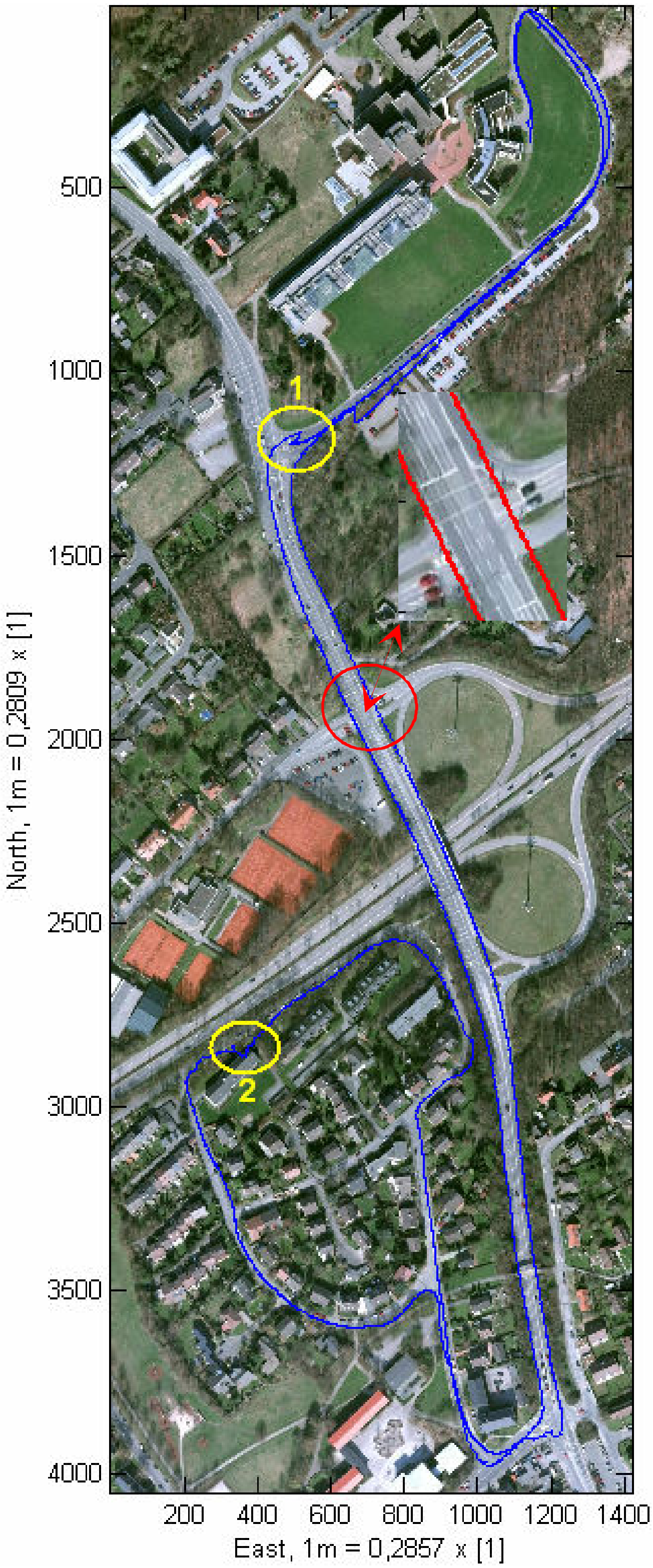


Fig. 6. Proposed navigation system.

REFERENCES

Brant, A., and J.F. Gardner (1998). Constrained navigation algorithms for strapdown inertial navigation systems with reduced set of sensors. In: *American Control Conference.* **Vol. 3**. pp. 1848–1852.

Bose, C. Sam (2000). *GPS/INS Integrated Navigation Systems.* Technalytics Inc. Cangoa Park Kalifornien, USA.

Barbour, N. and G. Schmidt (2001). Inertial Sensor Technology Trends. In: *IEEE Sensors Journal.* **Vol. 1**. pp. 332-339.

Britting, K.R. (1971). *Inertial Navigation System Analysis.* Wiley Interscience, New York.

Titterton, D. (2000). Strapdown inertial navigation technology. In: *The American Institute of Aeronautics and Astronautics and the Institution of Electrical Engineers.*

Foppe, K. (2001). Kombination von inertialen und satellitengestützten Beobachtungsverfahren zur ingenieurgeodätischen Überwachung von Brückenbauwerken. In: *Wissenschaftliche Arbeiten der Fachrichtung Vermessungswegen der Universität Hannover.* Nr. 242, Hannover.

Dissanayake, G., S. Sukkarieh, E. Nebot and H. Durrant – Whyte (2001). The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications. In: *IEEE Transactions on Robotics and Automation*. 17(5). pp. 731–747.

Farrell, J. and M. Barth (1998). *The Global Positioning System and Inertial Navigation.* 1 edition. McGraw-Hill Professional.

Mohinder, S.G., R.L. Weill and A.P. Andrews (2001). *Global Positioning Systems, Inertial Navigation and Integration.* John Wiley and Sons.

Woodman, O.O.J. (2007). *An introduction to inertial navigation.* Technical Report 696. University of Cambridge.

Zhu, J. (1994). Conversion of Earth-centered Earth-fixed coordinates to geodetic coordinates. In: *Aerospace and Electronic Systems, IEEE Transactions.* **Vol. 30**. pp. 957-961.

Robust decentralized controller design for 3D crane

Nguyen Quang Thuan and Vojtech Veselý *

**Slovak University of Technology, Faculty of Electrical Engineering and Information Technology*
Institute of Control and Industrial Informatics
Ilkovičova 3, 812 19 Bratislava, Slovak Republic

(tel.: +421 2 602 91 544, e-mail: thuan.quang@stuba.sk, vojtech.vesely@stuba.sk)

Abstract: The subject of this paper is to design robust decentralized PID controllers for the 3Dcrane to stabilize motion of the cart along axes-x, axes-y using the Small Gain Theorem, and Parameter Dependent Lyapunov Functional (PDLF) in time domain. The obtained results were evaluated and verified in the Matlab simulink and on the real model of the 3DCrane.

Keywords: decentralized control, PID controller, Parameter Dependent Lyapunov Functional (PDLF)

1. INTRODUCTION

The industrial crane model 3DCrane is one of the real processes built for control education and research at Department of Information Engineering and Process Control. The 3DCrane is a nonlinear electromechanical MIMO system having a complex dynamic behavior and creating challenging control problems as nonlinear, interactions between subsystems corresponding to motion of cart along the axes-x and axes-y, length of the payload lift-line dynamic changed … The technical equipments allow us to realize control crane by classical and advanced control method (INTECO Ltd.: 3DCrane User's manual).

The main aim of paper is to use knowledge of multivariable (Multi Input and Multi Output) system and stabilization of decentralized control systems (Zhining et al. 1997), knowledge about robust control of linear systems in the frequency domain (Veselý et al. 2006) and in the time domain to design robust PID/PD decentralized controllers stabilizing the cart motion process of 3DCrane along the both axis x/y with the different length of the payload lift-line (robust stability). Furthermore, the resulting feedback control system with designed controllers must satisfy robust performance conditions for tracking the desired position of the cart.

2. OVERVIEW OF THE 3DCRANE SYSTEM WITH ARTIFICAL INTERACTION

The 3DCrane system is a nonlinear electromechanical system having a complex dynamic behavior and creating challenging control problem. It is controlled from PC. Therefore it is delivered with hardware and software which can be easily mounted and installed in a laboratory. You obtain the mechanical unit together with the power supply and interface to the PC and the dedicated digital board configured in the Xilinx technology. The software operates under MS Windows using MATLAB and RTW toolbox package.

The 3DCrane setup (see Fig.1) consists of a payload hanging on a pendulum-like lift-line wound by a motor mounted on a cart. The payload is lifted and lowered in the z direction. Both the rail and the cart are capable of horizontal motion in the x direction. The cart is capable of horizontal motion along the rail in the y direction. Therefore the payload attached to the end of the lift-line can move freely in three dimensions.

The 3DCrane is driven by three DC motors. There are five identical measuring encoders measuring five state variables: the cart coordinates on the horizontal plane, the lift-line length, and two deviation angles of the payload.

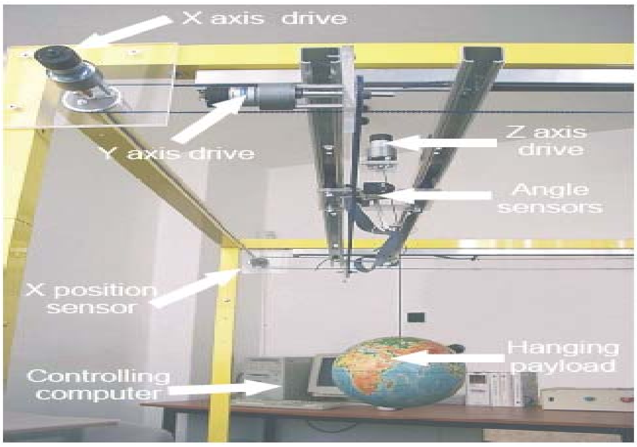


Fig.1. The 3DCrane setup

In the original model, there is no interaction between subsystems, and thus design robust controller for the motion of this 3DCrane system corresponds to design two independent robust controllers for two subsystems. The lift-line R is considered as an uncertainty.

To research the affect of the interaction between subsystems in MIMO system, we consider artificial interactions between the output signals $X(s)$, $Y(s)$ resp. position of card along axes x and axes y. The resulting output signals $X(s)$, $Y(s)$ with interaction corresponding to subsystems are:

$$X(s)=G_{11}(s)\,U_x(s)+\frac{0.25}{8s+1}\,U_y(s)$$
$$Y(s)=\frac{0.3}{7s+1}\,U_x(s)+G_{22}(s)\,U_y(s) \tag{1}$$

The main aim of paper is to design robust PID decentralized controllers for the 3DCrane system with interaction (1) for tracking a desired position of the cart on the both case of movement of crane along axes-x and axes-y.

Procedure to design robust PID decentralized controllers can be summarized as five sequential steps:

1. Choose a suitable control configuration, and then identify motion process of the 3DCrane along axes-x and axes-y at three operating points with lift-line z equals 0[m], 0.25[m], 0.5[m] resp. P_1, P_2, P_3
2. Check the selection of the control configuration for this system (Neitherlinski index (NI)).
3. In the case of succeeding selection of the control configuration, create unstructured model uncertainty for motion process. Otherwise, return the first step.
4. Design a robust decentralized PID controller for this process using Small Gain Theorem algorithm and PDLF.
5. Verify obtained result by simulating in Matlab and on the real model.

3. MAIN RESULT

3.1 Identification of positioning process

Using ARX or ARMAX identification method, result of process identification is described by the following transfer function matrices at three operating points.

Transfer function matrix of system at P1 is:

$$G_1(s)=\begin{pmatrix}\dfrac{-0.04723s+3.363}{s^2+13.64s+0.4507} & \dfrac{0.25}{8s+1}\\ \dfrac{0.3}{7s+1} & \dfrac{-0.01698s+3.032}{s^2+10.52s+0.07569}\end{pmatrix} \tag{2}$$

Transfer function matrix of system at P2 is:

$$G_2(s)=\begin{pmatrix}\dfrac{-0.03037s+3.163}{s^2+13.03s+0.4558} & \dfrac{0.25}{8s+1}\\ \dfrac{0.3}{7s+1} & \dfrac{-0.05296s+5.292}{s^2+18.82s+0.007261}\end{pmatrix} \tag{3}$$

Transfer function matrix of system at P3 is:

$$G_3(s)=\begin{pmatrix}\dfrac{-0.01513s+2.345}{s^2+9.741s+0.3612} & \dfrac{0.25}{8s+1}\\ \dfrac{0.3}{7s+1} & \dfrac{-0.01679s+4.165}{s^2+14.71s+0.08042}\end{pmatrix} \tag{4}$$

3.2 Check the selection of control configuration

In this section, we test the given selection of control configuration with nominal model $G_0(s)$ by using Neitherlinski index (NI.).

The Neitherlinski index (NI) is calculated by equation

$$NI=\frac{\det K}{\prod_{i=1}^{2}k_{ii}}\cong 1>0 \tag{5}$$

where K is stead-state gain matrix of system.

The positive value of Neitherlinski index indicates that, system is structurally stable.

The given selection of control configuration is correct.

3.3 Design of robust decentralized PID controller using method of equivalent subsystems

Consider the MIMO system described by a transfer matrix function $G(s)\in R^{mxm}$ and a decentralized controller $R(s)\in R^{mxm}$. For robust decentralized control procedure we have used the originally developed method, Method of Equivalent subsystems. For the ESM local controllers are designed according to the independent design approach using any frequency –domain design procedure. Resulting local controllers guarantee fulfillment of performance requirements imposed on the full system. Robust stability and performance is guaranteed using Small Gain Theorem. The design procedure of ESM approach the reader can consult at (Kozáková et al. 2009) and (Osuský et al. 2011).

For the following parameters the robust controller has been designed: demanded phase margin of equivalent subsystems is 70 degree (overshoot is about 10 percent), settle time is about 12sec. Using design procedure given at (Kozáková et al. 2009) and (Osuský et al. 2011) for inverse additive type uncertainty

$$G(s)=(I+l_{ia}G_o(s)\Delta(s))^{-1}G_o(s) \tag{6}$$

and robust stability condition

$$\sigma_M(M)<1,\forall\omega \tag{7}$$

where

$$M=l_{ia}(I+G_o(s)R(s))^{-1}G_o(s) \tag{8}$$

We have obtained the following robust controller R_i :

$$R_i = \begin{bmatrix} 2.798 + \dfrac{0.5579}{s} & 0 \\ 0 & 2.653 + \dfrac{0.5073}{s} \end{bmatrix} \qquad (9)$$

Due to (7) the slightly modified robust stability conditions is given at Fig.2

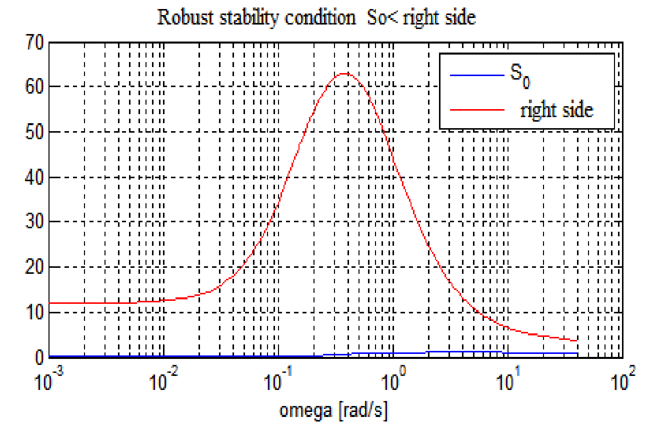


Fig.2.Verifying robust stability condition for input multiplicative model of uncertainty

From Fig.2, we can state that, the closed-loop feedback system with the PI controller R_i is robust stable. And now we verify the obtained result in Matlab simulation (see Fig.3) and on the real model (see Fig.4)

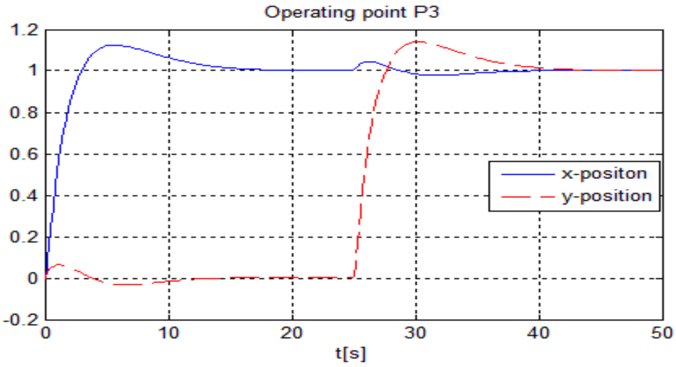


Fig.3. Position output signals with PI controller at third operating point in Matlab

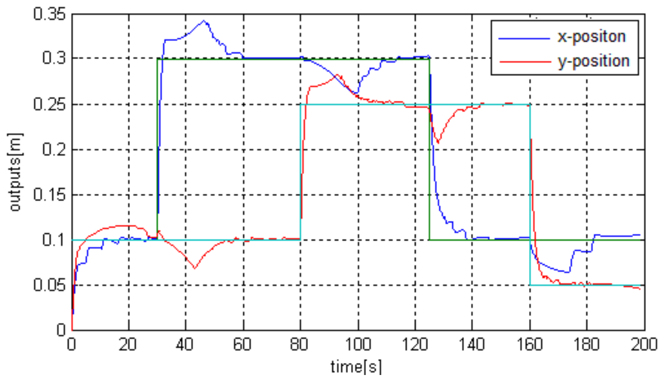


Fig.4 Position output signals with PI controller at third operating point on real model

From results of simulation in Matlab and on the real model we can show that, feedback system with designed PI controllers is robust stable with a demanded performance.

3.4 Design of robust decentralized PID controller using PDLF

Each of the transfer function matrices (2), (3) and (4) can be transformed to linear time invariant continuous system $(A_i, B_i, C, D = 0)$; $i = 1,2,3$, which are considered three vertices of the polytopic system. Our task is to find remain vertex (the fourth vertex) of this polytopic system.

We shall consider the following affine linear time invariant continuous time uncertain system

$$\begin{aligned} \dot{x} &= \left(A + \theta_1 \tilde{A}_1 + \theta_2 \tilde{A}_2\right)x + \left(B + \theta_1 \tilde{B}_1 + \theta_2 \tilde{B}_2\right)u \\ y &= Cx \end{aligned} \qquad (10)$$

where $\underline{\theta}_j \le \theta_j \le \overline{\theta}_j$; $j = 1,2$;

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Polytopic model is defined as follow:

$$\begin{aligned} \dot{x} &= A(\xi)x + +B(\xi)u \\ y &= Cx \end{aligned} \qquad (11)$$

where

$$A(\xi) = \sum_{i=1}^{4} \xi_i A_i \ , \ B(\xi) = \sum_{i=1}^{4} \xi_i B_i \, , \sum_{i=1}^{4} \xi_i = 1 \ ,$$

$\xi_i \ge 0; i = 1 \ldots N = 4$

Vertices of polytopic system are created by the combination of extreme values of θ_j .

$$\begin{aligned} A_i &= A + \theta_1 \tilde{A}_1 + \theta_2 \tilde{A}_2 \\ B_i &= B + \theta_1 \tilde{B}_1 + \theta_2 \tilde{B}_2 \quad , \quad i = 1 \ldots N = 4 \end{aligned} \qquad (12)$$

We suppose that, the extreme values of $\overline{\theta}_j = -\underline{\theta}_j = 1$, $j = 1,2$. Polytopic system will be obtained if for $N! = 24$ combinations of extreme value θ_j , by solving equation system $A_i = A + \theta_1 \tilde{A}_1 + \theta_2 \tilde{A}_2$, $i = 1 \ldots (p+1) = 3$ we have matrices $A, \tilde{A}_1, \tilde{A}_2$ for which the maximal eigenvalues of respective matrix A_4 will be minimal.

The best combination of θ_1, θ_2 for calculation of matrices $A_1, A_2, A_3, A_4 (B_1 \ldots B_4)$ is as follow:

$$\begin{bmatrix} 1 & \underline{\theta}_1 & \underline{\theta}_2 \\ 1 & \overline{\theta}_1 & \overline{\theta}_2 \\ 1 & \overline{\theta}_1 & \underline{\theta}_2 \\ 1 & \underline{\theta}_1 & \overline{\theta}_2 \end{bmatrix} \qquad (13)$$

Consider the uncertain system (15), where

$$B = \begin{bmatrix} 3.2630 & -0.0388 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.8333 & 0 & 4.1620 & -0.0350 \end{bmatrix}^T$$

$$\tilde{B}_1 = \begin{bmatrix} -0.5090 & 0.0161 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5665 & 0.0001 \end{bmatrix}^T$$

$$\tilde{B}_2 = \begin{bmatrix} 0.4090 & -0.0076 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5635 & -0.0181 \end{bmatrix}^T$$

$$A = \begin{bmatrix} 0 & -0.4532 & 0 & 0 & 0 & 0 \\ 1 & -13.335 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3.333 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3.333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0415 \\ 0 & 0 & 0 & 0 & 1 & -14.670 \end{bmatrix}$$

$$\tilde{A}_1 = \begin{bmatrix} 0 & 0.0447 & 0 & 0 & 0 & 0 \\ 0 & 1.9495 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.0024 \\ 0 & 0 & 0 & 0 & 1 & -2.0950 \end{bmatrix}$$

$$\tilde{A}_2 = \begin{bmatrix} 0 & -0.0473 & 0 & 0 & 0 & 0 \\ 0 & -1.6445 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0366 \\ 0 & 0 & 0 & 0 & 1 & -2.0550 \end{bmatrix}$$

Consider PID control law as follow:

$$u = Fy + F_d \frac{dy}{dt} = FCx + F_d C_d \dot{x} \qquad (14)$$

Closed-loop feedback system with PID controller (19) is:

$$M_d(\xi)\dot{x} = A_c(\xi)x \qquad (15)$$

where

$$M_d(\xi) = I - B(\xi)F_d C_d \ , \ A_c(\xi) = A(\xi) + B(\xi)FC$$

Consider cost function as follow

$$J = \int_0^\infty \left(x^T Q x + u^T R u + \dot{x}^T S \dot{x} \right) dt \qquad (16)$$

The closed-loop feedback system (15) with the PID controller (14) is robust stable and guarantees the cost function (16) if and only if there exist matrices $P = \sum_{i=1}^{4} \xi_i P_i \ , P_i > 0;$ $i = 1 \ldots N = 4 , H , G ,$ F and F_d then the following inequality is satisfied (Rosinová et al. 2007)

$$\begin{bmatrix} A_{ci}^T H^T + HA_{ci} + Q + C^T F^T RFC & \left(P_i - M_{di} H + G^T A_{ci} \right)^T \\ P_i - M_{di} H + G^T A_{ci} & -M_{di}^T G - G^T M_{di} + S \end{bmatrix} < 0 \qquad (17)$$

For given cost function with $R = 1 * I$, $Q = S = 0.1 * I$ by using BMI to solve (17), the PID controller is obtained as follow:

$$F = \begin{bmatrix} 1.4065 & 0 & 0.1508 & 0 \\ 0 & 1.1903 & 0 & 0.1218 \end{bmatrix},$$
$$F_d = \begin{bmatrix} 0.1859 & 0 \\ 0 & 0.1579 \end{bmatrix} \qquad (18)$$

And now we verify the obtained result in Matlab simulation (see Fig.5) and on the real model (see Fig.6)

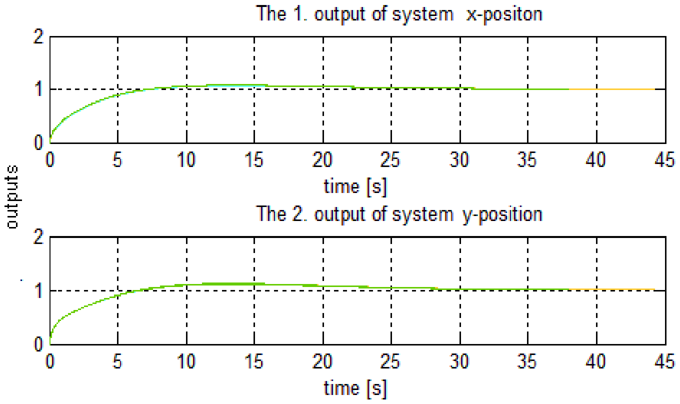


Fig.5. Position output signals with PID controller at three operating points in Matlab

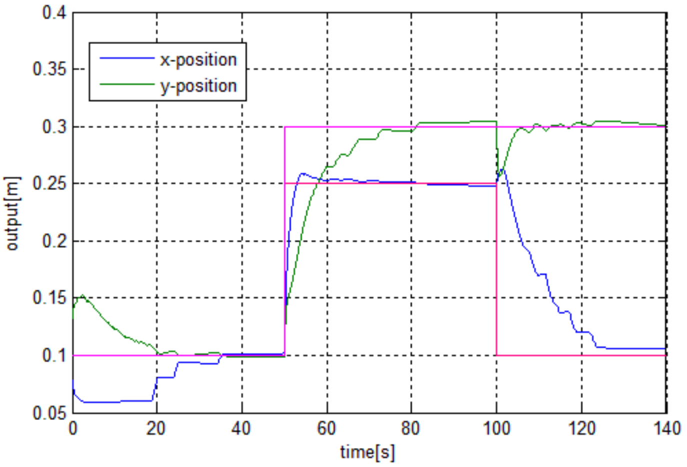


Fig.6. Position output signals with PID controller at third operating point on real model

From simulating results on real model, we can conclude that, the cart of the crane tracks a desired position.

4. CONCLUSION

In this paper, we have researched and applied successfully the knowledge of multivariable system, stabilization of decentralized control systems and the knowledge of robust control theory in the frequency domain and also in the time domain to control the 3DCrane system.

There was a sequential procedure to design robust decentralized controllers for the 3DCrane system. We have identified process of the cart motion along axis x and y. The identification was executed at three operating points corresponding to following length of the payload lift-line: 0, 0.25 and 0.5 [m]. From resulting identified three transfer function matrices, we have designed robust decentralized PID controller to stabilize the cart motion and track the desired position. The resulting PID controllers are verified in the Matlab simulink and on the real model. The cart was at the desired position.

ACKNOWLEDGMENTS

The work has been supported by Grant N1/0544/09 and APVV-0211-10 of the Slovak Scientific Grant Agency.

REFERENCES

INTECO Ltd.: *3DCrane User's manual. RT-CON with USB board version*. [Online]. INTECO Ltd... URL: < http://users.utcluj.ro/~tbuzdugan/App/inteco.pdf>.

Zhining, G., Mahamed, A.: 1997. Stabilization of decentralized control systems. *Journal of Mathematical Systems, Estimation, and Control,* 1997, Vol. 7, No.1, pp. 1-16.

Veselý, V., Kozáková, A., Grman, Ľ.: 2006. Robust control of linear systems in the frequency domain. *Journal of electrical engineering*, 2006, Vol. 57, No. 6, pp. 338-346.

Tham, M. T.: 1999. Multivariable control: An Introduction to decoupling control. Department of Chemical and Process Engineering, University of Newcastle upon Tyne.

Rosinová, D., Veselý, V.: (2007). Robust PID decentralized controller design using LMI. *International Journal of Computers, Communications & Control,* 2007, Vol. II, No. 2, pp. 195-204.

Kozáková A., Veselý V, and Osuský J. :(2009). A new Nyquist /based technique for tuning robust decentralized controllers, *Kybernetika*, Vol 45, N1, 63-83.

Osusky J. and Vesely V. (2011). Robust decentralized controller design with specified phase margin. In. *Process Control*, Vysoke Tatry, submitted.

Comparison of two methods for determining the optical flow

Dipl.-Ing. P. Seibold*

** FernUniversität Hagen, D-58084 Hagen*
(Tel: +49 (2331) 987-1100; e-mail: Peter.Seibold@FernUni-Hagen)

Abstract: An unmanned Arial Vehicle (UAV) outfitted with autonomous control devices shall navigate to predefined positions. By means of cameras and optical flow the position, height above ground, orientation and velocity is determined. Two flow methods, differential technique by Bruce D. Lucas and Takeo Kanade and normalized cross correlation are presented and compared.

1. INTRODUCTION

An unmanned airship shall be regulated in conjunction with other methods by means of image processing in its position, orientation and speed [1][2]. In order to receive 3D information two cameras are mounted under an airship. Every 20ms simultaneously a picture is taken from the left and right camera. The picture of the left camera is compared to the right and to the preceding left picture. The first comparison delivers the height above ground. A measure of the translation and the rotation is obtained by evaluating the relation of the two sequential pictures. With the results the new position and orientation can be determined in respect to the preceding position. With the known starting point the absolute position is cumulated by dead reckoning.

The optical flow [3] [4] determines the relative position of corresponding image dots or regions of two images. The images may have been taken at different times by one camera or simultaneously by slightly shifted positions of two cameras. So far no other light weight system can measure the height above ground in a wider area than cameras.

In contradiction to land based vehicles an airship has six degrees of freedom (x,y,z and roll, yaw, pitch). This aggravates the detection of the pose parameters.

2. OPTICAL FLOW TECHNIQUES

In this paper a differential technique is compared with a region-based matching[4].

2.1 SIMPLIFIED EXPLANATION OF THE DIFFERENTIAL METHOD

For the one-dimensional case (e.g., a picture line) a first approximation for the intensity I_1 around a pixel is given by the linear equation:

$$I = x\left.\frac{dI}{dx}\right|_{x=x1} \tag{1}$$

With the intensity I_2 of a pixel of the second image the displacement Δx can be calculated by:

$$\frac{dI}{dx}\Delta x + \Delta I = 0 \tag{2}$$

more general:

$$\frac{\delta I}{\delta x}dx + \frac{\delta I}{\delta t}dt = 0 \tag{3}$$

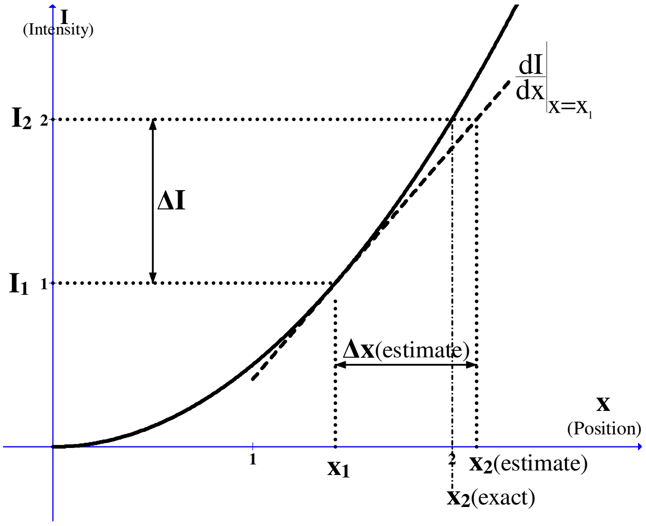


Fig. 1 Differential method for determining optical flow for the one-dimensional case

Fig.1 visualizes eq. (2). Knowing the intensity I_1 at the position x_1 an estimate x_2 for the position of I_2 can be calculated by using the derivate of I(x) at x_1. The exact position of x_2 can be approximated by iteration [5]. The first iteration step is to calculate $\left.\frac{dI}{dx}\right|_{x=x_{2estimate}}$ and to get the x-value of this tangent for I_2. The remaining error is already after this step less than 1 per cent.

The differential method does not accept intensity variations for the corresponding pixels. Also displacements larger than one pixel may cause errors[4]. If larger displacements are to be detected, the images have to be consecutively reduced to smaller sizes and the results must be transposed to enlarged sizes up to the original [6].

For the two dimensional case equation (3) extends to eq. (4) with two unknowns dx and dy.

$$\frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \tag{4}$$

Therefore, other constraints are introduced, such as the assumption of the constancy of the optical flow around a region of the pixel at x, y as proposed in 1981 by Bruce D. Lucas and Takeo Kanade [7].

2.2 EXPLANATION OF A REGION-BASED MATCHING

In contrast to the differential method, which determines the optical flow for each pixel, the region-based matching delivers only the displacement values of a region.

One method is the cross-correlation. By convolution and its maximum, the position of a partial image is detected in the other image. Since the image energy varies in different regions and due to intensity variations the detection may fail. This is avoided by normalizing the values of the overlapping areas. In addition the average intensities are subtracted. We obtain the zero mean normalized cross-correlation (NCC) [8] [9].

$$\gamma_{u,v} = \frac{\sum_{x,y}[f(x,y)-\bar{f}_{u,v}][t(x-u,y-v)-\bar{t}]}{\sqrt{\sum_{x,y}[f(x,y)-\bar{f}_{u,v}]^2}\sqrt{\sum_{x,y}[t(x-u,y-v)-\bar{t}]^2}} \quad (5)$$

$\gamma_{u,v}$ is the correlation coefficient at the location u, v.

Because of the zero mean the range is $-1 <= \gamma_{u,v} <= 1$

f(x,y) is the intensity value at x, y of the image to be searched.

t(x,y) is the intensity value at x, y of the image sought (template).

$\bar{f}$ and $\bar{t}$ are respectively the average intensities in the vicinity of u, v. The area has the size of the template

The mean-free template τ is:

$$\tau(x,y) = t(x,y) - \bar{t} \quad (6)$$

The mean values of $\bar{t}$ and $\bar{f}_{u,v}$ are:

$$\bar{t} = \frac{1}{M_x M_y}\sum_{x=1}^{M_x}\sum_{y=1}^{My} t(x,y) \quad (7)$$

$$\bar{f}(u,v) = \frac{1}{M_x M_y}\sum_{x=u}^{u+M_x-1}\sum_{y=v}^{v+My-1} f(x,y) \quad (8)$$

$M_x M_y$ is the size of the template.

The mean value $\bar{t}$ has to be calculated only once since it is independent of u,v.

Because of the zero mean of τ:

$$\sum_{x,y}\tau(x-u,y-u) = 0 \quad (9)$$

the numerator N of $\gamma_{u,v}$, eq. (5), can be transformed to this:

$$N = \sum_{x,y}[f(x,y)\tau(x-u,y-v)] - \bar{f}_{u,v}\sum_{x,y}\tau(x-u,y-v) \\ = \sum_{x,y} f(x,y)\tau(x-u,y-v) \quad (10)$$

Eq. (10) is a convolution of the image with the reversed template $\tau(-x,-y)$ and can be computed by $F^{-1}\{F(f)F^*(\tau)\}$. F^{-1} is the inverse Fourier transformation. F^* is the complex conjugate.

Each not normalized $\gamma'_{u,v}$, which is the numerator N from eq.(10), has to be divided by the denominator in order to get the normalized coefficients. The equation for the denominator shall be simplified in order to reduce computation effort.

The first term of the denominator is:

$$\sum_{x,y}[f(x,y)-\bar{f}_{u,v}]^2 = \\ \sum_{x,y}[f(x,y)]^2 - \sum_{x,y}2[f(x,y)\bar{f}_{u,v}] + \sum_{x,y}[\bar{f}_{u,v}]^2 \quad (11)$$

$\sum_{x,y}[f(x,y)]^2$ is easily computed with sum tables [11], see example below.

With eq. (8) the second summation is:

$$\sum_{x,y}2[f(x,y)\bar{f}_{u,v}] = 2\bar{f}_{u,v}\sum_{x,y}f(x,y) \\ = \frac{2}{M_x M_y}\left[\sum_{x,y}f(x,y)\right]^2 \quad (12)$$

The second term of the denominator is calculated once:

$$\sqrt{\sum_{x,y}[t(x-u,y-v)-\bar{t}]^2} \quad (13)$$

and is valid for all u,v.

This results into:

$$\gamma_{u,v} = \frac{\sum_{x,y}f(x,y)\tau(x-u,y-v)}{\sqrt{\sum_{x,y}[f(x,y)]^2 - \frac{1}{M_x M_y}\left[\sum_{x,y}f(x,y)\right]^2}} \bullet \frac{1}{\sqrt{\sum_{x,y}[t(x-u,y-v)-\bar{t}]^2}} \quad (14)$$

The sums $\sum_{x,y}[f(x,y)]^2$ and $\sum_{x,y}f(x,y)$ are computed with sum tables [11] and they need only an addition of three values for each u,v. The sum tables must be calculated only once.

Example for a sum table:

The image pixel shall have intensity values as shown in table 1. This results to the table 2 as the sum table. Its origin is at the lower left corner.

1	2	3	4
5	6	7	8
9	1	2	3
4	5	6	7

Table 1: Image intensity values

19	33	51	73
18	30	**45**	63
13	19	27	37
4	9	**15**	22

Table 2: Sum table

To get the sum of the center square of the image, table 1, (6+7+1+2=16), take the value of the upper right edge of this square in the sum table and subtract the sums of adjacent areas on the left and bottom side. The 4 is subtracted twice, therefore it has to be added to the subtractions. The result is:

$$45 - 18 - 15 + 4 = 16 \quad (15)$$

3. COMPARISON OF BOTH METHODS WITH 512X512 PIXEL IMAGES

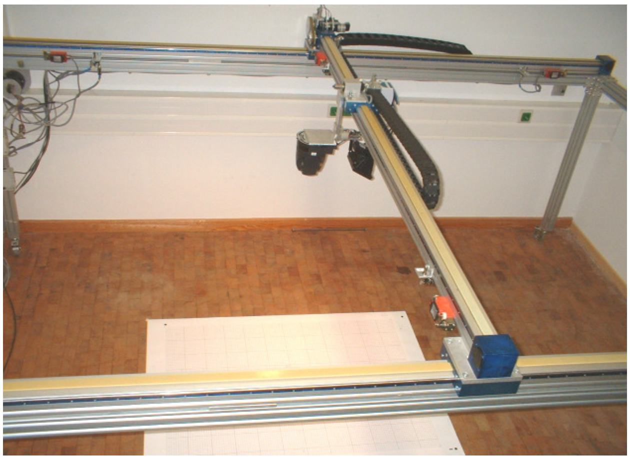


Fig. 2 XY-table with mounted camera

A camera is moved by a XY-table over a scene. The exact position of the camera is given by XY-coordinates which can be set with a precision of better than 0.1mm.

The following tables 3 and 4 show the translation in x and y direction and the angle of the relative rotation of consecutive images as calculated with B. Lucas and T. Kanade method and normalized cross correlation (NCC) in respect to the expected values (Exp.) as determined by the XY-table.

3.1 TRANSLATION

Experiments with real scenes reflect that the differential method of Lucas and Kanade (L&K) detects shifts of up to 30 pixels and rotations of 15 °. NCC can identify any shifts as long as the template is inside the image. However, rotations over 5 ° are detected increasingly uncertain.

	Δx			Δy			$\Delta\varphi$		
	Exp.	L&K	NCC	Exp.	L&K	NCC	Exp.	L&K	NCC
	0,0	-0,1	0,0	33,8	32,9	34,0	0,0	-1,1	0,0
	0,0	-0,2	0,0	33,8	31,5	35,0	0,0	-9,1	0,0
	0,0	0,3	0,0	33,8	32,4	33,0	0,0	-2,9	0,0
	0,0	0,2	0,0	33,8	35,2	35,0	0,0	-0,6	0,0
	0,0	0,0	0,0	33,8	33,2	33,0	0,0	0,0	0,0
	0,0	-0,1	0,0	33,8	34,5	34,0	0,0	0,6	0,0
	0,0	0,1	0,0	33,8	33,6	33,0	0,0	0,0	0,0
	0,0	0,2	0,0	33,8	34,1	34,0	0,0	0,6	0,0
	0,0	0,3	0,0	33,8	33,1	34,0	0,0	1,1	0,0
Cum. Values:	0,0	0,7	0,0	304,2	300,6	304,9	0,0	-11,4	0,0

Table 3: Translation in y-direction

Legend:

Δx, Δy :	displacement in pixel of a region around the center of two consecutive images.
$\Delta\varphi$:	rotation angle of two consecutive images.
Exp.:	expected value as set by the XY-table.
L&K:	differential method of B. Lucas and T. Kanade.
NCC:	Normalized Cross Correlation.
Cum Values:	cumulated values.

Due to the application as pose detection of an airship, in contrast to the often-quoted measure of error (angular error) [10], the deviation of the cumulative value is considered. For translations $(\Delta x, \Delta y)$ both methods deliver good values, NCC somewhat better. At the orientation $\Delta\varphi$, which should be zero, the differential method deviates greatly at one pair of images from the expected value.

3.2 ROTATION

The camera was continuously rotated by 10 degrees around the central axis. As table 4 shows, the values of the differential method are closer to the expected values and is therefore superior to the correlation method.

The advantage of the NCC, is that a failure can be detected with the relationship between a poor value and a low correlation maximum.

	$\Delta\varphi$		
	Exp.	L&K	NCC
	10,0	7,2	6,8
	10,0	8,9	6,8
	10,0	9,9	11,3
	10,0	10,0	11,3
	10,0	10,1	9,1
	10,0	9,9	8,5
	10,0	9,8	8,0
	10,0	9,7	9,1
	10,0	9,8	1,1
Cum. Values:	90,0	85,4	72,0

Table 4: Rotation by 10 degrees

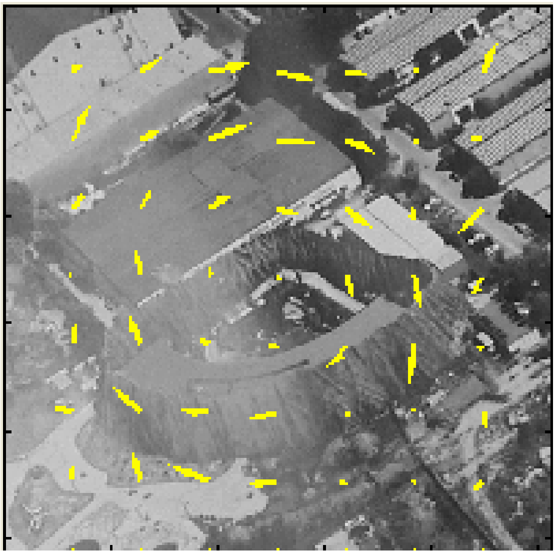


Fig. 3: Flow vectors of a rotated scene. Arrows represent the optical flow of two consecutive images

3.3 BRIGHTNESS CHANGE

Due to brightness change of the scenes the shutter of a camera may alter the exposure. To simulate this effect every 2nd image of a real flight over a landscape has been darkened by a multiplication with 0.95.
As expected, the differential method (Fig. 4) has failed. The composed image calculated using the optical flow is chaotic. The flow vectors as seen in Fig. 5 reflect this irregularity. The composite image by using NCC, Fig. 6, is satisfactory. Without the brightness attenuation the differential method provides the same good result as NCC.

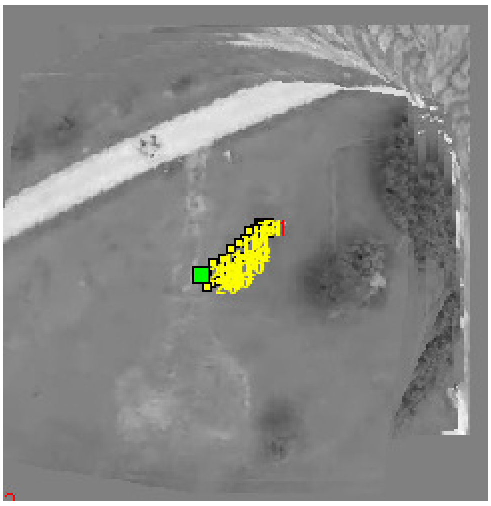


Fig. 4: Composite image, differential method with brightness change

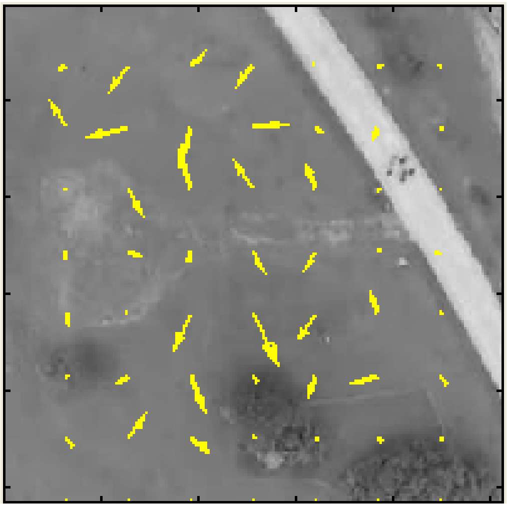


Fig. 5: Chaotic flow vectors of one pair of images with the differential method. Brightness is changed.

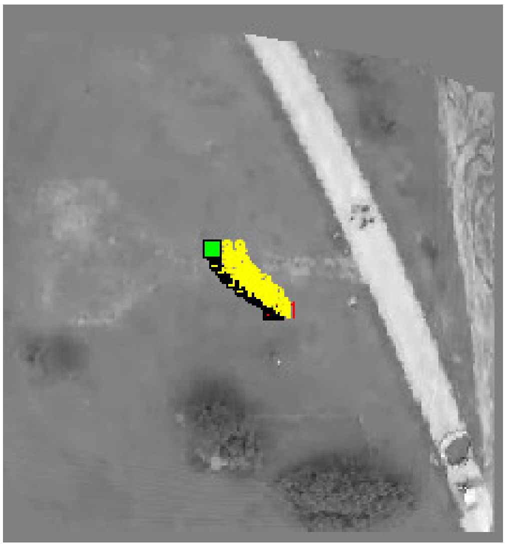


Fig. 6: NCC
Composite image with brightness change

3.4 COMPUTING EFFORT

Relevant to the applicability in real time is the calculation time. As a first estimate the calculation time on a PC is used, even when operating in a DSP or FPGA, memory access times are more important than multiplications and additions. A PC reduces memory access times by a large "pre-cache". The differential method requires on a PC 460ms for the calculation of an image pair, while NCC with 170ms is almost three times faster. In order to control the airship autonomously the target process time for two consecutive images shall be less than 20 ms. This task will be accomplished by using DSP for fast sequential arithmetic operation and FPGA for parallel processing.
The amount of multiplications in respect to the processing time are not any more that important since DSPs accomplishes them in one cycle. Other operations like divisions, square roots take many more cycles and should be avoided.

Memory usage in respect to image size M_x (columns), M_y(rows) and a required precision P_i.:
The precisions may be set as used for the calculations of tables 3 and 4 to:
P_1: one byte
P_2: 2 bytes (half precision)
P_4: 4 bytes (single precision)

Differential method:
For first, 2nd image and pyramid: 2.6 $M_xM_yP_1$
Flow vectors x and y: 2 $M_xM_yP_2$
Sum for 512x512 image: 1.2MByte
NCC:
For first and 2nd image: 2 $M_xM_yP_1$
FFT (complex) for both images and inverse FFT (real): 5 $M_xM_yP_4$
Sum tables: 2 $M_xM_yP_4$
Sum for 512x512 image: 7.9MByte

The real amount in bytes is depending on the required accuracy. Even when some memory may be reused, NCC needs by far more memory resources.

4. SUMMARY

NCC is superior to the differential method in the detection reliability, accuracy and speed. Also the allowed displacement range is much wider. While NCC needs only the corresponding region to be within both consecutive images, the differential method starts to fail at more than 30 pixel distance. However the differential method is superior at rotations and for the density of the flow vectors. Since for the application in an airship the maximal rotation angle of two consecutive images is expected to be less then 5 degrees and a density of the optical flow down to one pixel is not necessary, NCC is the preferred method.

Further test flights will be done in order to check the maximum dynamics of movements and the requirement for the optical flow. Also ideas to detect roll and pitch will be developed.

REFERENCES

[1] Chris Mccarthy and Nick Barnes, Performance of Optical Flow Techniques for Indoor Navigation with a Mobile Robot, Proceedings of the IEEE International Conference on Robotics & Automation, 2004

[2] Franck Ruffier and Nicolas Franceschini, Optic flow regulation: the key to aircraft automatic guidance, Robotics and Autonomous Systems, 2005

[3] B. K. P. Horn and B. G. Schunck, Determining optical flow, Artificial Intelligence, 1981

[4] J. L. Barron et al., Performance of optical flow techniques, International Journal of Computer Vision, 1994

[5] Andrés et al., Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods, 2005

[6] Peter J. Burt and Edward H. Adelson, The Laplacian Pyramid as a Compact Image Code, IEEE Transactions on Communications, 1983

[7] Bruce D. Lucas and Takeo Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision, Proceedings of the 7th International Joint Conference on Artificial Intelligence, 1981

[8] J. P. Lewis, Fast Normalized Cross-Correlation, Industrial Light & Magic, 1995

[9] A. J. H. Hiiet et al., Fast normalized cross correlation for motion tracking using basis functions, Comput. Methods Prog. Biomed.,2006

[10] Fleet, D.J. and Jepson, A.D., Computation of Component Image Velocity from Local Phase Information, IJCV, 1990

[11] Crow, Franklin C., Summed-area tables for texture mapping, SIGGRAPH Comput. Graph., 1984

Model Predictive Control for industrial applications

Georgios Papafotiou

ABB Corporate Research, Segelhof, 5405 Dättwil, Switzerland

Model Predictive Control represents an exciting academic research field and at the same time a well established and mature technology in many industrial applications, where physical processes need to be controlled in an efficient and reliable way. Until recently, however, its appeal has been mainly restricted to processes with rather slow dynamics with sampling times ranging from a few minutes to many hours, such as the ones encountered in the areas of (petro)chemicals, minerals and metals. The main reason for this restriction can be traced to the computational demand that optimization-based algorithms can pose to the control hardware platforms, since in its most common version MPC requires the online solution of a constrained optimization problem at each time step. Nowadays, however, the increased computational capacity that is becoming available in the commonly employed controllers, coupled with recent algorithmic advances, has encouraged the emergence of MPC applications in the automotive, and more recently the power electronics industry, where the time scales are in the range of milli- or even micro-seconds. This talk tries to emphasize this visible trend in industrial reality and present recent developments in the application of MPC for the efficient control of Medium Voltage induction motor drives.

Application of Quantitative Feedback Theory for Wind Turbine Controller Design

Goran Benčić*, Mate Jelavić, Nedjeljko Perić*****

* *KONČAR – Power Plant and Electric Traction Engineering, Zagreb, Croatia*
(goran.bencic@koncar-ket.hr)

** *KONČAR – Electrical Engineering Institute, Zagreb, Croatia*
(mjelavic@koncar-institut.hr)

*** *Faculty of Electrical Engineering and Computing, Zagreb, Croatia*
(nedjeljko.peric@fer.hr)

Abstract: *To enable wind turbines to produce power under great variety of wind conditions a sophisticated control system is needed. Wind turbine system is highly nonlinear and its dynamics changes rapidly with the change of wind speed. Many classical control methods fail to properly address this uncertainty of wind turbine dynamics. For that reason Quantitative Feedback Theory is presented and its application to synthesis of rotor speed controller.*

1. INTRODUCTION

Modern wind turbines have to operate in wide range of operating conditions determined primarily by wind speed. To make it possible for wind turbine to produce power in such a variety of operating conditions a sophisticated control system is needed that will account for changes in operating conditions and accompanying changes in wind turbine dynamics [1]. The power of air that moves at speed v_w over the area swept by turbine rotor of radius R is given by (1):

$$P_W = \frac{1}{2}\rho_{air}R^2\pi v_w^3 \tag{1}$$

where ρ_{air} is density of air. From expression (1) it is clear that wind energy increases rapidly with increase in wind speed. This results in two very different operation regions of wind turbine, each of them placing specific demands upon control system. During weak winds power contained in the wind is lower than the rated power output of wind turbine generator. Therefore, the main task of the control system in this region is to maximize wind turbine power output by maximizing wind energy capture. It can be shown [2] that for each value of wind speed energy conversion efficiency is maximal for only one particular value of rotor speed. Since modern wind turbines are connected to grid using AC-DC-AC frequency converters, generator frequency is decoupled from grid frequency which enables variable speed operation. Therefore it becomes possible to vary the rotor speed and to maintain optimal energy conversion during varying wind speeds. On the other hand, during strong winds power of the wind is greater than the rated power output of wind turbine generator. Therefore, the wind energy conversion has to be constrained in this region to assure generator operation without overloading. Very efficient method for constraining wind energy conversion is pitching the rotor blades around their longitudinal axis which deteriorates their aerodynamic efficiency and therefore only a part of wind energy is used for driving the generator.

The main task of wind turbine control system is to obtain continuous power production under operating conditions determined by various wind speeds. As turbine power is directly proportional to its speed, power control can be done by controlling turbine speed. The principle scheme of wind turbine speed control system is shown in Fig. 1.

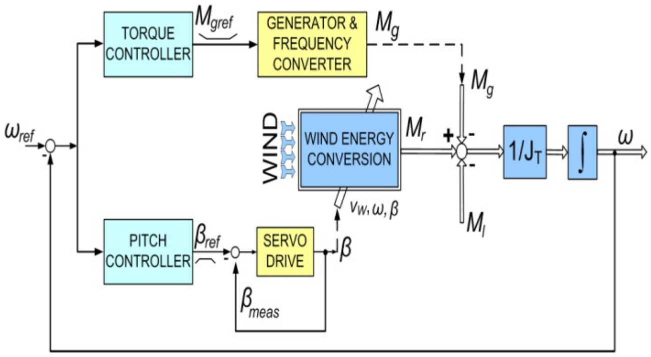


Fig. 1. Principle scheme of wind turbine control system [3]

As it can be seen in this figure turbine speed can be influenced and thus controlled by two means – by generator electromagnetic torque M_g which opposes rotor driving torque M_r and by pitch angle β which alters the wind energy conversion. For this reason turbine speed control system consists of two control loops: torque control loop and pitch control loop. Those control loops operate simultaneously but depending on operation region one of them is dominant. In the below rated operation region the torque control loop is used to control turbine speed to values that will result in maximal wind power capture. This control loop is not in the scope of this paper. Details on its specifics can be found in e.g. [3]. In the above rated region this control loop just holds generator torque at its rated value. The pitch control loop is used for setting the adequate pitch angle that will keep turbine speed at its reference value under all operating conditions determined by various winds. Below rated wind speed this loop sets pitch angle to value that assures maximal wind power capture which is usually around 0°. In this paper we assume that all blades have the same pitch angle what is known as collective pitch. Controller in this loop, although used to control turbine speed, is commonly termed pitch controller. Blade positioning is mostly done using electrical servo drives that rotate blades by means of gearboxes and slewing rings. Position control of servo drives is usually achieved using frequency converters. This control loop design is rather simple and is not in the scope of the paper.

2. PROBLEMS OF CLASSICAL CONTROL SYSTEMS

The main problem for most of classical control methods is handling of nonlinear dynamical systems. Even simple models of wind turbines are highly nonlinear due to nature of aerodynamic conversion that takes place on all rotor blades. These models usually don't take into consideration aeroelasticity of the blades, wake effects, yaw errors, stall effect, tower shadow, wind shear effects etc. and still present a tough challenge for most of classical methods. The core of the problem mostly lies in inability of methods to explicitly account for uncertainty of process dynamical behavior that arises from changes in working conditions (higher wind speed, lower wind speed). Furthermore, when a controller is parametrized, there are usually no guarantees of stability and quality of disturbance rejection when operating point changes. For that reason it is necessary to perform extensive time simulations to a posteriori determine if initial specifications for stability and disturbance rejections are satisfied in all cases. QFT on the other hand rises up to this challenge as it can a priori process uncertainty, quantify it and used it in combination with closed loop specifications. It can also a priori guarantee fulfillment of closed loop specification.

3. MOTIVATION FOR UTILIZATION OF QFT

In the beginning of 1960s Horowitz introduced a new frequency domain based control method called Quantitative Feedback Theory (abbr. QFT) which presented a generalization of Bode's frequency domain work [4]. During Horowitz' involvement in the development of control system for Israeli battle aircrafts, QFT method was completed and received a form in which it is used today [4]. Successful utilization of QFT in aircraft control has proved the power of the method and enabled its application in helicopter control systems. When one takes into consideration that much of wind turbine aerodynamical modelling stems directly from helicopter aerodynamical modelling, it is only logical to conclude that QFT should handle in a satisfying manner control of rotor speed above rated wind speed. The main characteristic of QFT is the ability to explicitly take into account uncertainty of process that is to be controlled, and use this knowledge to develop a controller able to meet certain specifications (i.e. for efficient disturbance rejection, noise reduction, etc.). Due to high transparency of the method it is possible to surveil almost every aspect of the problem in hand and thus make needed trade-offs between quality of disturbance rejection, amount of stability margins, controller complexity and bandwidth utilization. This feature is especially appealing as it enables engineers to synthesize efficient low-bandwidth linear controllers of low order. Utilization of low-bandwidth controllers decreases system's sensitivity to noise and unmodelled dynamics. QFT is a completely rounded control method as it is applicable to various control systems: linear, nonlinear, time (non)varying, continuous and discrete, (non)minimum phased, Multiple Input Single Output (abbr. MISO), Multiple Input Multiple Output (abbr. MIMO), with output and state signals feedback, time-delayed (variant of QFT Smith predictor was developed for this purpose) [5]. It is even applicable to certain class of uncertain distributed systems whose behavior is described with partial differential equations (i.e. control of large scale manipulators) [5].

4. MISO QFT

The basis of all QFT methods (all variants of MIMO QFT, discrete QFT, QFT for non-minimum phased systems) is comprised in 2-degree of freedom structure called MISO QFT [5] shown in Fig. 2.

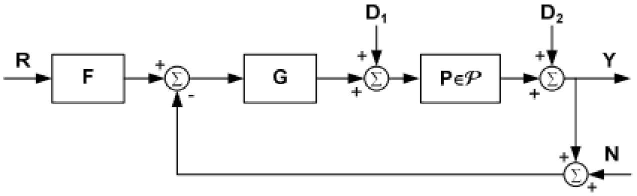


Fig. 2. MISO QFT control structure

The elements in Fig. 2 are described below:

$\mathcal{P}$ – a set of transfer functions where $P_i(j\omega) \in \mathcal{P}$ describing the area of process parametric uncertainty.

G – QFT controller intended to make this feedback system robust, reject disturbances and reduce sensitivity to noise.

F – prefilter that enables quality tracking of reference signal R.

Signals in Fig. 2 are: measurement noise N, disturbance D_1 on process P input, disturbance D_2 on process P output, reference signal R.

The process of obtaining an adequate QFT controller $G(s)$ and prefilter $F(s)$ can be described through following steps:

1) Determine the set of transfer functions $\mathcal{P} = \{P_i(j\omega_i)\}$ that describe the whole range of process dynamical behavior.

2) Choose a nominal process $P_0(j\omega)$ from the given set $\mathcal{P}$ (any one will do).

3) Choose discrete frequency set $\Omega = \{\omega_1, \omega_2, \dots, \omega_M\}$ from frequency range relevant for control. Further controller synthesis is performed on this discrete set Ω.

4) Generate templates (sets that describe area of phase vs amplitude variations) for every frequency from Ω. In other words, if phase and amplitude values are calculated for every $P_i \in \mathcal{P}$ for certain frequency $\omega_k \in \Omega$, then this set of values is called the template $\Pi(j\omega_k)$.

5) Determine a set of specification for closed loop system behavior (i.e. allowable upper and lower boundary for tracking of R, upper boundary for disturbance rejection, stability, control effort, etc.) and translate them to frequency domain.

6) Using Nichols chart, given specifications and templates, find frequency boundaries $\mathcal{B}_i$ on Nichols chart. For every specification there is a set of boundaries $\mathcal{B}_i$ generated on Nichols chart. This set is calculated only for frequencies from Ω. For example, $\mathcal{B}_i(j\omega_k)$ would present a boundary for i-th specification evaluated on ω_k. Crucial detail of this algorithm is that all of these boundaries are calculated in dependence of before mentioned nominal process $P_0(j\omega)$.

7) Draw the nominal open loop $L_0(j\omega) = P_0(j\omega)\, G(j\omega)$ on the same Nichols chart and commence with classical loop shaping procedures in order to satisfy calculated boundaries.

8) Draw the whole set of closed loop transfer functions on Bode diagram and find suitable prefilter F to satisfy servo specifications (if such exist) for tracking of reference R signal.

9) Perform frequency and time validation of control design. Iterate if necessary.

Step 6) is crucial for QFT method and will be explained in a graphical manner which could offer the reader a better insight. For example, a stability margin specification is given as (2):

$$\left|\frac{L(j\omega)}{1+L(j\omega)}\right| \le M_m \tag{2}$$

This relation is represented as exterior of a red closed curve around the critical point (-180°, 0 dB) on Nichols chart on Fig. 3.

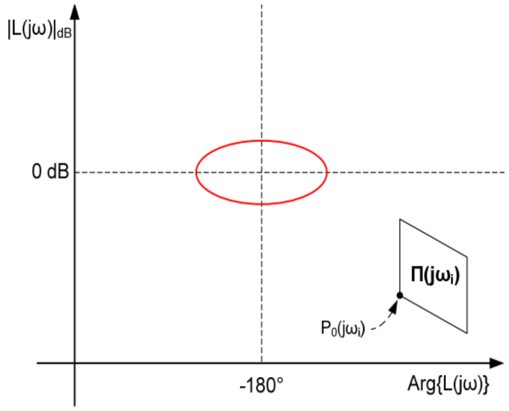


Fig. 3. Closed curve around critical point and the template $\Pi(j\omega_i)$

$\Pi(j\omega_i)$ represents the process template and $P_0(j\omega_i)$ represents the nominal process. The template needs to remain outside the region enclosed by the red curve. Firstly the template needs to be moved maximally close to the red curve (none of the points belonging to the template are allowed to enter the enclosed region) and the position of the nominal process need to be marked for every position of the template. Such movement of the template in magnitude-phase plane (Nichols chart) is possible if controller is connected as it enables adjustment of phase and magnitude i.e. translation Connect these markings of the nominal process (green line in Fig.4.). This green line actually represents the stability boundary $\mathcal{B}_s(j\omega_i)$ on frequency ω_i. If during step 7) the value of open loop transfer function $L_0(j\omega_i)$ remains outside the $\mathcal{B}_s(j\omega_i)$, then there is a guarantee that none of all possible closed loop systems values on ω_i will be within forbidden enclosed area. Similar graphical logic applies to other types of specifications. QFT software tools handle boundary generations by extracting template boundary (thus reducing the computation burden due to many insignificant interior points) and then solving systems of quadratic inequalities. The controller synthesis is finally performed on a set of resultant boundaries that represent the intersection of all boundaries.

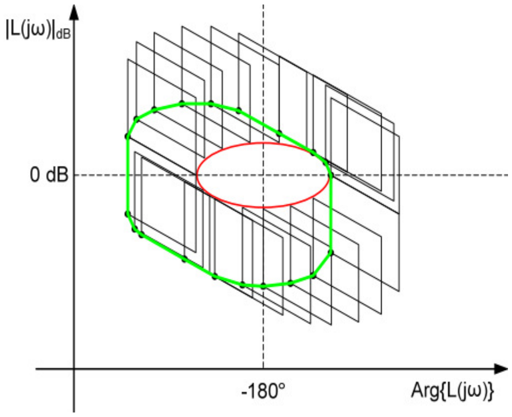


Fig. 4. Template moved around the curve forming a stability boundary $\mathcal{B}_s(j\omega_i)$

5. WIND TURBINE MODELLING

The first step in control system design is to obtain a suitable process model to describe dynamical behavior of wind turbine. Combination of blade element and momentum theory yields quite satisfactory description of wind turbine aerodynamic effects which are at center of scope in modelling. It is primarily utilized in simulation tools but lacks simplicity in order to be suitably used in the process of controller design due to iterative nature of the method. For this reason a different, more analytical, approach is used that develops a simplified mathematical model usual in the literature dealing with controller design. It will be described briefly, while the details on it can be found in [6] and [7].

Wind power P_w, given by expression (1), can never be completely transformed into wind turbine power P_{wt} and afterwards into electrical power P_{el}. The amount of wind power that is converted into turbine power P_{wt} can be described by expression (3):

$$P_{wt} = P_w C_p \tag{3}$$

where C_p represents a performance coefficient.

The theoretical maximum for C_p is determined by the Betz' law [2] and equals $16/27 \approx 0.59$. The interesting part about assumptions made in deriving Betz' law is that no particular turbine design was considered and no additional losses were included (wake losses, friction losses, etc.) which means that 16/27 is an absolute limit for power extraction process. Modern wind turbines reach at best performance coefficient of 0.5. The value of C_P varies in dependence on wind speed v_w, rotor speed ω and blade pitch angle β. Wind speed and rotor tip speed are usually bound together introducing parameter λ that is called tip speed ratio [2] given by expression:

$$\lambda = \frac{\omega R}{v_w} \tag{4}$$

Typical dependence of performance coefficient upon tip speed ratio with pitch angle used as a parameter is shown in Fig. 5.

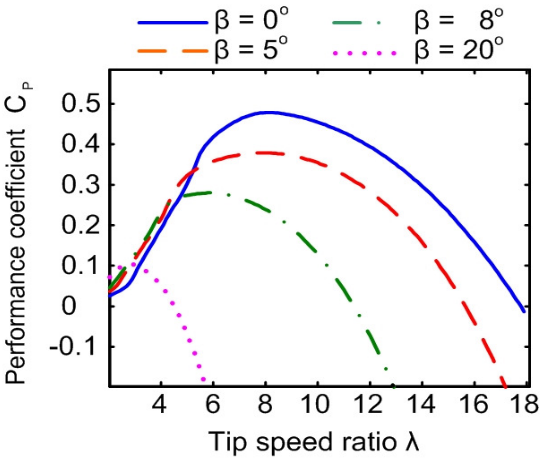


Fig. 5. Performance coefficient as a function of tip speed ratio [1]

Aerodynamic torque that drives wind turbine rotor is given by (5):

$$M_r = \frac{P_r}{\omega} = \frac{1}{2}\frac{\rho_{air} R^2 \pi v_w^3 C_p(\lambda,\beta)}{\omega} \tag{5}$$

Using relation defined by (4) a rearrangement of expression (5) is obtained as follows (6):

$$M_r = \frac{1}{2}\frac{\rho_{air} R^3 \pi v_w^2 C_p(\lambda,\beta)}{\lambda} \tag{6}$$

A quotient of performance coefficient C_p and tip speed ratio λ forms a new dimensionless parameter known as torque coefficient C_Q [2]:

$$C_Q(\lambda,\beta) = \frac{C_p(\lambda,\beta)}{\lambda} \tag{7}$$

Now the rotor speed ω can easily be found using principle equation of motion given by :

$$J_t \frac{d\omega}{dt} = M_r - M_g - M_{loss} \tag{8}$$

where M_g is generator electromagnetic torque, J_t is total moment of inertia of generator rotor and wind turbine, while M_{loss} is loss torque caused by friction losses (usually neglectable).

Wind turbine considered in this paper is In this paper we consider wind turbine with generator that is directly coupled with turbine rotor. This turbine setting known as direct drive system uses synchronous multipole generator that rotates at small speed of turbine rotor. Since rotor and generator speeds are the same no distinction between them is made throughout the paper. Because there is no gearbox between rotor and generator their moments of inertia can just be summed together in order to calculate total moment of inertia J_t. The coupling of rotor to the generator in direct drive solutions is very stiff and it can be considered as rigid thus removing any torsional oscillations what simplifies the control system design.

Before going further an important issue has to be addressed. Namely, expressions (5), (6) and (7) in this form would be valid only for structure with rigid tower and blades. In real situation the absolute wind speed v_w in mentioned expressions has to be replaced by wind speed that is "seen" by rotor blades. This wind speed seen by the rotor is the resultant of three factors: absolute wind speed v_w, speed of the tower movement perpendicular to wind speed (i.e. tower nodding speed) $\dot{x}_t$ and speed of blade movement perpendicular to wind speed (i.e. speed of blade flapwise movement). Influence of tower nodding on wind turbine control is much more pronounced than influence of blade flapwise movement. Therefore we focus only on tower nodding considering rotor blades as rigid. This results in a following expression describing the wind "seen" by rotor blades:

$$v_w^* = v_w - \dot{x}_t \tag{9}$$

Tower nodding originates from the fact that wind turbine tower is very lightly damped structure due to its great height (more than 100 meters in modern wind turbines) and need for moderate mass. To model the wind turbine tower precisely we would have to use model with distributed parameters and to describe it in terms of mass and stiffness distribution. Such a model wouldn't be very suitable for controller design so it has to be substituted by model with concentrated parameters. This can be done using modal analysis that is very common tool in wind turbine analysis [1], [3]. It describes a complex oscillatory structure as a composition of several simple oscillatory systems each of them being described by means of mass, stiffness and damping. By this representation complex tower oscillations are seen as a sum of many simple oscillations characterized by their modal frequencies which are one of the most important structural properties of wind turbine. It has been shown in practice [5] that fairly good modeling of wind turbine tower nodding can be achieved using two modal frequencies (two modes). Since we are here primarily interested in building model suitable for controller design we use only the first modal frequency. The justification for this lies in the fact that for the turbine in scope second modal frequency is more than 6 times greater than the first modal frequency and therefore falls out of the controller frequency bandwidth.

By using only one modal frequency tower dynamics can be described as:

$$M\ddot{x}_t + Dx_t + C\dot{x_t} = F(t) \tag{10}$$

where M, D, and C are modal mass damping and stiffness respectively and $F(t)$ is the generalized force that is originated by wind and that causes wind turbine tower oscillations. Tower modal properties in expression (10) are related to first tower modal frequency ω_{0t} as follows [6]:

$$\begin{aligned} D &= 2\zeta_t \omega_{0t} M \\ C &= (\omega_{0t})^2 M \end{aligned} \tag{11}$$

where ζ_t is structural damping. For steel structure structural damping is mostly set to 0.005 [6]. Modal mass M can be calculated as [2]:

$$M = \int_0^{h_t} m(h)\Phi(h)^2 dh \tag{12}$$

where h_t is the height of the tower, $m(h)$ is the mass distribution along the tower height and $\Phi(h)$ is the tower's first mode shape. Note that actual distribution of mass along the tower has to be modified in order to include mass of the rotor and the nacelle which is assumed to be concentrated at the tower top.

Driving force F is mostly the rotor thrust force F_t caused by the wind. It can be shown [6] that thrust force, similar to aerodynamic torque, depends on wind speed, rotor speed and pitch angle. So similarly to (6) it can be expressed as [6]:

$$F_t = \frac{1}{2}\rho_{air} R^2 \pi v_w^2 C_t(\lambda, \beta) \tag{13}$$

where C_t is the, so called, thrust coefficient.

Expressions (6), (8), (10) and (13) form the simplified nonlinear model of wind turbine that is used in the following sections for controller design. Model is summarized below taking into account the fact that wind speed seen by the rotor is a sum of wind speed and tower nodding speed:

$$\begin{aligned} J_t \frac{d\omega}{dt} &= M_r - M_g - M_{loss} \\ M_r &= \frac{1}{2}\rho_{air} R^3 \pi (v_w - \dot{x}_t)^2 C_Q(\lambda, \beta) \\ F_t &= \frac{1}{2}\rho_{air} R^2 \pi (v_w - \dot{x}_t)^2 C_t(\lambda, \beta) \\ M\ddot{x}_t &+ Dx_t + C\dot{x_t} = F(t) \end{aligned} \tag{14}$$

Torque and thrust coefficients C_q and C_t are usually provided by wind turbine blade manufacturers or can be calculated using professional simulation tools.

On Fig. 6. a block diagram depicts the simulation model of the wind turbine used to obtain results that follow. The central part of it is the aerodynamical model shown on Fig. 7. that implements equations (6) and (13) defining M_r and F_t. C_Q and C_t are represented by 2D look-up tables with pitch angle β and tip speed ratio λ as their input signals. Torque M_r and thrust force F_t represent resulting output signals. The control system of the pitch drive will not be addressed in this paper and can be approximated, for small reference pitch changes, in a satisfactory manner by 2nd order aperiodical system. In reality, pitch drive would use cascaded position and speed control loops that would have to overcome aerodynamic torque developed around the longitudinal blade axes, stiction and friction induced torques inside the blade bearings.

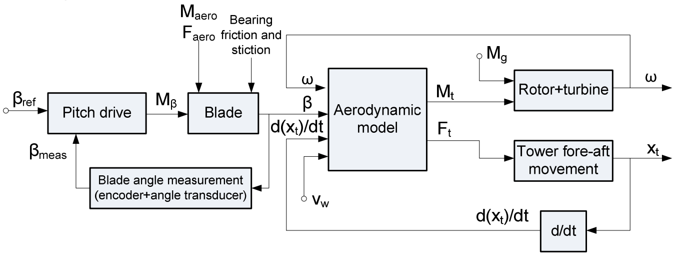


Fig. 6. Block diagram of the wind turbine simulation model

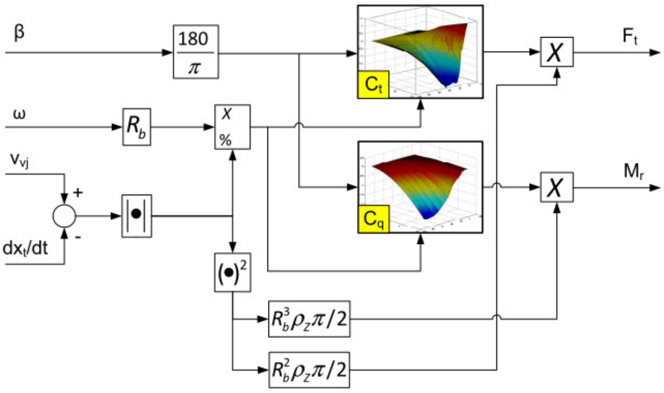


Fig. 7 Block diagram of aerodynamical model

In order to use QFT method for speed controller synthesis a linear model is required. From system of equations given by (14), second and third equation need to be linearized as shown below:

$$\Delta M_r = \left.\frac{\partial M_r}{\partial v_w}\right|_{O.P.} (\Delta v_w - \Delta\dot{x}_t) + \left.\frac{\partial M_r}{\partial \beta}\right|_{O.P.} \Delta\beta + \left.\frac{\partial M_r}{\partial \omega}\right|_{O.P.} \Delta\omega \tag{15}$$

$$\Delta F_t = \left.\frac{\partial F_t}{\partial v_w}\right|_{O.P.} (\Delta v_w - \Delta\dot{x}_t) + \left.\frac{\partial F_t}{\partial \beta}\right|_{O.P.} \Delta\beta + \left.\frac{\partial F_t}{\partial \omega}\right|_{O.P.} \Delta\omega \tag{16}$$

Term O.P. used in equations above is an abbreviation for operating point. An automated procedure was used in order to obtain partial derivatives in (15) and (16). Perturbation was introduced to stationary values of input signals and it was observed on output points in which extent these perturbations were amplified. Discrete wind range $\mathcal{V}_w$ was used to define operating points above rated wind speed (approximately 12 m/s):

$$\mathcal{V}_w = \{12,13,14,\ldots,24,25\}\ [m/s] \tag{17}$$

In this regime of operation it can be considered that constant nominal generator torque M_g is used, so no ΔMg is introduced into the system. Therefore this dynamics will be neglected. By combining linearized equations (15) and (16) with (8) and (10) an expression can be obtained that brings together into a classical relation pitch angle β (plant input) and wind speed v_w (disturbance input) with rotor speed ω (plant output):

$$\omega(s) = G(s)\beta(s) + G_d(s)v_w(s) \tag{18}$$

For every wind speed $v_{w,i} \in \mathcal{V}_w$ accompanying pair of transfer functions $G_i(s) \in \mathcal{G}$ and $G_{d,i}(s) \in \mathcal{G}_d$ is obtained.

Families $\mathcal{G}$ and $\mathcal{G}_d$ of transfer functions are shown on Fig. 8. and Fig. 9.

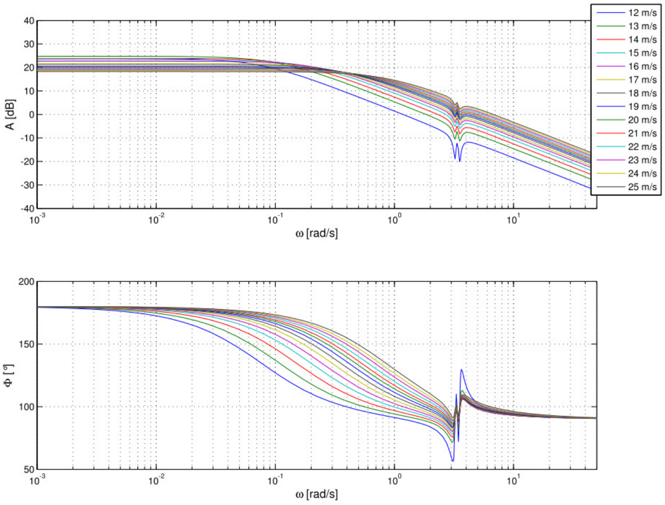


Fig. 8. Bode plot of transfer function family $\mathcal{G}$

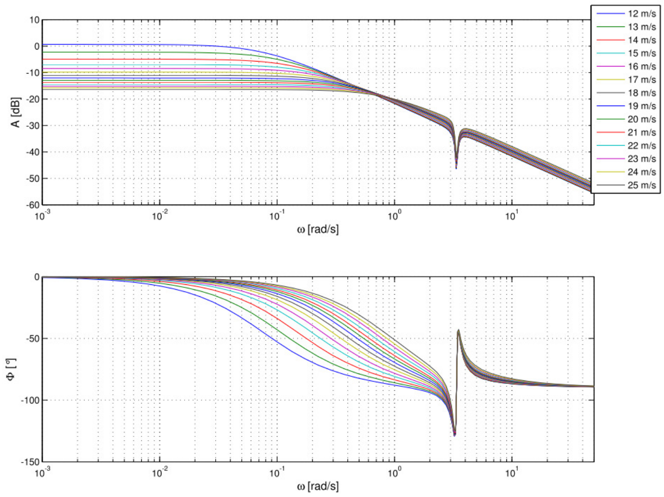


Fig. 9. Bode plot of transfer function family $\mathcal{G}_d$

Observe a phase shift on lower (relevant) frequency range by 180° on Fig. 8. that relates rotor speed change to pitch angle change. This means that a small rise in pitch angle produces negative change, due to phase shift, in rotor speed i.e. slowing down of the rotor. Physically this causes a decrease in angle of attack and consequently lowering of tangential forces on blade sections is caused. This in turn cumulatively decreases the value of driving torque M_r when contributions of all blades and all blade sections are summed.

6. DESIGN SPECIFICATIONS

Two types of specifications are defined and later on accompanying open loop boundaries on Nichols chart are calculated in order to facilitate controller design process. First appropriate discrete set of design frequencies needs to be specified. Regarding this problem there are no strict rules, instead some useful guidelines exist. Generally it is useful to choose frequencies that give results with meaningful differences in calculated boundaries. This can be computationally bothersome as it requires iterations. As a rule of thumb frequencies separated by an octave inside meaningful frequency range should suffice. Special care should be taken if process dynamics exhibits resonances at certain frequencies as then few frequencies around the resonance frequency should be chosen to appropriately describe abrupt phase and magnitude changes. Below is given the set Ω of frequencies that were used in calculations:

$$\Omega = \{10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1, 2.5, 3.3, 3.38, 3.47, 5, 20\} \qquad (19)$$

Due to natural frequency of tower first mode at $\omega_8 = 3.38\, rad/s$, nearby frequencies $\omega_7 = 3.3\, rad/s$ and $\omega_9 = 3.47\, rad/s$ were chosen. All the frequencies above $20\, rad/s$ are represented by $\omega_{11} = 20\, rad/s$ as their templates degenerate into virtually same shape that is solely dominated by variations of process magnitude. This can be stated as follows:

$$G(j\omega) \approx \frac{K}{(j\omega)^{\lambda}} \text{ when } \omega \gg \omega_b \qquad (20)$$

where λ represents pole excess, ω_b represents bandwidth frequency and $K \in [K_{min}, K_{max}]$ represents amplitude variation. This fact is depicted on Fig. 10. by the template on $20\, rad/s$ that shows very small variations in phase values and dominant variations in amplitude values.

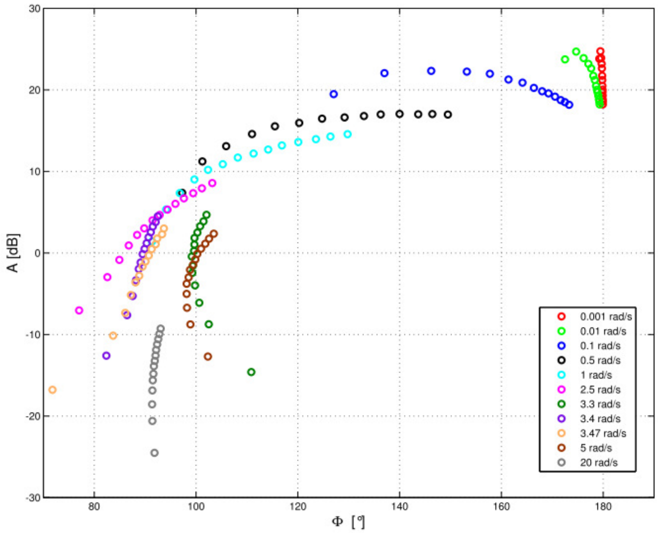


Fig. 10. Process templates representing variations of amplitude vs phase

First type of specifications refers to defining of stability margin factors. Instead of amplitude and phase margin, another relation was used that wraps them together stating (see expression (2)):

$$\left|\frac{C(j\omega)G_i(j\omega)}{1 + C(j\omega)G_i(j\omega)}\right| \leq M_m \, for \; \forall G_i \in \mathcal{G} \qquad (21)$$

where $C(j\omega)$ represents controller transfer function. Amplitude margin (A.M.) and phase margin (P.M.) are bound together to M_m as follows:

$$\begin{aligned} A.M. &= -20 \log\left(\frac{M_m}{1 + M_m}\right) \\ P.M. &= 180° - \text{acos}\left(\frac{1}{2M_m^2} - 1\right) \end{aligned} \qquad (22)$$

Fig. 11 shows visually the meaning of relation (21) depicting A.M. and P.M. as extremes of amplitude and phase distance to critical point $(-180°,\ 0\ dB)$.

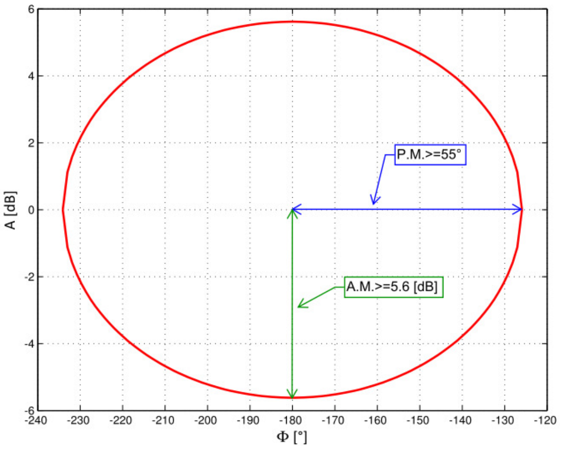


Fig. 11. Stability margin specifications on Nichols chart for $M_m = 1.1$

Second type of specifications refers to quality of disturbance compensation i.e. ability to maintain nominal rotation speed in spite of acting wind gust. Following relation needs to be satisfied:

$$\frac{\omega(s)}{v_w(s)} = \left|\frac{G_{d,i}(s)}{1 + G_i(s)C(s)}\right| \le |G_{dist}(s)| \qquad (23)$$

for $\forall G_{d,i} \in \mathcal{G}_d$ and $\forall G_i \in \mathcal{G}$. $G_{dist}(s)$ is defined as:

$$G_{dist}(s) = \frac{1.8s^2 + 3.557s}{s^2 + 1.456s + 0.526} \qquad (24)$$

Unit step response and frequency amplitude characteristic of (24) is shown on Fig. 12.

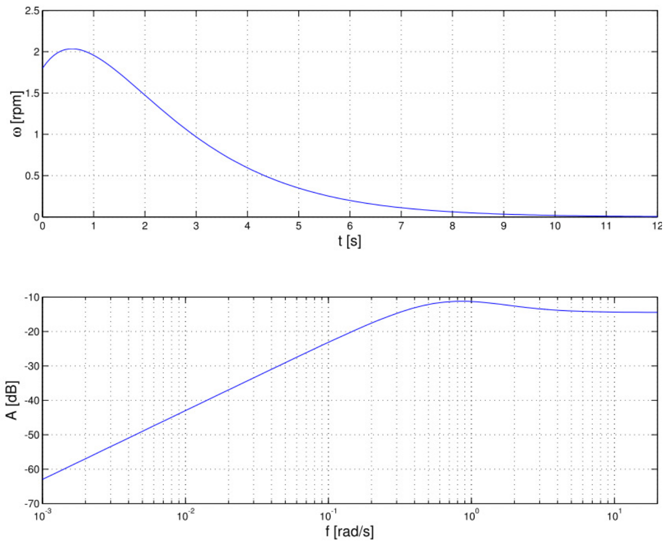


Fig. 12. Specified characteristic of response to unit step disturbance and accompanying frequency amplitude characteristic

There is no particular restriction on the initial part of the response which explains almost constant attenuation frequency characteristic in high frequency range. Namely, process response cannot suddenly jump at $t = 0\ s$ to certain value in a step like manner so there would be no point in defining any particular shape of response in initial period of time as this requires unnecessarily aggressive and complex controller design due to dominate high frequency design requests.

7. CONTROLLER DESIGN

Using defined specifications, discrete frequency set Ω and templates generated for $\forall\omega_i \in \Omega$ it possible to calculate, solving systems of quadratic inequalities, necessary stability and disturbance boundaries and depict their intersection (resultant boundaries) on Nichols chart.

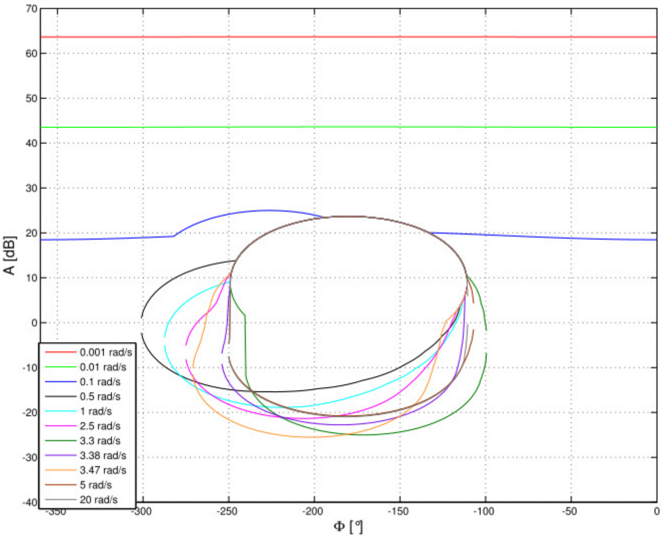


Fig. 13. Resultant boundaries on Nichols chart

A $12\,m/s$ linearized model was chosen as nominal plant process $G_0(s)$ so all the boundaries on Nichols chart were calculated in reference to this model. This in turn means that adequate loopshaping of open loop characteristic $L_0(s) = C(s)G_0(s)$ that satisfies given boundaries results in a fact that all closed/open loop characteristics satisfy accompanying closed/open loop characteristic. Controller transfer function is given as:

$$C(s) = -1.65\frac{(s + 0.5)(s + 1.4)}{s(s + 5)} \qquad (25)$$

On Fig. 14. it can be observed that point $\omega_4 = 0.5\ rad/s$ is not completely out of its boundary which was done on purpose as this would require movement of controller zero $z_1 = -0.5$ even closer to zero. This in turn weakens the integral action of the controller necessary for precision of stationary part of response as this zero would nearly cancel its action. Rise in complexity of the controller would be able to solve this issue but this is where QFT transparency comes handy as it enables us to make tradeoffs in controller design. It was also observed that neglecting of initial shape of response on Fig. 12 made the design easier with no evident loss in quality of response i.e. more simple controller was obtained that performs almost equally well. $z_2 = -1.4$ was inserted in order to obtain raise in the phase value of open loop so as to circumvent the round boundaries on their lower right part. “Optimal” QFT controllers would have to minimize the cost of feedback, meaning that minimum of bandwidth should be utilized to satisfy given specifications. In order to gain such a controller points on nominal open loop characteristic $L_0(j\omega)$ should be maximally close to their boundary.

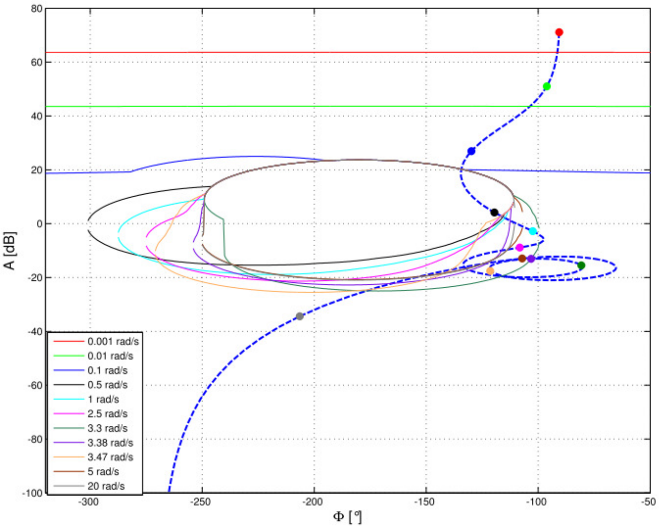


Fig. 14. Nominal open loop on Nichols chart vs calculated open loop boundaries

8. DESIGN VALIDATION

It remains to perform a validation of design by checking if for all family members prescribed specifications are satisfied. Fig. 15. shows validation of stability on all linear models.

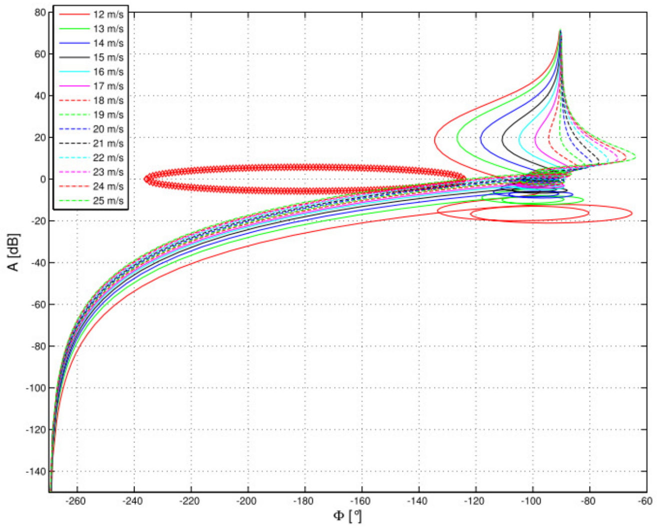


Fig. 15. Validation of stability specifications in frequency domain (boundary marked with red diamond markers)

Likewise Fig. 16. and Fig. 17. validate satisfactory behaviour in time and frequency domain of all linear models. Small ripple superimposed on rotor speed response stems from the fact that the tower top is oscillating towards/from the direction of wind meaning that relative wind speed v_w^* given by (9) is oscillatory changing. This in turn causes oscillatory changes in angle of attack of all blade sections and, cumulatively, introduction of oscillatory component in turbine drive torque.

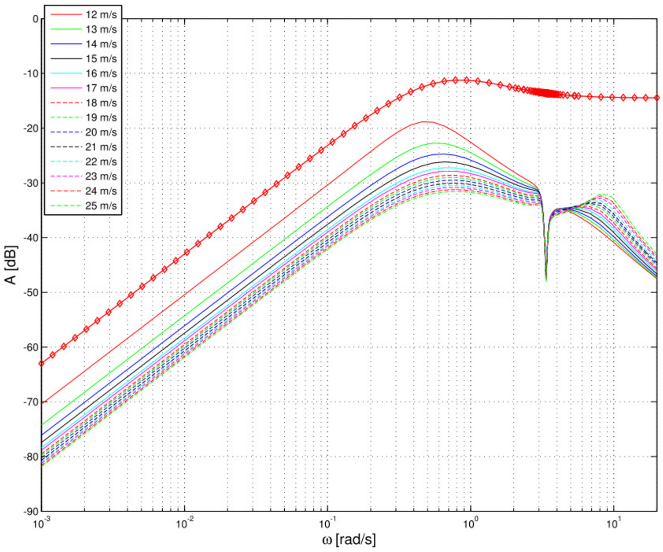


Fig. 16. Validation of disturbance specifications in frequency domain (boundary marked with red diamond markers)

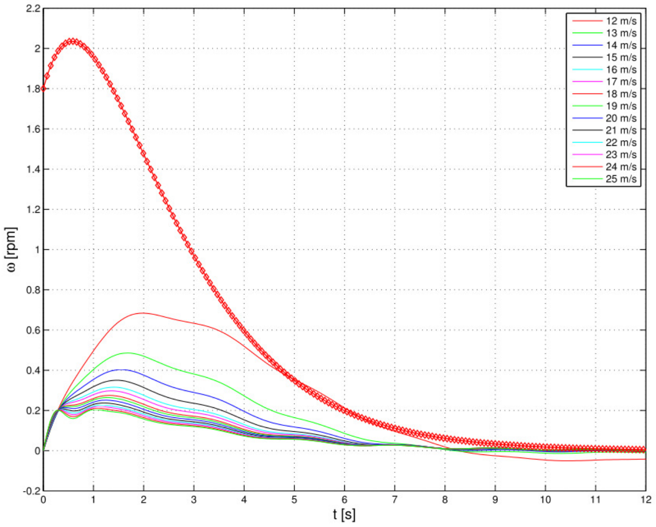


Fig. 17. Validation of disturbance specifications in time domain (boundary marked with red diamond markers)

So far validation was performed on family of linear models obtained by linearization of model given by (14) for wind speed ranging from 12 to 25 m/s. Plot on Fig. 18. confirms that given specifications have been satisfied even for nonlinear model. No particular differences are observed comparing validation performed on family of linear models and on nonlinear model. It is interesting to observe how the controller $C(s)$ behaves when faced with a simulation on a professional wind turbine simulation tool (Bladed, [8]) that also offers possibility of obtaining family of linear models of very high order (>40). This model besides the first and second mode of fore-aft movement, also includes equal number of side-side tower modes and rotor in-plane and out-plane modes that were neglected in simplified nonlinear model given by (14). The response of rotor speed and pitch angle when performing simulations in Bladed is given on Fig. 19. It shows almost equal “main” dynamic of responses compared to responses of simplified nonlinear process. Response from Bladed contains though richer contents due to high order effects originating, among others, from in-plane and out-plane movement of blade sections in reference to their stationary position. Controller $C(s)$ would perform better if it was designed upon boundaries generated in reference to high order linear models obtained from Bladed. In this case an introduction of gain scheduling that reduces gain in high wind speed range would aid the controller and reduce the blade oscillatory movement thus reducing tear-and-wear.

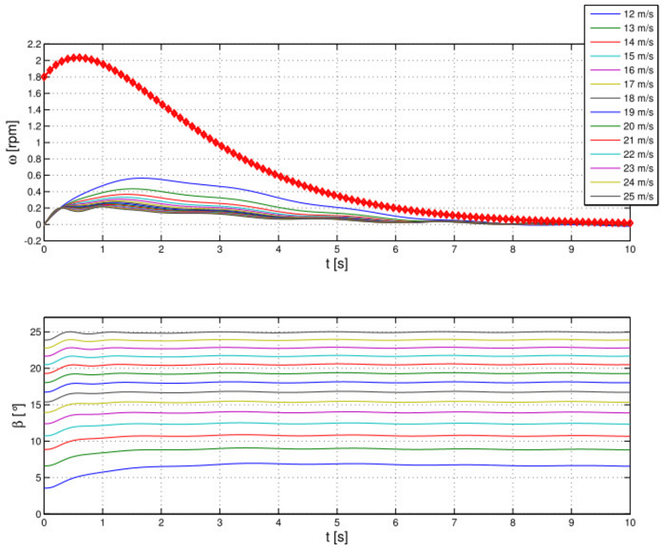


Fig. 18. Validation on nonlinear model

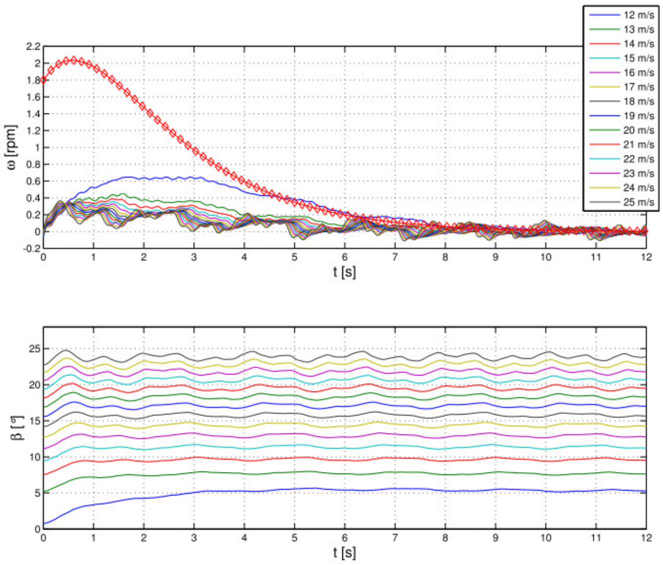


Fig. 19. Validation in Bladed

9. CONCLUSION

QFT proved to be an adequate method for synthesis of rotor speed controller despite existing variations in wind turbine dynamics. This should not come as a surprise as it was mentioned earlier that QFT had been very successfully integrated in helicopter and airplane control systems. Usually robust controllers are of high order but in this case, due to transparency of the QFT and its ability to explicitly address the uncertainty of wind turbine dynamics, an efficient controller of second order was obtained that uses no other aids (feedforward action, gain scheduling, etc.) to achieve specified closed loop behavior. Strong point of this method is also the ability it gives the user to perceive when the combination of process uncertainty and performance demands poses to big of an obstacle for chosen control structure. In this case it was possible to conclude that (judging by Bladed simulation results) introduction of gain reducing element scheduled on pitch angle would aid the performance of the controller in high wind regime thus obtaining a hybrid solution that combines adaptive and robust algorithms.

ACKNOWLEDGMENTS

This work was financially supported by Končar – Electrical Engineering Institute and the Ministry of Science Education and Sports of the Republic of Croatia.

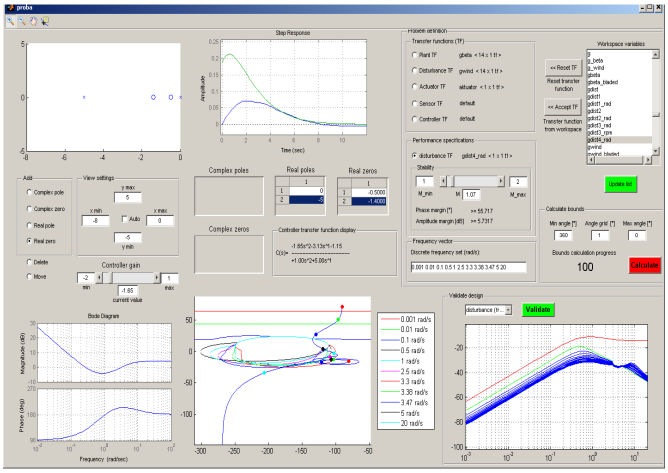
REFERENCES

1. Damping of Wind Turbine Tower Oscillations through Rotor Speed Control. M. Jelavić, N. Perić, I. Petrović. Monaco : s.n., 2007. International Conference on Ecologic Vehicles & Renewable Energies.
2. T. Burton, D. Sharpe, N. Jenkins, E. Bossanyi. Wind energy handbook. s.l. : John Wiley and sons, 2001.
3. Modelling and control of variable-speed wind trubine drive-system dynamics. P. Novak, T. Ekelund, I.Jovik, B. Schmidtbauer. 4, 1995, Control system magazine, Vol. 15, pp. 28-33.
4. In memoriam – The life of prof. Isaac Horowitz. Chait, Y. i Jayasuriya, S. prosinac 2005, IEEE Control Systems Magazine.
5. Houpis, C.H., Rasmussen, J.R. i Garcia-Sanz, M. Quantitative Feedback Theory – Fundamentals and Application. 2. edition. Boca Raton : CRC Press, 2006.
6. E. L. van der Hooft, P. Schaak, T. G. van Engelen. Wind turbine control algorithms, Dowec WP1 - task 3 ECN-C-03-111. Petten, Netherlands : ECN Wind Energy, 2003.
7. F. D. Bianchi, H. De Battista, R.J. Mantz. Wind turbine control system, principles, modeling and gain scheduling design. s.l. : Springer, 2006.
8. Bossanyi, E. GH Bladed user's manual. Bristol : Garrad Hassan and Partners Limited, 2009.
9. Aranda J., Díaz J.A., Dormido S.,. SISO-QFTIT An interactive software tool for the design of robust controllers using the QFT methodology USER'S GUIDE. Madrid : U.N.E.D Departamento de Informática y Automática , 2005.
10. Borghesani, C., Chait, Y. and Yaniv, O. The QFT Frequency Domain Control Design Toolbox For Use with MATLAB. 3. izdanje. s.l. : Terasoft, Inc., 2003.
11. MIMOQCAD: a Mathematica based multivariable control system CAD package. Breslin, S.G., Grimble, M.J., Houpist, C.H. 1996. Symbolic Computation for Control (Digest No: 1996/078), IEE Colloquium on .
12. Identification of Wind Turbine Model for Controller Design. M. Jelavić, N. Perić, I. Petrović. Portorož, Slovenia :

s.n., 2006. Proceedings of the 12th International Power Electronics and Motion Control Conference. pp. 1608-1613.

Appendix A. QFT GUI

Although several tools exist that offer the possibility of interactive QFT controller design (see [9], [10], [11]), an attempt was made to implement a simple QFT tool within Matlab® environment. As a result QFT GUI application (see Fig. 20) was implemented which offers its user the possibility to define the process in a structure given by (18). Furthermore, sensor and actuator dynamics can also be selected. This test version of GUI enables defining of two types of specification that were used in this article (disturbance rejection, stability margin). Calculate button translates given specification on to the Nichols chart (in the form of open loop boundaries) where controller design commences. The controller is designed interactively as the effects of either movement, deletion or addition of zeroes and poles are seen on Nichols chart. At any point the user can validate quality of controller design in time and frequency domain. It is important to stress that the validation is performed on the user defined set of linear processes. It is up to user to validate if the controller design is adequate on a nonlinear process model.



Problem definition

Transfer functions (TF)

○ Plant TF	gbeta < 14 x 1 tf >
○ Disturbance TF	gwind < 14 x 1 tf >
○ Actuator TF	aktuator < 1 x 1 tf >
○ Sensor TF	default
○ Controller TF	default

Workspace variables: g, g_beta, g_wind, gbeta, gbeta_bladed, gdist, gdist1, gdist1_rad, gdist2, gdist2_rad, gdist3_rad, gdist3_rpm, gdist4_rad, gwind, gwind_bladed

Performance specifications

● disturbance TF gdist4_rad < 1 x 1 tf >

Stability: M_min 1, M 1.07, M_max 2

Phase margin [°] >= 55.717

Amplitude margin [dB] >= 5.7317

Frequency vector

Discrete frequency set (rad/s): 0.001 0.01 0.1 0.5 1 2.5 3.3 3.38 3.47 5 20

Calculate bounds

Min angle [°] -360 Angle grid [°] 1 Max angle [°] 0

Bounds calculation progress 100

View settings: y max 5, x min -8, ☐ Auto, x max 0, y min -5

Controller gain: -2 min, -1.65 current value, 1 max

Complex poles

Complex zeros

Real poles

	1
1	0
2	-5

Real zeros

	1
1	-0.5000
2	-1.4000

Controller transfer function display

```
      -1.65s^2-3.13s^1-1.15
C(s)= ---------------------
      +1.00s^2+5.00s^1
```

Validate design: disturbance (fr...

Fig. 20. QFT GUI application

ROBUST PID CONTROLLER DESIGN FOR COUPLED-TANK PROCESS

Ivan Holič and Vojtech Veselý*

**Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology*
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel: +421-2-60291-539, e-mail: ivan.holic@stuba.sk, vojtech.vesely@stuba.sk

Abstract: The paper deals with the design of the robust PID controller for real uncertain Coupled-Tank process in the frequency domain. Only the first independent tank is considered (single-input single-output system). Robust controller is designed in two ways. The first approach is performed with the Edge Theorem and the Neimark's D-partition method for the affine model and the second one is performed with the modification of the Neimark's D-partition which ensures desired phase margin.

Keywords: SISO, Robust PID Controller, Edge Theorem, D-partition, phase margin.

1. INTRODUCTION

Control of real processes inherently includes uncertainties (modeling errors due to linearization and approximation, disturbances etc.), which have to be considered in the adequate control design. Therefore robustness belongs to an important control design qualities: closed loop system stability and performance should be guaranteed over the whole uncertainty domain, (Vesely et al. 2006).

There exist various approaches to robust stability analysis and robust control design for uncertain linear systems. In this paper the frequency domain PID controller design for real Coupled-Tank process is considered. Liquid tank processes play important role in industrial application such as in food processing, filtration, pharmaceutical industry, water publication system, industrial chemical processing and spray coating (Ramli et al. 2009). Many industrial applications are concerned with level of liquid control, may it be a single loop level control or sometimes multi loop level control (Ramzad et al. 2008). In this paper only the first tank with liquid is used (SISO).

The paper is organized as follows. The next section gives details about Coupled-Tank process. Section 3 introduces a PID controller design using two approaches. In section 4 some results of robust PID controller design are presented. Several step responses of closed-loop system with proposed PID controller are plotted there. Finally, conclusion is given in section 5.

2. COUPLED-TANK PROCESS

The industrial Coupled-Tank process is one of the real processes built for control education and research at Institute of Control and Industrial Informatics. The apparatus consists of two tanks (T1 and T2 in Fig. 1), which can be coupled using valve V12 (the manual valve). Therefore the Coupled-Tank process with two tanks represents a multi-input multi-output (MIMO) system for opened valve V12 or two independent single-input single-output (SISO) systems for closed valve V12. Both tanks are made of Plexiglas. These two tanks are mounted on a platform with a metering scale before each tank indicating the approximate liquid level in tank. Exact liquid level in each tank is measured using an electronic sensor. Other components of system are liquid basin (reservoir), two pumps (Pump1 and Pump2 in Fig. 1), two outlet valves (V1 and V2 in Fig. 1) and electronic circuit communicating with *LABREG* software in computer. This software is made for identification and control of real processes. The *LABREG* operates in MATLAB using toolboxes *SIMULINK*, *Ident*, *Control* and *Real Time*. Cooperation between Coupled-Tank process and computer and *LABREG* software is ensured using Advantech data acquisition card of type PCI 1711. More about *LABREG* and mentioned toolboxes can be found in (Kajan et al. 2007).

The paper deals with design of robust controller for SISO system (valve V12 is closed), consequently there can be used one or two independent tanks. Only one tank process is considered, therefore the purpose is to control liquid level in the first tank by the inlet liquid flow from the first electronic DC pump (Pump1). The process input is $u_1(t)$ (voltage input to Pump1) and the output is $h_1(t)$ (liquid level in the first tank – T1). Input power is bounded by interval $\langle 0,10 \rangle$ volts and output signal is measured using electronic sensor. Q_{i1} and Q_{o1} in Fig. 1 denote the inlet and outlet flow rates for T1 respectively. Outlet flow is affected by electronic outlet valve (V1), which can be set manually from 0 to 10 volts (for 0 [V] is closed, for 10 [V] fully opened) and represent perturbation.

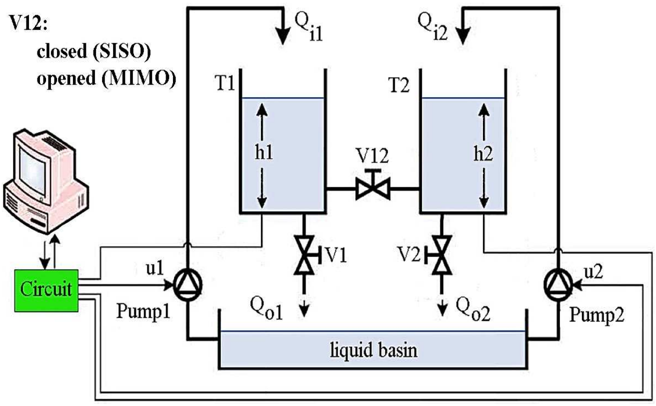


Fig. 1. Coupled-Tank process

3. PRELIMINARIES AND PROBLEM FORMULATION

3.1 Robust controller design using the Edge Theorem

For this theory affine model of the plant is used. It is used advantageously because a part of parameters of the real process vary dependently. Affine model is in this form

$$G(s)=\frac{b_0(s)+\sum_{i=1}^{p}q_ib_i(s)}{a_0(s)+\sum_{i=1}^{p}q_ia_i(s)} \tag{1}$$

where $b_0(s),\ b_i(s)$ and $a_0(s),\ a_i(s)$ are polynomials of numerator and denumerator and uncertain parameters q_i are from interval $\left[\underline{q_i},\overline{q_i}\right]$.

Each real uncertain parameter q_i varies within a *p*-dimensional domain. In other words, the parameter vector $q^T=\left[q_1,\ldots,q_p\right]$ varies in the hypercube (Ackerman 1997, Bhattacharyya et al. 1995)

$$Q=\left\{q\mid q_i\in\left[\underline{q_i},\overline{q_i}\right],\ i=1,2,...,p\right\} \tag{2}$$

Alternating minimal ($\underline{q_i}$) and maximal ($\overline{q_i}$) value of q_i , we obtain the polytope with 2^p vertices. Each vertex can be represented by a transfer function with constant coefficients. Transfer function (1) describes a polytopic system.

Consider the controller described by transfer function

$$G_R(s)=\frac{F_1(s)}{F_2(s)} \tag{3}$$

where $F_1(s)$ and $F_2(s)$ are polynomials with constant parameters.

If parameter q varies within a hypercube, it generates a polytopic family of closed-loop characteristic polynomials described as follows

$$\begin{aligned}p(s,q)&=b_0(s)F_1(s)+a_0(s)F_2(s)\\&+\sum_{i=1}^{p}q_i\left[b_i(s)F_1(s)+a_i(s)F_2(s)\right]\end{aligned} \tag{4}$$

or in more general form according to (Hypiusová et al. 2007, Hypiusová et al. 2008)

$$p(s,q)=p_0(s)+\sum_{i=1}^{p}q_ip_i(s) \tag{5}$$

where $q_i\in Q$.

Theorem 1 - Edge Theorem (Hypiusová et al. 2007)

The polytopic family of characteristic polynomials (5) is stable if and only if the edges of set Q are stable.

The Edge Theorem gives an elegant solution to the problem of determining the root space of polytopic systems. Therefore the robust stability of such systems can also be determined (Bhattacharyya et al. 1995). The stability condition for polytopic family of characteristic polynomials (5) is given in the following theorem using robust Hurwitz stability criteria. Using the Bialas Theorem stability of each edge of the polytopic box can be checked.

Theorem 2 - Bialas Theorem (Hypiusová et al. 2007)

The polynomial family

$$p(s,Q)=\left\{\lambda p_a(s)+(1-\lambda)p_b(s),\ \lambda\in\left[0,1\right]\right\} \tag{6}$$

is stable if and only if:

- $p_a(s),\ p_b(s)$ are stable,
- the matrix $\left(H_n^{(b)}\right)^{-1}H_n^{(a)}$ has no nonpositive real eigenvalues

where matrices $H_n^{(b)}$ and $H_n^{(a)}$ are Hurwitz matrices of following polynomials

$$\begin{aligned}p_b(s)&=p_{b0}+p_{b1}s+\ldots+p_{bn}s^n\ ,\ \ p_{bn}>0\\p_a(s)&=p_{a0}+p_{a1}s+\ldots+p_{an}s^n\ ,\ \ p_{an}>0\end{aligned} \tag{7}$$

By applying the Neimark's D-partition method with Edge Theorem, the required stability degree of closed-loop system can be guaranteed. The controller coefficients are chosen so that the vertices and edges of polytopic system are stable.

3.2 Robust controller design with desired phase margin

This approach is in details described in (Hypiusová et al. 2010a, Hypiusová et al. 2010b), where closed-loop system with $G_R(s)$ (transfer function of PID controller) and $G(s)$ (transfer function of the real plant) is considered. The real perturbed plant with unstructured inverse additive uncertainties is described as follows (Vesely et al. 2006)

$$G(s)=G_0(s)(I+w_{ia}(s)\Delta_{ia}(s)G_0(s))^{-1} \tag{8}$$

where $G_0(s)$ is nominal model, $w_{ia}(s)$ is stable weighting scalar transfer function, $\Delta_{ia}(s)$ is normalized matrix of unstructured uncertainty ($\Delta_{ia}(s) \le 1$).

Weighting scalar transfer function must be chosen for all ω in accordance with

$$\left| w_{ia}(\omega) \right| \ge l_{ia}(\omega) \tag{9}$$

and weighting function $l_{ia}(\omega)$ is the maximum singular value of difference $G_0(j\omega) - G_k(j\omega)$ for N ($k = 1, \ldots, N$) known transfer functions:

$$l_{ia}(\omega) = \max_k \sigma_M \left(G_0(jw) - G_k(jw) \right) \tag{10}$$

The nominal model stability is equivalent to the stability of the $M\Delta$-structure. We thus need to derive the robust stability conditions using $M\Delta$-structure for checking the stability according to (Hypiusová et al. 2010a, Skogestad et al. 2005) as follows

$$\sigma_M(M_0(s)) < \frac{1}{l_{ia}(\omega)} \tag{11}$$

and

$$M_0(s) = \frac{G_0(s)}{1 + G_R(s) G_0(s)} \tag{12}$$

where $G_R(s)$ and $G_0(s)$ are transfer functions of PID controller and nominal model. Nominal model has in this case the following form

$$G_0(s) = \frac{B_0(s)}{A_0(s)} = \frac{\left(\sum_{i=1}^{N} B_i(s) \right) \Big/ N}{\left(\sum_{i=1}^{N} A_i(s) \right) \Big/ N} \tag{13}$$

where B_i, A_i $(i = 1, \ldots, N)$ are polynomials of numerator and denumerator of N identified transfer functions of the real process (in N working points).

Consider transfer function of PID controller

$$G_R(s) = \frac{K_D s^2 + K_P s + K_I}{s} = K_P + \frac{K_I}{s} + K_D s \tag{14}$$

The robust PID controller design is performed with the modification of the Neimark's D-partition which ensures stability and desired phase margin of the closed-loop system with nominal model described in (13) as in (Hypiusová et al. 2010a).

The closed-loop system characteristic equation for nominal model is

$$1 + G_R(s) G_0(s) = 0 \tag{15}$$

From (15) the relationship between $G_R(s)$ and $G_0(s)$ can be obtained

$$G_R(s) = -\frac{1}{G_0(s)} \;\Rightarrow\; K_P + \frac{K_I}{s} + K_D s = -\frac{A_0(s)}{B_0(s)} \tag{16}$$

Using substitution $s = j\omega$, real and imaginary part of equation (16) are

$$\begin{aligned} &\mathrm{Re}: K_P = -\frac{A_0(j\omega)}{B_0(j\omega)} \\ &\mathrm{Im}: -\frac{K_I}{\omega} j + K_D j\omega = -\frac{A_0(j\omega)}{B_0(j\omega)} \end{aligned} \tag{17}$$

The D-curve in the complex plane $\mathbb{C}$ for parameter K_P can be plotted from real part (17) by changing value of ω step by step in interval $(0, \infty)$. Similarly it is with imaginary part (17), from where D-curve for parameters K_I and K_D can be plotted. Parameters of PID controller are obtained in two steps. In the first one it is possible to plot D-curve for K_{P1} and K_D (PD controller is obtained) and in the second one for parameters K_{P2} and K_I (PI controller).

When a phase margin is considered, the closed-loop system characteristic equation (15) can be rewritten according to (Hypiusová et al. 2010a, Hypiusová et al. 2010b)

$$1 + G_R(s) G_0(s) e^{-j\varphi} = 0 \tag{18}$$

where φ is the angle of desired rotation in radians (phase margin) and in this way it is possible to rotate the frequency plot. From (18) real and imaginary parts can be obtained, which describe the D-curves as

$$\begin{aligned} &\mathrm{Re}: K_P = -\frac{A_0(j\omega)}{B_0(j\omega) e^{-j\varphi}} \\ &\mathrm{Im}: -\frac{K_I}{\omega} j + K_D j\omega = -\frac{A_0(j\omega)}{B_0(j\omega) e^{-j\varphi}} \end{aligned} \tag{19}$$

Parameters of PD and PI controller are chosen from plotted D-curves. The final PID controller is represented as series connection of PD and PI controller and can be calculated as follows

$$\begin{aligned} G_R(s) &= \left(K_{P1} + K_D s \right) \left(K_{P2} + \frac{K_I}{s} \right) \\ &= \frac{K_{P2} K_D s^2 + \left(K_{P1} K_{P2} + K_D K_I \right) s + K_{P1} K_I}{s} \end{aligned} \tag{20}$$

The first controller (PD) is used for stabilization of system and the second one (PI) ensures desired phase margin.

4. DESIGN OF ROBUST PID CONTROLLER FOR COUPLED-TANK PROCESS

In this case, system step response is examined. Transfer function of the system in all three working points is obtained from the output step response of open loop system using BJ (Box-Jenkins) method of identification. More about BJ method of identification can be found in (Pintelon et al. 2006a, Pintelon et al. 2006b). We consider transfer functions of a liquid level in the first tank (see Fig. 1) obtained by identification in three working points:

WP1 (working point 1):

$$water\ pump\ voltage\ (input\ voltage) = 2,5\ [V]$$

$$step\ of\ water\ pump\ voltage\ in\ time\ t\ to\ 2,75\ [V]$$

$$outlet\ valve\ voltage\ (perturbation) = 7\ [V]$$

Transfer function is obtained by BJ method of identification as

$$G_{WP1}(s) = \frac{2,9676s + 29,9239}{816,9049s^2 + 202,6853s + 1} \qquad (21)$$

WP2 (working point 2):

$$water\ pump\ voltage\ (input\ voltage) = 3,5\ [V]$$

$$step\ of\ water\ pump\ voltage\ in\ time\ t\ to\ 4,5\ [V]$$

$$outlet\ valve\ voltage\ (perturbation) = 9\ [V]$$

Transfer function is obtained by BJ method of identification as

$$G_{WP2}(s) = \frac{0,7824s + 7,8848}{360,5413s^2 + 82,2612s + 1} \qquad (22)$$

WP3 (working point 3):

$$water\ pump\ voltage\ (input\ voltage) = 4\ [V]$$

$$step\ of\ water\ pump\ voltage\ in\ time\ t\ to\ 5\ [V]$$

$$outlet\ valve\ voltage\ (perturbation) = 10\ [V]$$

Transfer function is obtained by BJ method of identification as

$$G_{WP3}(s) = \frac{0,5461s + 5,5007}{297,473s^2 + 63,8584s + 1} \qquad (23)$$

Transfer function of the nominal model is obtained by (13) from the above three working points

$$G_0(s) = \frac{1,431s + 14,44}{491,6s^2 + 116,3s + 1} \qquad (24)$$

The respective polytopic (affine) model of the Coupled-Tank process is described by

$$G(s) = \frac{b_0(s) + q_1 b_1(s) + q_2 b_2(s)}{a_0(s) + q_1 a_1(s) + q_2 a_2(s)} \qquad (25)$$

where q_i, $i = 1, \ldots, N$ are uncertain coefficients and polynomials of numerator and denumerator are

$$b_0(s) = 1,75685s + 17,7123$$
$$b_1(s) = -0,11815s - 1,1921$$
$$b_2(s) = -1,0926s - 11,0195$$
$$a_0(s) = 557,1889s^2 + 133,2719s + 1$$
$$a_1(s) = -31,5341s^2 - 9,2014s$$
$$a_2(s) = -228,1818s^2 - 60,21205s$$

More about practical procedure on how to get the values of polynomials of numerator and denumerator can be found in (Vesely et al. 2006).

4.1 Robust controller design using the Edge Theorem

The robust PID controller is proposed using the Edge Theorem approach for the polytopic model defined in (25). The required degree of stability α is 0. Using Neimark's D -partition method the robust PID controller is designed

$$G_R(s) = \frac{1,6s^2 + 4s + 0,1}{s} \qquad (26)$$

Theorem 2 (Bialas Theorem) verifies stability and it can be said that the closed-loop polytopic system with robust controller is stable and the achieved degree of stability α in 4 vertices is 0,0258. Proposed robust PID controller (26) was set on the real process (first tank). Step responses in all three working points are depicted in Fig. 2,3,4.

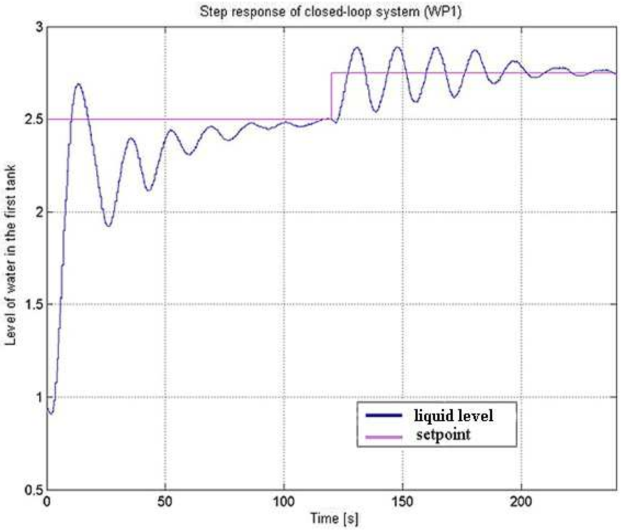


Fig. 2. Step response of closed-loop system in WP1 (the first working point)

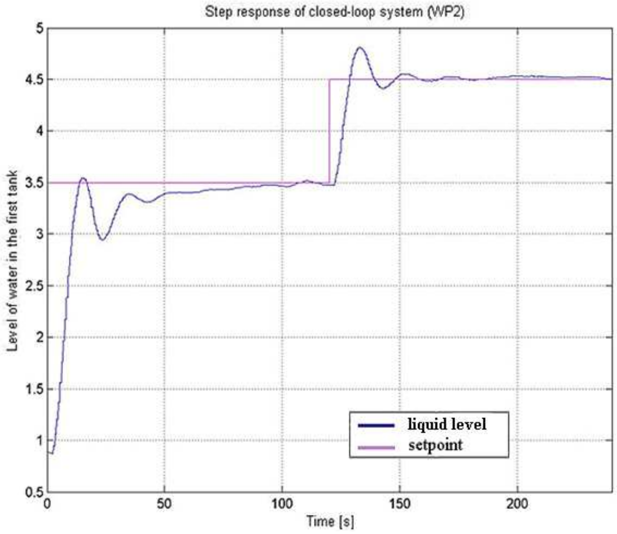


Fig. 3. Step response of closed-loop system in WP2 (the second working point)

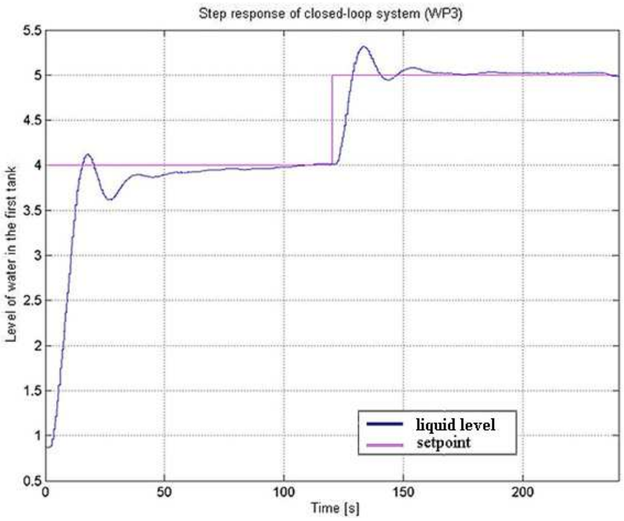


Fig. 4. Step response of closed-loop system in WP3 (the third working point)

4.2 Robust controller design with desired phase margin

Consider the transfer function of nominal model as in (24). Required phase margin φ_R is 45°. In the first step D-curve for parameters K_{P1} and K_D (PD controller) is plotted. These parameters are chosen from stable region above magenta line (see Fig. 5) because it is necessary to stabilize the system (the parameters need not be chosen only from the blue line in this step).

The PD controller has following coefficients: $K_{P1} = 3{,}569$ and $K_D = 0{,}6849$. Poles of characteristic equation of closed-loop system with PD controller are $-0{,}1332 \pm 0{,}2981\mathrm{j}$.

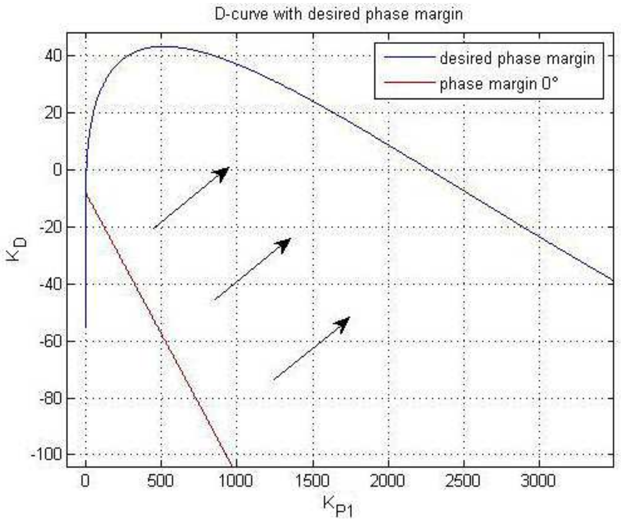


Fig. 5. D-curve for parameters K_{P1} and K_D

The next step consists in design of PI controller for nominal model with PD controller. Parameters of PI controller need to be chosen from plotted desired phase margin 45° (the blue line in Fig. 6). Chosen coefficients of PI controller are $K_{P2} = 0{,}3542$ and $K_I = 0{,}01713$. Transfer function of the final PID controller is

$$G_R(s) = \frac{0{,}2426s^2 + 1{,}276s + 0{,}06114}{s} \qquad (27)$$

Poles of characteristic equation of closed-loop system with PID controller (27) are $-0{,}0914 \pm 0{,}1398\mathrm{j}$ and $-0{,}0643$.

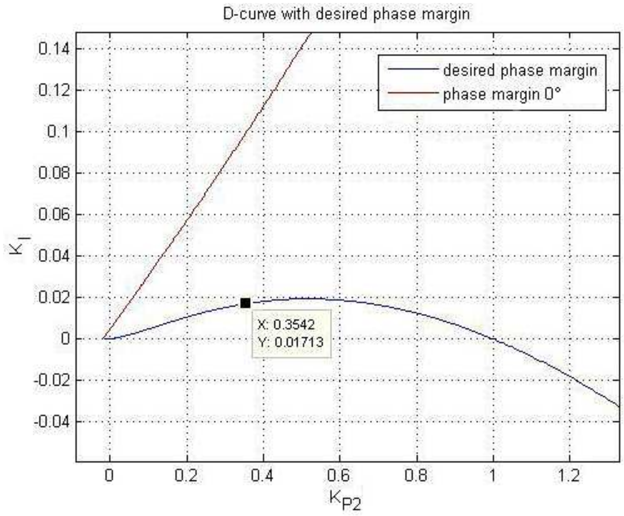


Fig. 6. D-curve for parameters K_{P2} and K_I

Fig. 7 and 8 show that the desired phase margin and robust stability are satisfied. Proposed PID controller (27) was set on the real process. Step responses in all three working points are plotted in Fig. 9, 10 and 11.

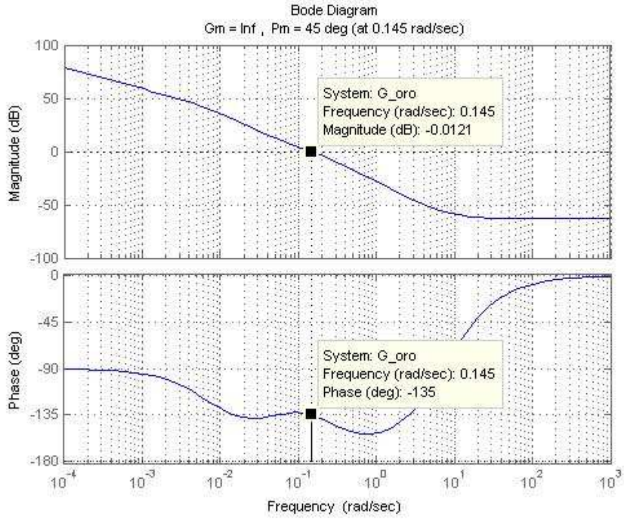


Fig. 7. Bode characteristics for Coupled-Tank process

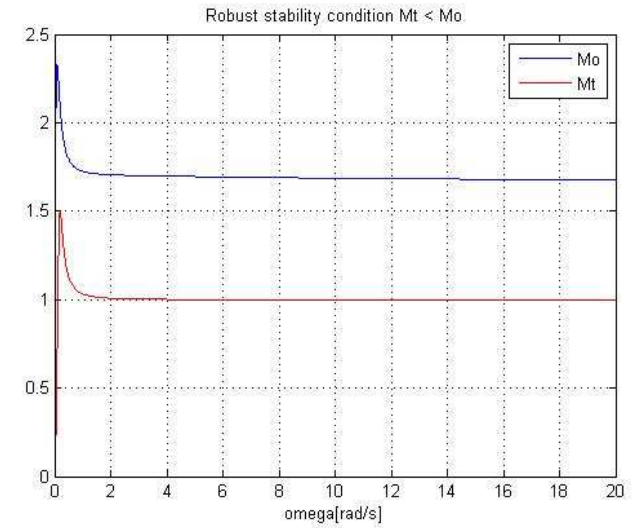


Fig. 8. Robust stability condition

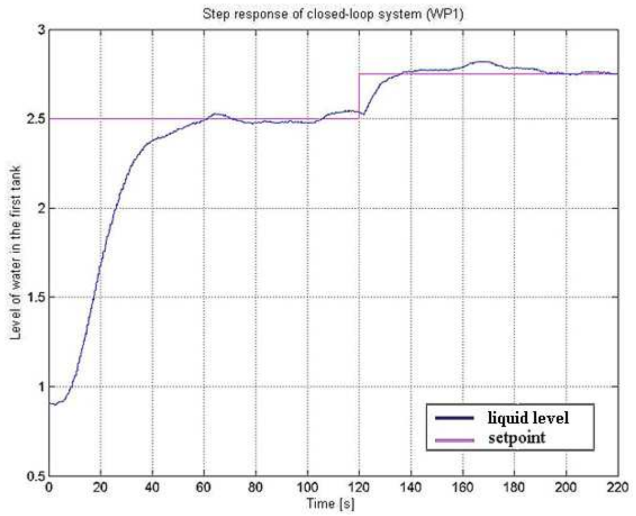


Fig. 9. Step response of closed-loop system in WP1

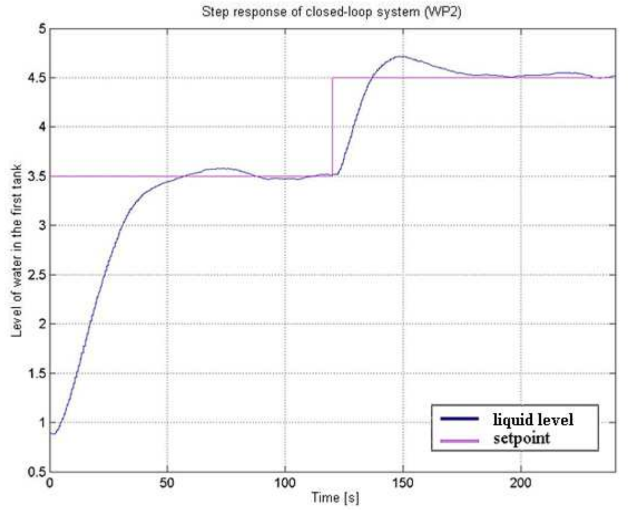


Fig. 10. Step response of closed-loop system in WP2

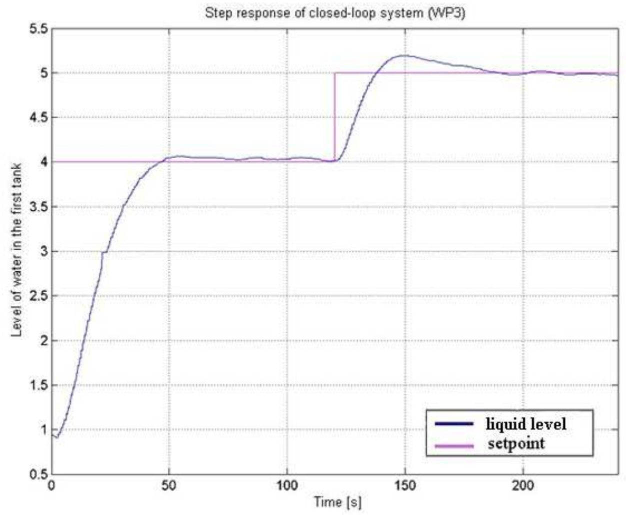


Fig. 11. Step response of closed-loop system in WP3

5. CONCLUSION

In this paper two approaches of robust PID controller design for real unstable Coupled-Tank process have been presented. The first one is Edge Theorem and the second approach is based on modification of the Neimark's D-partition method, which ensures not only stability of closed-loop system but also desired phase margin. From view of control quality, the robust controller design with desired phase margin using Neimark's D-curves is better. Results obtained in the paper will be used for control education at Institute of Control and Industrial Informatics. Further aspects of the studied approach concerning robust controller design, closed-loop or open loop identification of Coupled-Tank process with cascade controller, are under research.

ACKNOWLEDGMENTS

The work has been supported by Grant N1/0544/09 and APVV-0211/10.

REFERENCES

Ackerman, J. (1997). *Robust Control – Systems with Uncertain Physical Parameters.* Springer-Verlag London, 406 pp., ISBN 0-387-19843-1.

Bhattacharyya, S. P., Chapellat, H. and Keel, L. H. (1995). *Robust Control: The parametric Approach.* Prentice Hall, 647 pp., ISBN 0-13-781576-X.

Hypiusová, M. and Osuský, J. (2008). Robust Controller Design for Modular Servo System. In: *PROCESS CONTROL 2008 : Proceedings of the 8th International Scientific- Technical Conference.* Kouty nad Desnou, Czech Republic, June 9-12, ISBN 978-80-7395-077-4.

Hypiusová, M. and Osuský, J. (2010a). Robust controller design for magnetic levitation model. In: *AT&P Journal Plus.* No. 1, pp. 100-104, ISSN 1336-5010.

Hypiusová, M. and Osuský, J. (2010b). PID Controller Design for Magnetic Levitation Model. In: *Cybernetics and Informatics : International Conference SSKI SAV,* Vyšná Boca, Slovak Republic, February 10.-13., ISBN 978-80-227-3241-3.

Hypiusová, M., Osuský, J. and Kajan, S. (2007). Robust Controller Design Using Edge Theorem for Modular Servo System. In: *Technical Computing Prague 2007 : 15th Annual Conference Proceedings*, Prague, Czech Republic, November 14, ISBN 978-80-7080-658-6.

Inampudi, N. K. (2009). Developing, Implementing and Assessing Coupled-Tank Experiments in an Undergraduate Chemical Engineering Curriculum. A Thesis presented to the Faculty of the Graduate School at the University of Missouri, July.

Kajan, S. and Hypiusová, M. (2007). Labreg Software for Identification and Control of Real Processes in Matlab. In: *Technical Computing Prague 2007 : 15th Annual Conference Proceedings*, Prague, Czech Republic, November 14., ISBN 978-80-7080-658-6.

Numsomran, A., Suksri, T. and Thumma, M. (2007). Design of 2-DOF PI Controller with Decoupling for Coupled-Tank Process. In: *International Conference on Control, Automation and Systems 2007*, COEX, Seoul, Korea, October 17.-20., pp. 339-344.

Pintelon, R., Rolain, Y. and Schoukens, J. (2006a). Box-Jenkins identification revisited-Part II: Applications. In: *Automatica*, No. 42, 2006, pp. 77-84.

Pintelon, R. and Schoukens, J. (2006b). Box-Jenkins identification revisited-Part I: Theory. In: *Automatica*, No. 42, 2006, pp. 63-75.

Ramli, M. S., Raja Ismail, RM. T., Ahmad, M. A., Mohamad Nawi, S. and Mat Hussin, M. A. (2009). Improved Coupled Tank Liquid Levels System Based on Swarp Adaptive Tuning of Hybrid Proportional-Integral Neural Network Controller. In: *American Journal of Engineering and Applied Sciences 2*, No. 4, pp. 669-675, ISSN 1941-7020.

Ramzad, M. F. and MD Rozali, S. (2008). Modeling and Controller Design for Coupled-Tank Liquid Level System: Analysis & Comparison. In: *Journal Teknologi.* No. 48, June, pp. 113-141.

Skogestad, S., Postlethwaite, I. (2005). Multivariable feedback control: analysis and design (second edition). John Wiley & Sons, Ltd, 574 pp., ISBN 13 978-0-470-01167-6 (HB) 978-0-470-01168-3 (PBK).

Veselý V. and Harsányi L. (2006). *Robustné riadenie dynamických systémov*. Slovenská Technická Univerzita v Bratislave, 126 pp., 978-80-227-2801-0.

ROBUST TUNING OF PI CONTROLLER FOR IPDT PLANT

M. Huba [*,**]

[*] *STU FEI Bratislava, Slovakia*
(Tel: +421265429521; e-mail: mikulas.huba@stuba.sk)
[**] *FernUniversität in Hagen, FMI, PRT*
(Tel: + 49 2331 9871105; e-mail: mikulas.huba@fernuni-hagen.de)

Abstract: By considering robust tuning of the PI controller for uncertain Integral Plus Dead Time plant (IPDT) this paper demonstrates possibilities of the new Matlab/Simulink tool based on the performance portrait method. For plants with parameters defined over uncertainty intervals it enables to guarantee transient responses with specified deviations from ideal shapes at the plant output and input and to fulfill additional optimality specification, defined e.g. in terms of the minimal IAE values for the setpoint and disturbance steps, in terms of the maximal integral gain, etc. In difference to the robust tuning methods of the 1st generation considering typically controller parameters calculated from plant parameters specified by a single entry, in this new method uncertain plant parameters are specified by two entries characterizing their extreme values. As the ideal step responses at the plant output monotonic transients are considered, whereas at the plant input one-pulse step responses consisting of two monotonic intervals are required.

Keywords: Proportional control, optimal control, robust control, dead time.

1. INTRODUCTION

Tuning of the PI controller for the IPDT plant

$$F(s) = \frac{K_s}{s} e^{-T_d s}$$
$$K_s \in \langle K_{smin}, K_{smax} \rangle ; T_d \in \langle T_{dmin}, T_{dmax} \rangle \tag{1}$$

is frequently treated both in the process control and in the motion control areas. In connection with appropriate model reduction techniques it enables to approximate broad range of processes Åström and Hägglund (2005), Skogestad (2003). Consequently, high number of different "optimal" tuning rules based on this model may be found in the literature O'Dwyer (2006).

From the early beginning of PID control, for the controller tuning both the analytical (see e.g. Oldenbourg and Sartorius (1944,1951)) as well as experimental methods Ziegler and Nichols (1942) were used.

When considering tuning rules appropriate for education & practice, it is to agree with Skogestad (2003) that they should be 1. well motivated, 2. preferably model-based, 3. analytically derived, 4. simple and easy to memorize and 5. work well on a wide range of processes. When continuing with requirements of Skogestad (2006), controller tuning should enable achieving trade-off between: fast speed of response, good disturbance rejection, stability & robustness, less input usage and less sensitivity to measurement noise.

But, on the other hand, also the experimental controller tuning played always an important role, what may e.g. be demonstrated by the high popularity of the early tuning by Ziegler and Nichols (1942) that still gives inspiration for many new approaches based on the accumulated knowledge and broad simulation possibilities Åström and Hägglund (2004), Hägglund and Aström (2002). Of course, except of the analytical design the main requirements on such tuning remain mostly the same as above.

In this paper we are going to show that the requirements of robust control may be combined with high requirements on control performance, when the proposed tuning will guarantee specified performance not only for the nominal working point, but for any loop parameters of the uncertainty intervals in (1). Similarly as Åström and Hägglund (2004),Hägglund and Aström (2002), or Ziegler and Nichols (1942) the new method is based on carrying out series of simulation experiments on some sample of representative processes under requirement of chosen shape-related performance measures. Such an approach can today be easily performed by using tremendous power of computers for organizing and evaluating experiments, as well as for processing, visualizing, storing and recalling the achieved results for large number of control loops typical in practice. Thereby, one can easily extend spectrum of different qualitative & quantitative properties that will be evaluated and stored in computer database, to be chosen "on demand" and in different combinations by engineer carrying out design requiring particular specifications.

The paper is structured as follows. To characterize basic properties of the first generation of robust controller tuning methods and to enable their systematic comparison with the new proposed method, in Chapter 2 several tuning methods are discussed. In Chapter 3, basic requirements on robust controller tuning are summarized and in Chapter 4 performance measures for robust controller tuning in the time domain are introduced. In Chapter 5 the performance portrait for plant (1) is described and then used in Chapter 6 for controller tuning based on minimization of IAE values of setpoint step responses, or maximization of the integral gain values subject to shape related constraints for the plant input and output. The achieved results are compared with those corresponding to the first-generation robust tuning methods. Basic conclusions are summarized in Chapter 7.

2. FIRST GENERATION OF ROBUST CONTROLLER TUNING METHODS

Next we will briefly introduce several robust tuning methods that may be used for the IPDT plant and are interesting with respect to the paper aims.

2.1 Analytical controller design - TRDP

Based on generalization of the double real dominant closed loop pole Oldenbourg and Sartorius (1944,1951) to the triple real dominant pole (TRDP), whereby the PI controller is extended by the setpoint weighting according to

$$U(s) = K_c\left[bW(s) - Y(s)\right] + \frac{K_c}{sT_i}\left[W(s) - Y(s)\right] \tag{2}$$

interesting nominal tuning was analytically derived both for regulatory as well as tracking control tasks in Víte̒čková and Víteček (2008a), Vítečková and Víteček (2008b). The setpoint weighting can be shown to be equivalent to using prefilter

$$F_p(s) = \frac{bT_is + 1}{T_is + 1} \tag{3}$$

with T_i being the integral time constant. The approach is based on solving closed loop characteristic equation for a triple pole s_0 that for

$$\begin{aligned} A(s) &= s^2T_ie^{T_ds} + K_rK_s(T_is + 1) \\ \dot{A}(s) &= 2sT_ie^{T_ds} + s^2T_dT_ie^{T_ds} + K_rK_sT_i \\ \ddot{A}(s) &= 2T_ie^{T_ds} + 4sT_dT_ie^{T_ds} + s^2T_d^2T_ie^{T_ds} \end{aligned} \tag{4}$$

requires to fulfill

$$A(s_0) = 0; \dot{A}(s_0) = 0; \ddot{A}(s_0) = 0 \tag{5}$$

Solution of the last equation in (5) yields root

$$s_0 = -(2 - \sqrt{2})/T_d \tag{6}$$

for which from the first two equations in (5) one gets stable tuning with parameters

$$\begin{aligned} K_c &= 2(\sqrt{2} - 1)e^{\sqrt{2}-2}/(K_sT_d) \approx 0.461/(K_sT_d) \\ T_i &= (2\sqrt{2} + 3)T_d \approx 5.828T_d \end{aligned} \tag{7}$$

For the root $s_0 = -(2 + \sqrt{2})/T_d$ the resulting values $K_c = -0.1588/(K_sT_d); T_i = 0.17157T_d$ with negative loop gain do not guarantee the closed loop stability.

Zero of the closed loop transfer function

$$F_{wy} = \frac{K_sK_c(T_is + 1)}{s^2e^{T_ds} + K_sK_c(T_is + 1)} \tag{8}$$

can be cancelled by the prefilter denominator in (3) that removes overshooting typical for one-degree-of-freedom PI controllers. Simultaneously, by cancelling one of the triple pole (6) by the prefilter numerator (3) that further accelerates the transient responses, one gets the setpoint weighting coefficient

$$b = \frac{1/\left|s_0\right|}{T_i} = \frac{2 - \sqrt{2}}{2} \approx 0.293 \tag{9}$$

The corresponding maximal sensitivity and the complementary sensitivity peaks are $M_s = 1.70; M_t = 1.44$.

Examples of achieved transients compared with other tuning approaches are given in Figs. 2-3, 6-7 and 8-9. Basic advantage of the nominal tuning is given by compactness and elegance of its derivation. Though the method gives fast and smooth responses both in regulatory as well as tracking control, its extension to uncertain plants (1) and balancing different requirements on the setpoint and the disturbance response transient shapes at the plant input and output make already problems - the method does not include free tuning parameter enabling dynamics modifications.

2.2 SIMC PI Controller

As the 2nd example illustrating the analytical controller tuning we will mention the popular SIMC PI-rule (abbreviation from Simple/Skogestad Internal Model Control) for fast response with good robustness Skogestad (2003).

Firstly, by considering direct controller synthesis Rivera et al. (1986), Skogestad (2003) leading for a general first order plus dead time (FOPDT) plant

$$F_s = \frac{K_se^{-T_ds}}{s + 1/T_1} \tag{10}$$

to a simple first-order setpoint-to-output closed loop transfer function with time constant τ_c

$$F_{wy} = \frac{R(s)F(s)}{1 + R(s)F(s)} = \frac{R(s)K_s}{(s + 1/T_1)e^{T_ds} + R(s)K_s} \tag{11}$$

$$F_{wy} \overset{!}{=} \frac{1}{1 + \tau_cs}e^{-T_ds} \tag{12}$$

the PI controller

$$R(s) = \frac{s + 1/T_1}{K_s(\tau_c + T_d)s} \tag{13}$$

is derived, whereby the exponential term may be eliminated by using its first-order Taylor series approximation

$$e^{-T_ds} \approx 1 - T_ds \tag{14}$$

what requires to use $\tau_c \geq T_d$. For stable 1st order systems it is usually chosen $T_i = T_1$ and $K_c = 1/(K_sT_1(\tau_c + T_d))$. However, for integral systems, when $T_1 \to \infty$, solution (13) is actually approaching the proportional controller, what leads to poor rejection of input (load) disturbances. Of course, it is still possible to choose PI controller and to look for its appropriate tuning by other means, but it is no more the above mentioned direct controller synthesis of the IMC control. Therefore, the first question arises if the abbreviation SIMC is still appropriate for integral plants. In Skogestad (2003) tuning for such systems is derived by analyzing conditions of the critically damped closed loop system with the PI controller and integral delay-free plant ($T_d = 0$), when the double real dominant pole may be achieved by choosing

$$T_i = 4/(K_s K_c) \tag{15}$$

Finally, to consider dead time, the closed loop time constant in (12) was chosen as $\tau_c = T_d$ what yields

$$K_c = 1/(2K_s T_d); T_i = 8T_d \tag{16}$$

Such tuning that might be considered as simplification of the above method (double real dominant pole instead of the triple one) is not only simple, easy to remember but for the lag dominant plants it brings a reasonable improvement of the input-disturbance dynamics in comparing with the traditional IMC tuning rules and also with other tested methods (see Figs. 2-3, 6-7 and 8-9). It yields a reasonably fast response with moderate input usage and good robustness margins both in regulatory as well as tracking control. The analytical controller derivation is no more as compact as in the above case and as it was already mentioned above, it is no more the IMC control. The PI controller was not analytically derived, but chosen. Tuning of the integral part was made for delay-free system what leads to a question, in which range of the dead-time values it will keep the expected performance. But, on the other hand, together with the "half-rule" enabling to deal effectively with more complex plants it shows on necessity to link the controller design to approximative loop modelling and detecting its weakest points by the possibly simplest means.

When comparing integral loops with controller (16) with the IMC control of stable plants, it is also to note that for the integral plant the output setpoint step responses typically have overshooting, whereas in controlling stable 1st order plants (10) the closed loop step responses (12) are monotonic both at the plant input and output. When aiming to monotonic setpoint step responses at the plant output also in controlling integral plants, it is again possible to introduce setpoint weighting (2-3). However, since the method does not give information about the dominant closed loop poles, the calculation based on cancelling one real closed loop pole requiring to choose

$$b = 1/\left|s_1\right| T_i = 0.702 \tag{17}$$

does no more guarantee purely monotonic output (due to the obviously complex remaining dominant closed loop pole). So, a setpoint weighting guaranteeing purely monotonic output can be determined just experimentally as

$$b = 0.592 \tag{18}$$

What is again to be stressed is that the tuning is typically done just in a nominal point. By using specifications in the frequency domain, it is indeed shown that for integrating processes the suggested settings (16) give the gain margin $GM = 2.96$, the phase margin $PM = 46.9°$, the maximal sensitivity and the complementary sensitivity peaks $M_s = 1.70; M_t = 1.30$, and the maximum allowed time delay error with respect to stability is $1.59T_d$, but the controller tuning does not directly depend on the extreme values of the plant parameters in (1). The method neither includes free parameter enabling to balance dynamics of the setpoint and disturbance responses.

2.3 *Non-Convex Optimization Based PI Control*

As the 3rd tuning approach to be compared with the newly developed tuning the numerical non-convex optimization method Åström et al. (1998) will be mentioned. Based on the frequency-domain loop specifications by the maximum and complementary sensitivity peaks $M_s = 1.40$ and $M_t = 1.45$ it gives

$$\begin{gathered} b = 0.66; K_c = 0.282/(K_s T_d) \\ K_i = \frac{K_c}{T_i} = \frac{0.0418}{K_s T_d^2} \Rightarrow T_i = 6.746 T_d \end{gathered} \tag{19}$$

The optimization problem used for derivation of above results was specified as follows: find controller parameters that maximize the integral gain $K_i = K_c/T_i$ subject to the constraints that the closed-loop system is stable, the Nyquist curve of the loop transfer function satisfies the encirclement condition and that it is outside a circle that has the M_s and M_t circles in its interiors. Although it might seem at the first glance that standard optimization routines yield sufficient tools to solve this problem numerically, it was shown that *"the optimization problem is nontrivial because the constraint, which is infinite dimensional, defines a set in parameter space which is not convex"* Åström et al. (1998) and as a result the found controller parameters do give IAE values (Figs. 2-3, 6-7 and 8-9) that are much larger than those corresponding to other tested approaches. In the nominal case they do not allow achieving monotonic output setpoint step response even when choosing $b = 0$. But, they give relatively good responses for the relatively large deviation from the nominal case. So, they give a nice illustration of the fact that despite apparent simplicity the dynamics of the PID control is still tricky enough to be solved by standard, as well as specialized optimization routines. The dynamics specification in the frequency domain that is usually sufficient in dealing with robust stability problem seems not to be the best alternative for characterizing higher performance requirements in terms of the deviations from the shape related properties defined through the time domain responses.

2.4 *AMIGOs tuning for PI Controller*

Obviously being aware of too conservative tuning (19), in Hägglund and Åström (2002) new tuning rules were published based on Approximative M_s-constrained Integral Gain Optimization (AMIGO). These results corresponding to the maximum and complementary sensitivity peaks $M_s = 1.48$ and $M_t = 1.39$ extended by the choice b=0 to achieve monotonic step responseswill be used, when

$$b = 0; K_c = 0.35/(K_s T_d); T_i = 7T_d \tag{20}$$

The corresponding transient responses for setpoint and disturbance steps are in Figs. 2-3, 6-7 and 8-9. The nominal properties are slightly improved and by choosing b=0 the setpoint step responses are nearly monotonic at the plant output.

3. NEEDS FOR ROBUST CONTROLLER DESIGN

The most important feature of all above mentioned design methods is that they give robust tuning based on a single nominal point. The fact that real plants have just exceptionally properties characterized by fixed completely known point, is considered just indirectly, by choosing controller tuning that is sufficiently conservative to be usable also in the case of possible plant-model mismatch. So, possible uncertainty due to finite measurement precision or due to nonlinear character of real processes is paid by conservativeness of the tuning. All methods for controller tuning based on single set of parameters of the nominal plant model must be sufficiently robust against plant model uncertainties to be usable in practice. But, with exception of possible parameter changes allowed with respect to the robust stability, all the up to now mentioned methods do not directly give information specifying, how far the model parameters may deviate from the nominal point to keep the specified plant dynamics. They are just working with a conservativeness degree chosen equally for all possible applications. Some flexibility of the non-convex optimization in Åström et al. (1998) allowed by choice of the maximal sensitivity M_s is far from the originally proclaimed aims *"... to have a design parameter to change the properties of the closed-loop system. Ideally, the parameter should be directly related to the performance of the system, it should not be process oriented. There should be good default values so a user is not forced to select some value... The design parameter should also have a good physical interpretation and natural limits to simplify its adjustment."* All above mentioned methods are working with M_s values from a relatively narrow range 1.4-1.7, but despite to this their robustness and performance reasonably differ.

4. PERFORMANCE MEASURES FOR ROBUST CONTROL

Next, we are going to look for more appropriate tuning parameter(s) and method enabling to fulfil aims of robust control without leading to unnecessarily conservative tuning. From the performance point of view, at the plant output the expected dynamics is frequently specified by the setpoint step responses yielding monotonic transients.

The ideal continuous signal at the plant input giving after integration by the plant dynamics monotonic output will be denoted here as the one-pulse control. It may be characterized as a pulse having one extreme point that is dividing the overall transient into two monotonic control intervals.

Both such shape-related properties were, however, just rarely in focus of contemporary control research. Monotonic control together with a performance index for its evaluation was e.g. mentioned in Åström and Hägglund (2004), Hägglund and Aström (2002). One of recent reviews on PID control Keel et al. (2008) is mentioning just output non-overshooting control, without discussing possible specifications at the plant input and output that may be much harder. This is consequence of the development of last decades, when methods applied were dominated by the mathematical convenience and concentrated mostly on traditional performance criteria like gain margin, phase margin, maximum sensitivity, H_∞ norm, ISE, etc. Because of lacking analytical tools, the controller will be robustly tuned by using numerically derived areas of parameters corresponding to the above mentioned shape-related properties. The aim is to expand such nice dynamics of the nominal case as e.g. given by the tuning (7-9) that simultaneously fulfillis requirements on ideal shapes both at the plant input and output, but corresponding just to a single point with exactly known plant parameters to plant parameters known over uncertainty intervals Huba (2009), Huba et al. (2009), Huba (2010).

4.1 Ideally nonovershooting, monotonic and one-pulse responses

By its nature, definitions of the one-pulse control may be based on definition of the monotonic output control. This represents subset of non-overshooting control that represents subset of stable control.

The output transients $y(t)$ with $y(0) = 0$ corresponding to the setpoint step, $w = const \neq 0$ are classified according to validity of

$$y(t)/w \leq 1, \forall t \in (0, t_{sim}) \tag{21}$$

as *non-overshooting control.*

When fulfilling relations

$$0 \leq y(t_1)/w \leq y(t_2)/w \leq 1; \forall 0 \leq t_1 < t_2 \leq t_{sim} \tag{22}$$

the output response may be denoted as the *monotonic control* and in the case of the output fulfilling (22) and the input fulfilling

$$\begin{array}{r} sign(\dot{u}(t_1))sign(u(t_m)) \geq 0, \forall t_1 \in \langle 0, t_m \rangle \cup \\ \cup sign(\dot{u}(t_2))sign(u(t_m)) \leq 0, \forall t_2 \in \langle t_m, t_{sim} \rangle \end{array} \tag{23}$$

the dynamics may be denoted as *one-pulse control.* For all that $u(t_m); t_m \geq 0$ corresponds to the maximal control signal amplitude during transient and t_{sim} represents simulation time that should be larger than maximal possible settling time.

Since the settling time used for characterizing speed of output transient strongly depends on the defined measurement precision (given e.g. by ϵ), the much less dependent IAE (Integral of Absolute Error) defined as

$$IAE = \int_0^\infty |e(t)|\, dt \tag{24}$$

will be used as the time-related performance index for quantitative evaluation of responses.

4.2 Amplitude deviations from ideal shapes

In practice, but also in case of computer simulation, it has sense to weaken the above conditions on non-overshooting control by introducing some tolerable overshooting defined by new small positive constant

$$\epsilon > 0 \tag{25}$$

and to find in this way controller parameters corresponding to

$$y(t)/w \leq 1 + \epsilon, \forall t_1 \in (0, t_{sim}) \tag{26}$$

E.g. by choosing $\epsilon = 0.01$, the setpoint step responses with overshooting up to 1% of the setpoint value w will be tolerated and included under denotation as the non-overshooting control. In this paper this approach will only be used for $\epsilon \leq 0.1$, because responses with larger overshooting may also be achieved in other ways (e.g. without using setpoint weighting) and so the design should consider also other alternatives.

A continuous nearly monotonic signal $y(t)$ with the initial value $y_0 = y(0)$ and with the final value $y_\infty = y(\infty)$ will be denoted as ϵ_y -monotonic when it fulfills condition

$$\begin{array}{l}[y(t) - y(y-T)]\, sign(y_\infty - y_0) \geq -\epsilon_y \\ T \leq t < \infty, T \in (0, T_{max}), \forall T_{max} > 0\end{array} \tag{27}$$

Thereby, in order not to prolong the time required for testing with any positive T_{max}, this has to be chosen to enable capturing sufficient part (e.g. half-period) of the superimposed signal. Number of samples that need to be tested Huba (2010) may be decreased, if all subsequent local extreme points fulfill condition

$$[y_{le,i+1} - y_{le,i}]\, sign(y_\infty - y_0) \geq -\epsilon_y; i = 1, 2, 3, \ldots \tag{28}$$

The amplitude deviations from one-pulse control (23) are based on evaluating amplitude deviations from monotonicity over both monotonic intervals before and after the dominant extreme point $u(t_m); t_m \geq 0$.

Non-overshooting specifications (not distinguishing between non-overshooting and monotonic control) exist also in the frequency domain (see e.g. Keel et al., 2008) but their application is extremely complicated, especially when speaking about dead time systems.

4.3 Integral deviations from ideal shapes

Specific integral measure for deviations from monotonicity was introduced by Åström and Hägglund (2004), Hägglund and Aström (2002). Here, we will prefer new measures for deviations from monotonic and one-pulse shapes that may be easily tested numerically, by evaluating simulated or experimentally measured transients corresponding to the setpoint and disturbance step responses and are also appropriate for constrained control.

To evaluate control effort required to achieve the required output behavior, Total Variance (TV) criterion was proposed Skogestad (2003), Skogestad and Postlethwaite (2007) defined as

$$TV = \int_0^\infty \left|\frac{du}{dt}\right| dt \approx \sum_i |u_{i+1} - u_i| \tag{29}$$

Under non-perfect control it is not easy to be evaluated analytically. So, typically, its values are computed by simulation after appropriate discretization with sampling period as small as possible. According to Skogestad and Postlethwaite (2007) in Matlab it may be simple computed by the command `sum(abs(diff(u))`.

Very simple integral measure for evaluating deviations from strict monotonicity defined for the plant output $y(t)$ with the initial value $y(0)$ and the final value $y(\infty)$ by modification of the TV criterion will be denoted here as the TV_0 criterion Huba (2010)

$$TV_0 = \sum_i |y_{i+1} - y_i| - |y(\infty) - y(0)| \tag{30}$$

$TV_0 = 0$ just for strictly monotonic response, else $TV_0 > 0$.

In controlling unstable and integral plants the number of significant control pulses cannot decrease below the number of unstable poles Huba (2009), Huba (2010). To stress contribution of the superimposed oscillation in systems with 1P dominant control it is then appropriate to work with the TV_1 criterion defined as

$$TV_1 = \sum_i |u_{i+1} - u_i| - |2u_m - u(\infty) - u(0))| \geq 0 \tag{31}$$

This gives zero values just for strictly 1P control signal and may be applied also to constrained control signal. For control signals with superimposed higher harmonics it takes positive values.

Graphically represented in the plane of loop parameters, together with quantitative measures, such properties will be giving *performance portrait* of particular control loop. In this way, the new approach continues in developing trends recommended e.g. by Ackermann (2002)

5. CLOSED LOOP PERFORMANCE PORTRAIT (PP) AND ROBUST CONTROLLER DESIGN

The closed loop PP represents information about the loop performance corresponding to the setpoint and the disturbance step responses expressed over a grid of (possibly normalized) loop parameters including all possible working points. By containing information about required loop properties for different loop paramterers, the PP may be used both for optimally choosing the nominal controller tuning for a comletely known plant, or for the robust controller tuning of a plant with interval parameters.

For a loop represented by a parameter vector

$$P = \{p_1, p_2, \ldots p_S, p_{S+1}, p_{S+I}\} \tag{32}$$

with the dimension

$$D = S + I \tag{33}$$

each entry of the first subset of parameters $p_i; i = 1, \ldots, S$ is given as a signle value that has to be fixed during the controller tuning.

There may also exist some uncertain (plant) parameters

$$p_i \in \langle p_{imin}, p_{imax}\rangle\, ; \qquad i = S+1, \ldots, S+I \tag{34}$$

that vary over some (known) intervals. Next, we define such limits also for the first subset of parameters (e.g. by some preliminary robust stability analysis method), so that all parameters may be expressed in the above form.

In computation of the PP all parameters p_i take just discrete $n_i + 1$ levels

$$\begin{array}{l}p_{i,j} = p_{imin} + (p_{imax} - p_{imin}) j / n_i; \\ j = 1, 2, \ldots, n_i; n_i > 1; i = S+1, \ldots, S+I\end{array} \tag{35}$$

Both the nominal as well as the robust control design may now be carried out in two ways: as determination of an optimal controller parameter set, or as a determination of an optimal working point of a controller expressed by means of the plant parameters vectors. When the number of the controller parameters exceeds number of the plant parameters, combination of both approaches is possible.

When e.g. working with the uncertain plant model (1), the controller (2- 3) is specified by three parameters b, K_c, T_i. In addition to the plant parameters K_s and T_d, specification of the setpoint response (12, 13) additionally requires determination of at least one time constant τ_c. It means that in total there are 6 parameters that determine the resulting dynamics. If two of them, $K_s \in (K_{smin}, K_{smax}); T_d \in (T_{dmin}, T_{dmax})$ are uncertain, the task of the control design may be formulated as:

a) to find directly the controller parameters b, K_c, T_i, or

b) for the controller parameter defined by formulas introduced ich Chapter 2 to find an appropriate location of the operating point K_{s0}, T_{d0} and the free design parameters b.

Both has to be done in such a way that over all grid points corresponding to chosen tuning and to all possible values of the uncertain interval parameters the required shape-related performance measures will be achieved. The necessary amout of computation and the achieved precision will obviously depend on the level of quantization and on the choice of the limits introduced for the free parameters that have to be determined.

PP required for such a design may be generated by simulation, or by real time experiments. When it is based on normalized parameters, it may then be repeatadly used for different tasks with different values of particular loop parameters. Although such PP generation may be connected with numerical problems, especially those related to the nature of grid computations, when one has to balance precision of achieved results (quantization level in considered grid) with the total number of evaluated points and the corresponding computation time, it gives very promising results especially when dealing with dead time systems.

The first attempt to analyze optimal robust tuning of the IPDT plant by the performance method was done by Huba et al. (2009) in 2D space of normalized parameters $K = K_cK_sT_d$ and $\Omega_f = T_d/T_i$. The setpoint weighting b has been chosen as minimum of the optimal nominal values calculated from the two above parameters over the uncertainty set, what was leading to slightly conservative tuning. Now, the setpoint weighting will be considered as the third independent coordinate and the performance portrait will be generated over grid of points in 3D with the uncertainty subspace given by the coordinates vector $K = K_cK_sT_d$; $\tau_i = T_i/T_d$; and parameter b representing the third coordinate. The integer variable describing particular levels of this parameter will be displayed in the following figures as k.

6. COMPARATIVE ANALYSIS OF PI TUNINGS

6.1 Nominal Tuning for min IAE

The simplest strategy for designing robust controller tuning seems to be to find such controller parameters b, K_c, T_i that will guarantee for all possible plant parameters (1) minimal mean IAE values subjected to amplitude or integral deviations on the plant input and output.

Fig. 1 shows several windows of one layer (with $k = 16$) of the 3D performance portrait calculated for the setpoint step responses over 27x27x21 points for $K \in \langle 0.1, 1.4\rangle$; $\tau_i \in \langle 3.5, 15.5\rangle$; $b \in \langle 0, 1\rangle$. The position of the optimal operating point gives minimal IAE value for the tolerated output amplitude deviation from monotonicity and input amplitude deviation from one-pulse control $\epsilon_y = \epsilon_u = 10^{-3}$.

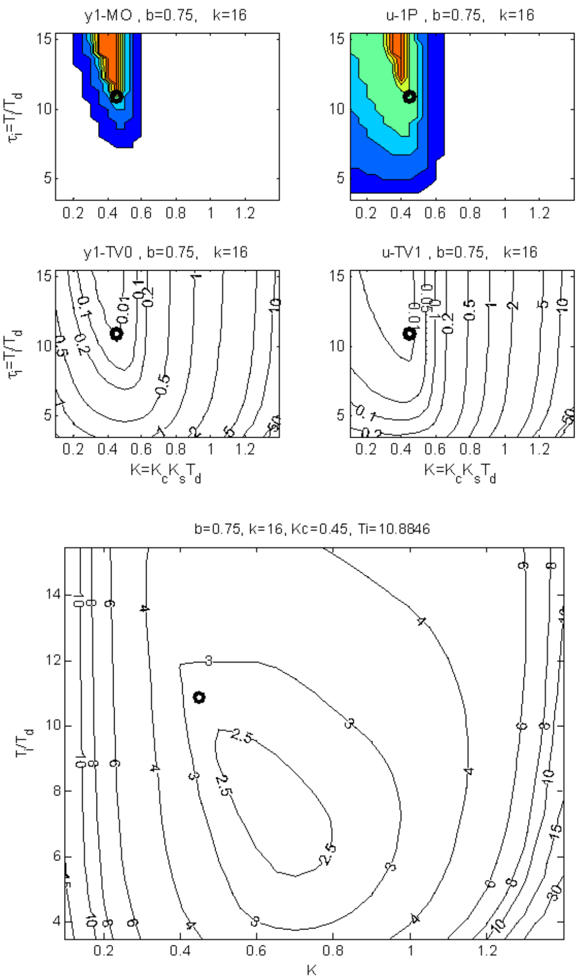


Fig. 1. One layer of the PP ($k = 16$) calculated for the setpoint step responses over 27x27x21 points and containing the optimal nominal tuning corresponding to min IAE. Note the similarities between the amplitude and integral measures for the plant output and input

By comparing the amplitude and integral deviations from ideal shapes it is possible to conclude that usually it would be enough to work with one set of such measures, whereby, due to their simplicity, the integral measures could be preferred and the amplitude deviations could be estimated as

$$\epsilon_y \le TV_0(y)/2; \epsilon_u \le TV_1(u)/2 \tag{36}$$

The identity holds just then when the analyzed transition has exactly one additional pulse with the amplitude given by the particular value of ϵ.

The setpoint and disturbance responses in Fig. 2 and Fig. 3 corresponding to the found optimal parameters

$$K_c = 0.45; T_i = 10.88; b = 0.75 \tag{37}$$

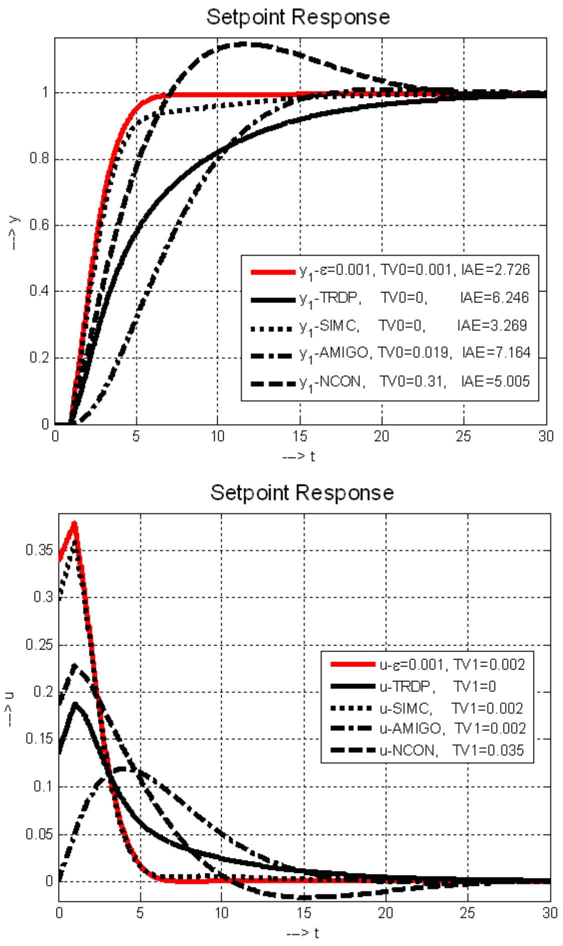


Fig. 2. Setpoint step responses at the plant output and input corresponding to the optimal tuning according to Fig. 1 (red) compared with TRDP (7-9), SIMC (16) and non-convex optimization (19).

do not represent an absolute optimum. By broadening the PP to larger integral time constants and by increasing number of grid points (decreasing the quantization step), the identified optimal solutions tend to those corresponding to pure P control and $T_i \to \infty$. This trivial handicap (with respect to the disturbance response) can be avoided by optimizing weighted sum of the setpoint and disturbance responses. But already without such modification, the achieved results show that the new method enables to optimize the setpoint responses by keeping acceptable disturbance response.

6.2 Nominal Tuning for max K_i

Next we are going to compare the new Performance Portrait method with the optimization based approaches. By their numerical procedures both approaches are very close each other. Similarly as in the PI controller tuning by the non-convex optimization Åström et al. (1998), or by its later modification Hägglund and Aström (2002), also the PP method will be used to find the maximal integral gain, but instead of the previously considered constraints on the maximum sensitivity, the search will now be subject to the shape related constraints puts on the plant input and output step responses.

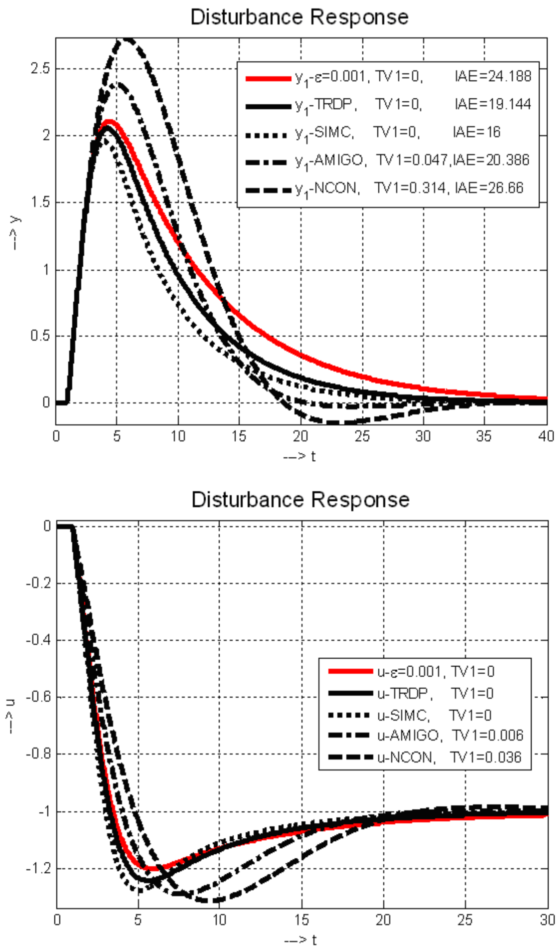


Fig. 3. Load-disturbance step responses at the plant output and input corresponding to the optimal tuning according to Fig. 1 (red) compared with TRDP (7-9), SIMC (16) and non-convex optimization (19).

Fig. 4 shows several windows of one layer of the 3D performance portrait from the above example corresponding to the location of the optimal nominal point yielding

$$K_c = 0.65; T_i = 4.42; b = 0 \qquad (38)$$

The corresponding setpoint and disturbance responses in Fig. 6 and Fig. 7 show that this approach does not give the absolutely best setpoint response (this was not required), but the achieved disturbance response is already the absolutely best one. Again, the look up of the optimal tuning was fully based on the performance portrait corresponding just to the setpoint step response. Although the TV_1 values of the disturbance response are relatively close to the absolute minimum, when necessary, this parameter may be further improved by considering also the PP of the disturbance response.

Transients in Fig. 6 and Fig. 7 show that the best nominal setpoint step responses are achieved by the SIMC tuning that also gives relatively good disturbance responses.

Surprisingly, the disturbance responses corresponding to the non-convex optimization NCON Åström et al. (1998), or the AMIGOs tuning Hägglund and Aström (2002) derived by optimization for the optimal disturbance response (maximal K_i gain) give the worst IAE results.

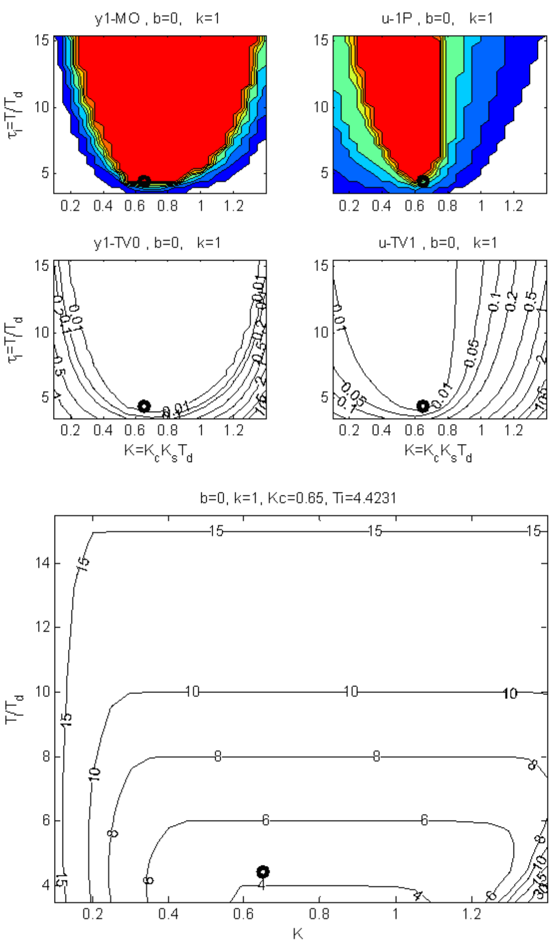


Fig. 4. The first layer ($k = 1$) of the PP calculated for the setpoint step responses over 27x27x21 points for $K \in \langle 0.1, 1.4\rangle$; $\tau_i \in \langle 3.5, 15.5\rangle$; $b \in \langle 0, 1\rangle$ and indicating the optimal nominal tuning corresponding to max K_i corresponding to the tolerated output amplitude deviation from monotonicity and input amplitude deviation from one-pulse control $\epsilon_y = \epsilon_u = 10^{-3}$. Note the similarities and differences between the amplitude and integral measures for the plant output and input.

6.3 Robust Tuning for max K_i

The seemingly bad results of the nominal tuning based on the nonconvex optimization subject to sensitivity constraints may be explained by considering interval plant parameters. Consider e.g. plant with the dead time uncertainty

$$T_{dmin} = 0.3; T_{dmax} = 1.0 \qquad (39)$$

and the corresponding robust controller tuning. Since the uncertain parameter T_d is included both in the PP parameter $K = K_cK_sT_d$, as well as in $\tau_i = T_i/T_d$, all possible operating points given by the optimal controller tuning

$$K_c = 26; T_i = 1; b = 0 \qquad (40)$$

sweep in the corresponding layer of PP in Fig. 5 parabolic curve segment. All its points need to satisfy the above given tolerances on the deviations from the output monotonicity and input one-pulse response ($\epsilon_y = \epsilon_u = 10^{-3}$).

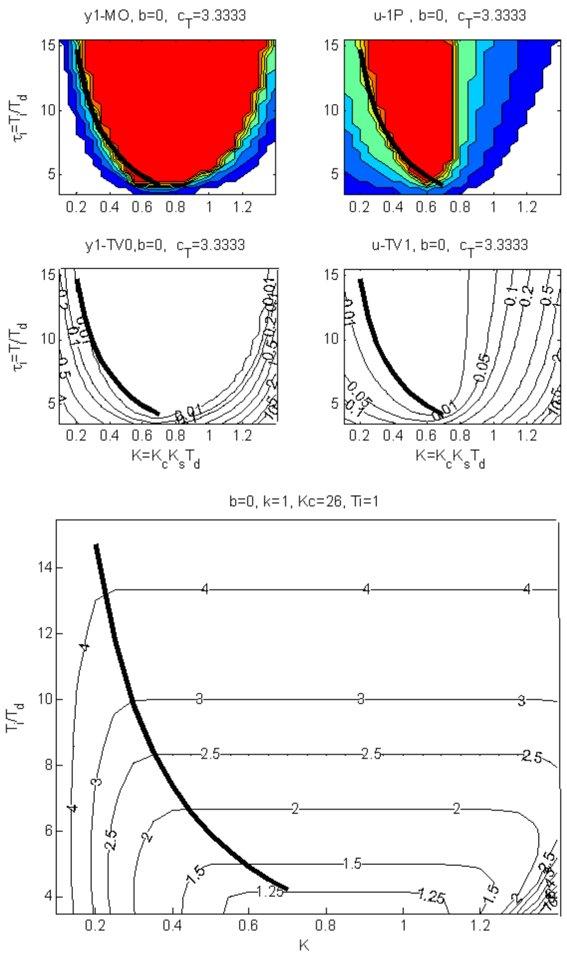


Fig. 5. The first layer ($k = 1$) of the PP calculated for the setpoint step responses over 27x27x21 points for $K \in \langle 0.1, 1.4\rangle$; $\tau_i \in \langle 3.5, 15.5\rangle$; $b \in \langle 0, 1\rangle$ and displaying the amplitude and integral deviations from monotonicity at the plant output and from the 1P at the plant input (above) and from the output IAE values (below); The Uncertainty Curve Segment corresponding to $T_d \in \langle 0.3, 1.0\rangle$ and $\epsilon_y = \epsilon_u = 10^{-3}$ satisfies to the requirement $K_i = K_c/T_i \overset{!}{=} max$.

For both limit values of T_d show Fig. 8 and Fig. 9 that the new method gives the best disturbance responses by simultaneously keeping the shape related performance measures for the setpoint step responses.

All previously mentioned method were tuned around the symmetrically chosen nominal operating point

$$T_{d0} = (T_{dmin} + T_{dmax})/2 = 0.65 \qquad (41)$$

The TRDP method that seems to be slightly conservative in the nominal case gives now good performance over the whole considered uncertainty interval, just for larger difference between extreme dead time values it would already lead to oscillatory behavior.

SIMC method is the best one for $T_d = T_{dmin}$, but for $T_d = T_{dmax}$ it already leads to oscillatory behavior what could be at least partially compensated by non-symmetrical choice of the nominal operating point. The PP method could be used to find new position of the operating point

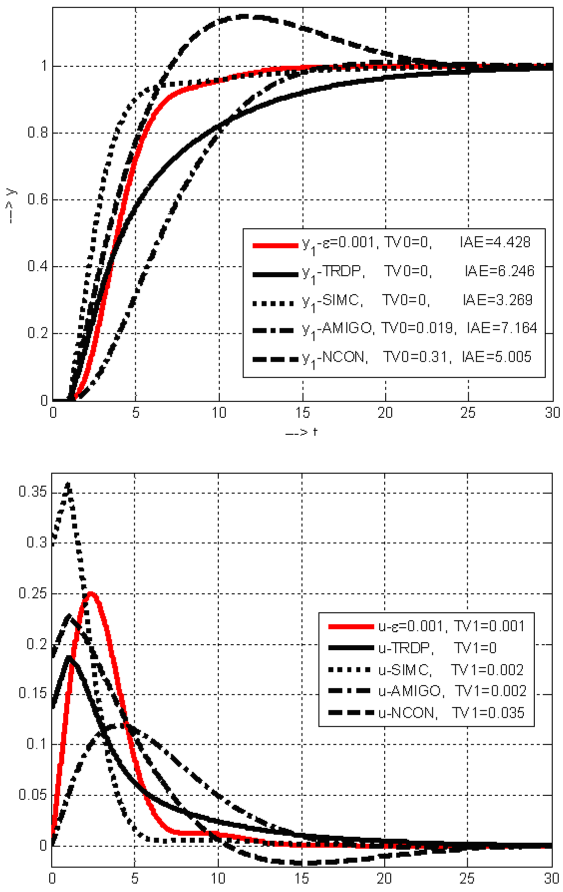


Fig. 6. Setpoint step responses at the plant output and input corresponding to the optimal tuning according to Fig. 4 (red) compared with TRDP (7- 9), SIMC (16) and non-convex optimization (19).

symmetrizing deviations corresponding to the limit dead time values.

The NONC and AMIGOs tuning lead for extreme dead time values to surprisingly better performance with lower deviations from ideal schapes than in the nominal case.

7. CONCLUSIONS

New control design method based on amplitude and integral deviations of the transient responses at the plant input and output from their ideal shapes was proposed and illustrated by the frequently treated task of the PI controller tuning in this paper.

The carried out comparative analysis including several first-generation robust tuning approaches for the IPDT uncertain plant has shown their typical features: in some context they may give excellent properties, just to know when, how and which controller tuning and the operating point have to be used. The new approach showed to be much more effective and efficient than the approaches based on the plant characteristics in the frequency domain in all analyzed situations. Whereas the traditional methods are not only typical by a preprogrammed degree of conservativeness and they also do not give information, how the operating point should be chosen with respect to uncertainty intervals of the considered uncertain parameters the new method directly gives solution optimally fitting the specified performance measures without any redundant conservativism for all possible operating points specified by the uncertainty intervals, or indicates that the specified performance may not be achieved by any tuning of the specified controller.

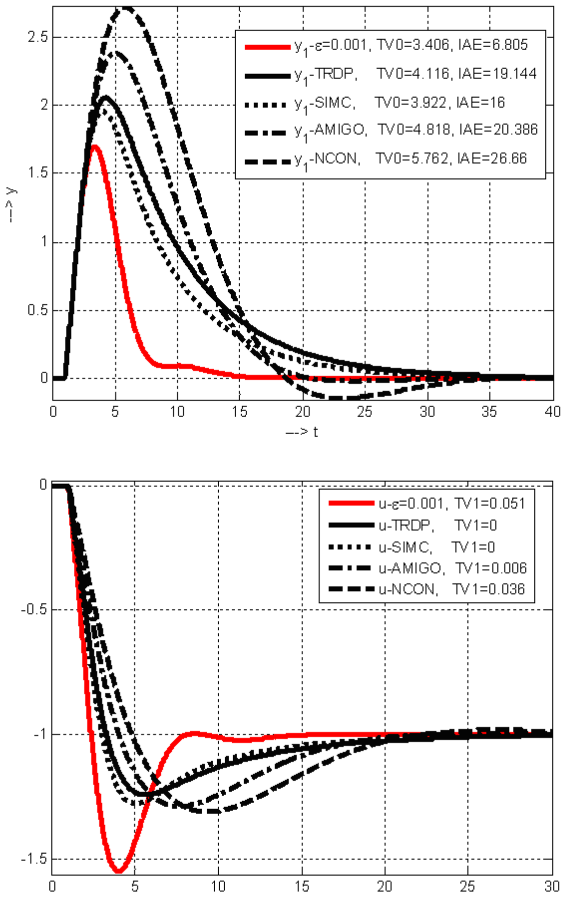
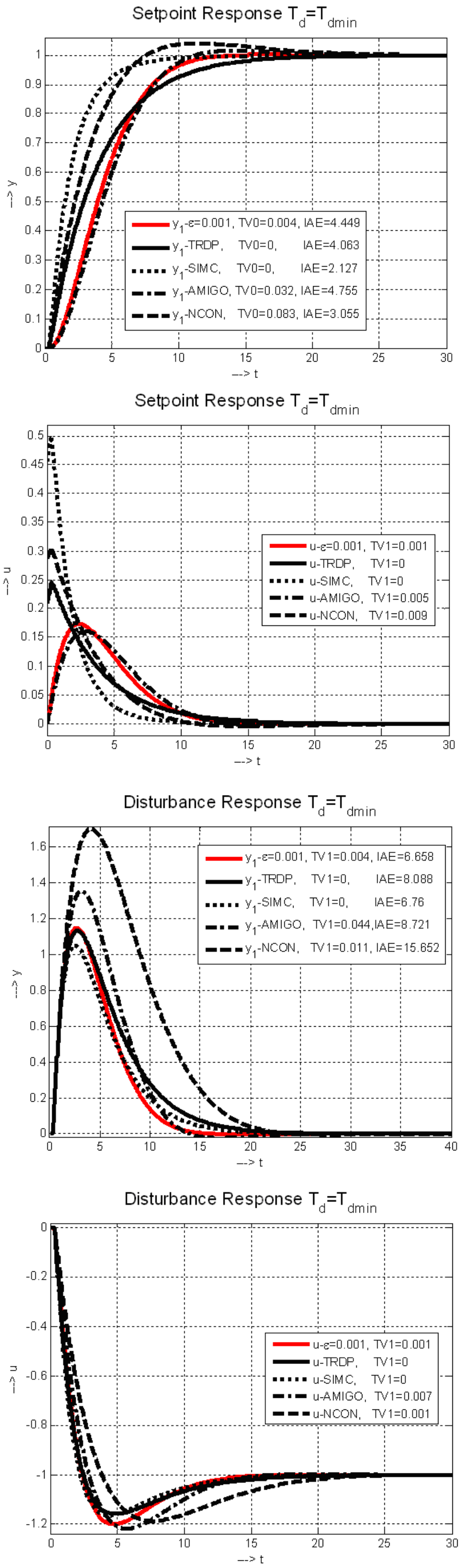


Fig. 7. Load-disturbance step responses at the plant output and input corresponding to the optimal tuning according to Fig. 2 (red) compared with TRDP (7- 9), SIMC (16) and non-convex optimization (19).

The new method avoids the second step of the traditional approach, when, after deciding, how the controller parameters should be expressed by means of the plant parameters, for plants with parameters taking values from an uncertainty interval it is not clear, how to choose the operating point in order to get the results that are the optimal for all possible values of the uncertain parameter.

In the comparative analysis it was shown that with respect to symmetry of the deviations from ideal shapes at the plant input and output an intuitive assignment of the operating point of a particular parameter to the centre of its uncertainty interval may not give satisfactory results. It is e.g. important for the SIMC tuning that gives excellent responses around the nominal operating point, but it is non-symmetrically sensitive to the dead-time uncertainty,



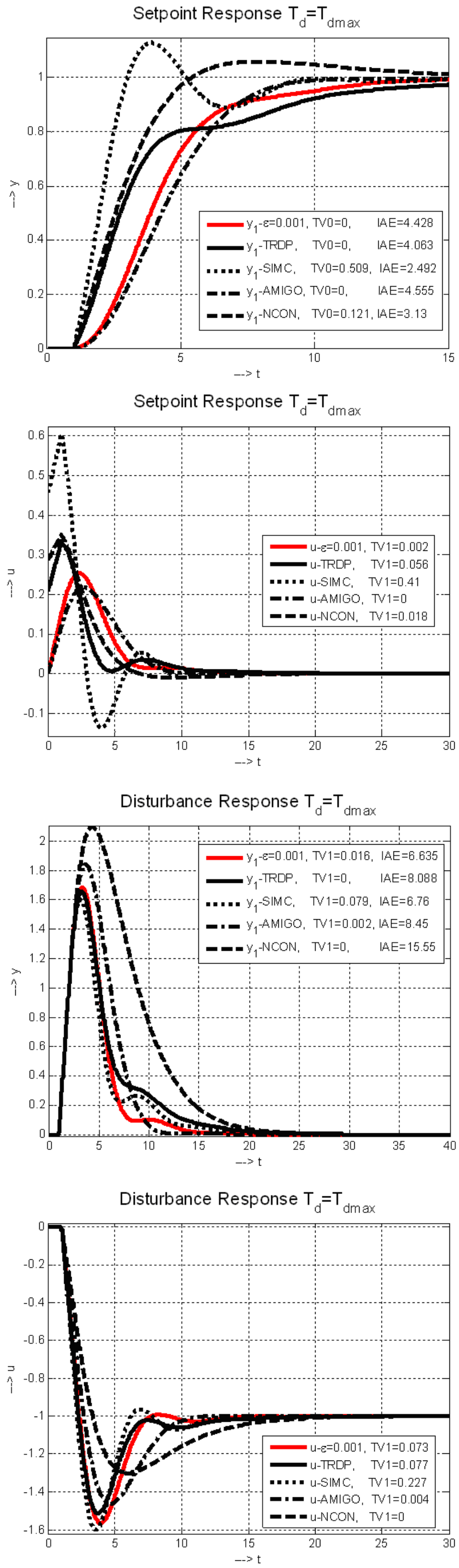


Fig. 8. Setpoint and disturbance steps for $T_d = T_{dmin}$; $\epsilon_y = \epsilon_u = 0.02$.

Fig. 9. Setpoint and disturbance steps for $T_d = T_{dmax}$; $\epsilon_y = \epsilon_u = 0.02$.

what consequently requires non-symmetrical choice of the nominal operating point over the uncertainty interval.

On the other hand, the TRDP method is able to guarantee output-monotonic and input-one-pulse transients for a broad neighborhood around the nominal working point, whereby by increasing deviation from the nominal point the conservativeness of the tuning decreases.

The robust tuning based on non-convex optimization (NONC) does not allow monotonic output step responses even in the nominal case, but the shape of responses is rather robust against dead-time perturbation and with increased deviation from the nominal case the performance improves what could explain motivation leading to this design. Its modified version AMIGOs removes the high overshooting of the nominal setpoint step responses and still gives relatively robust responses in the perturbed situations.

Analysis of the new approach to the robust PI controller tuning based on experimental identification of parameter areas corresponding to tolerable deviations from output-monotonic and input-one-pulse control clearly showed that the new method represents new generation of optimal tuning approaches that are able to guarantee believably chosen performance requirements for all considered loop parameters. So it is possible to avoid stiff character of the first-generation tuning formulas that may not only be too conservative, but also too sensitive in some applications.

Since the new method fully relies on a computer support, its use may be very simple and besides of the recommended tuning a lot of additional information characterizing the optimal solution and the overall context of the proposed tuning may be offered. Its drawback is that the designer is no more able to fully rely just on his "pen and paper", but the same happens in many other situations in our life.

For practical use, the above analysis of optimal controller tuning should yet be completed by analysis of the control constraint effects, since the parallel integral control is well known by the integrator windup that can fully destroy the control dynamics.

ACKNOWLEDGMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09, by the Grant VEGA No. 1/0656/09 and by the Grant VEGA No. 1/0369/10. It was also supported by a grant No. NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Ackermann, J. (2002). *Robust Control: The Parameter Space Approach - 2nd ed.* Springer, Berlin.

Åström, K.J. and Hägglund, T. (2004). Revisiting the ziegler-nichols step response method for pid control. *J. Process Control*, 14, 635–650.

Åström, K.J. and Hägglund, T. (2005). *Advanced PID Control.* ISA-The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.

Åström, K.J., Panagopoulos, H., and Hägglund, T. (1998). Design of pi controllers based on non-convex optimization. *Automatica*, 34, 585–601.

Hägglund, T. and Aström, K.J. (2002). Revisiting the ziegler-nichols tuning rules for pi control. *Asian J. Contr.*, 4:4, 364–380.

Huba, M. (2009). Robust design of integrating controllers for ipdt plant. In M. Fikar and M. Kvasnica (eds.), *Proc. 17th Int. Conf. Process Control '09, Š. Pleso, www.kirp.chtf.stuba.sk/pc09/data/papers/109.pdf*, 353–357.

Huba, M. (2010). Designing robust controller tuning for dead time systems. *IFAC Int. Conf. System Structure and Control, Ancona, Italy.*

Huba, M., Marko, Ľ., Bahnik, P., and Oravec, I. (2009). Numerical issues in designing pi controller for ipdt plant. www.kirp.chtf.stuba.sk/pc09/data/papers/110.pdf, 57–64.

Keel, L.H., Kim, Y.C., and Bhattacharyya, S.P. (2008). Advances in three term control. pre-congress tutorials & workshops. *17th IFAC World Congress Seoul, Korea.*

O'Dwyer, A. (2006). *Handbook of PI and PID controller tuning rules. 2nd Ed.* Springer, London.

Oldenbourg, R.C. and Sartorius, H. (1944,1951). *Dynamik selbsttï£¡tiger Regelungen.* R.Oldenbourg, München.

Rivera, D.E., Morari, M., and Skogestad, S. (1986). Internal model control. 4. pid controller design. *Ind Eng. Chem. Res.*, 25,1, 252–265.

Skogestad, S. (2003). Simple analytic rules for model reduction and pid controller tuning. *Journal of Process Control 13*, 291–309.

Skogestad, S. (2006). Tuning for smooth pid control with acceptable disturbance rejection. *Ind. Eng. Chem. Res.*, 45, 7817–7822.

Skogestad, S. and Postlethwaite, I. (2007). *Multivariable Feedback Control Analysis and Design.* John Wiley, N. York.

Vítečková, M. and Víteček, A. (2008a). Two-degree of freedom controller tuning for integral plus time delay plants. *Int. Conference Cybernetics and Informatics, February 10 - 14, 2008, ŽDIAR, Slovak Republic.*

Vítečková, M. and Víteček, A. (2008b). Two-degree of freedom controller tuning for integral plus time delay plants. *ICIC Express Letters. An International Journal of Research and Surveys, Japan*, Volume 2, Number 3, 255–229.

Ziegler, J.G. and Nichols, N.B. (1942). Optimum settings for automatic controllers. trans. *ASME*, 759–768.

ROBUST DECENTRALIZED CONTROLLER DESIGN WITH SPECIFIED PHASE MARGIN

Jakub Osuský and Vojtech Veselý

Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
Tel.: +421 2 60291111 Fax: +421 2 60291111
e-mail: jakub.osusky @ stuba.sk

Abstract: This paper presents the robust decentralized controller design in the frequency domain for stable plants. Robust condition based on M-delta structure is included in controller design. In controller design for MIMO systems equivalent subsystem method is used. For subsystems of equivalent model, frequency method ensuring desired phase margin is applied. Design procedure is illustrated on two tanks process.

1. INTRODUCTION

PID controllers are standard and well-proven solution for the majority of industrial applications. Over the years, a plenty of PID tuning rules were developed see e.g. (Šulc and Vítečková, 2005). In this paper decentralized PID controller design approaches are developed for stableand unstable systems and extended to satisfy robust stability conditions in terms of unstructured uncertainty developed in (Kozáková A., Veselý, 2005. Controllers for subsystems are designed for specified phase margin.

The paper is organized as follows: preliminaries and problem formulation are given in Section 2, robust stability conditions, ESM, and robust design procedure in Section 3. In Section 4 is a detailed robust decentralized PID controller design procedure illustrated on two tanks process. Conclusions are drawn at the end of the paper.

2. PRELIMINARIES AND PROBLEM FORMULATION

In frequency domain robust controller design very often consists of two steps: controller design and robust condition verification. If robust condition is not passed than controller is redesigned and condition is verified again, so these types of approaches are iterative.

Aim of this paper is to develop robust controller design method which will consist of only one step. So the robust stability condition will be included into the design procedure so that the designed controller will ensure robust stability without any iteration.

3. THEORETICAL RESULTS

3.1 Robust stability conditions

When designing a controller a major source of difficulty is plant model inaccuracy; hence uncertainty models are to be used which means that instead of a single model a class Π of perturbed models is to be considered. Denote $\tilde{G}(s) \in \Pi$ any perturbed plant model and $G(s) \in \Pi$ the nominal plant model. A simple uncertainty model is obtained using unstructured uncertainty $\Delta(s)$. Commonly used uncertainty forms are: additive (a) and input multiplicative (i) uncertainties

Standard feedback configuration with unstructured uncertainty of any type can be rearranged to obtain the general $M-\Delta$ structure in Fig. 1 where $M(s)$ represents the nominal model and $\Delta(s):\sigma_{\max}[\Delta(j\omega)] \le 1$ the normalized perturbation.

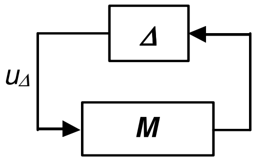


Fig. 1 – $M-\Delta$ structure

Robust stability condition for unstructured perturbations is formulated in terms of stability of the $M-\Delta$ system: if both the nominal system $M(s)$ is stable (nominal stability) and the normalized perturbation $\Delta(s)$ is stable, closed-loop stability is guaranteed for

$$\sigma_{\max}[M(j\omega)] < 1\ ,\quad \forall\omega \qquad (1)$$

For individual uncertainty types $M = l_k M_k$, $k = a, i$ in particular

- for additive uncertainty

$$M_a(s) = -[I + R(s)G(s)]^{-1}R(s)$$

$$l_a(\omega) = \max_{\tilde{G}\in\Pi}\sigma_{\max}[\tilde{G}(s) - G(s)] \qquad (2)$$

- for input multiplicative uncertainty

$$M_i(s) = -[I + R(s)G(s)]^{-1}R(s)G(s)$$

$$l_i(\omega) = \max_{\tilde{G}\in\Pi}\sigma_{\max}\{G^{-1}(s)[\tilde{G}(s) - G(s)]\} \qquad (3)$$

In view of (2), (3), condition (1) reads as follows

$$\sigma_{\max}(M_k(s)) < \frac{1}{|l_k(s)|},\ k = a, i \qquad (4)$$

In the robust stability condition for input multiplicative uncertainty $M_i(s)$ represent the complementary sensitivity function.

$$M_i(s) = -[I + R(s)G(s)]^{-1}R(s)G(s) = -T(s) \quad (5)$$

Denote right side of inequality (4) as $U(s)$.

$$U(s) = \frac{1}{|l_i(s)|} \qquad (6)$$

$U(s)$ does not depend on controller $R(s)$ so it can be calculated before controller design.

Similar it is for additive uncertainty where inequality (4) is rearrange into following form:

$$\sigma_{\max}(M_a(s)G(s)) < \frac{G(s)}{|l_a(s)|} \qquad (7)$$

Than

$$M_a(s)G(s) = T(s) \qquad (8)$$

And

$$U(s) = \frac{G(s)}{|l_a(s)|} \qquad (9)$$

If controller $R(s)$ is designed so that maximum value M_t of complementary sensitivity function $T(s)$ is smaller than minimal value of $U(s)$, system with designed controller satisfy robust stability condition (1).

This can be reached using any frequency controller design method ensuring desired phase margin (PM), for SISO systems because (Skogestad and Postletwaithe, 1997):

$$PM \geq 2\arcsin\left(\frac{1}{2M_t}\right) \qquad (10)$$

For MIMO systems due to interactions is ensuring of desired phase margin in subsystems more complicated.

Hence in this paper for MIMO systems equivalent subsystem method (Kozaková, et al., 2009) will be used for transforming nominal model $G(s)$ into diagonal model of equivalent subsystems $G^{eq}(s)$. For subsystems of equivalent model $G^{eq}(s)$, SISO method ensuring desired phase margin will be used.

3.2 Equivalent subsystem method

Equivalent subsystem method will be used to simplify the full nominal model matrix into diagonal equivalent one. Subsystems in equivalent matrix are called equivalent subsystems and are calculated with taking account into interactions.

In this paper only equations necessary for equivalent subsystem calculation will be written. More details about this method are in (Kozaková, et al., 2009).

Full matrix of nominal model $G(s)$ can be split into matrix containing diagonal elements $G_d(s)$ of $G(s)$ and $G_m(s)$ containing off-diagonal elements.

$$G^{eq}(s) = G_d(s) - P(s) = diag\{G_i(s) - p_i(s)\}_{i=1,2,...,m} =$$
$$= diag\{G_i^{eq}(s)\}_{i=1,2,...,m} \qquad (11)$$

G_i^{eq} is a diagonal matrix of equivalent subsystems. For individual subsystems, (11) yields

$$1 + R_i(s)G_i^{eq}(s) = 0 \qquad i = 1,2,...,m \qquad (12)$$

which are the m equivalent characteristic equations.

In the context of the independent design philosophy, the design parameters $p_i(s),\ i = 1,2,\dots,m$ represent the bounds for individual designs. To be able to provide closed-loop stability of the full system using a DC controller, $p_i(s),\ i = 1,2,\dots,m$ are to be chosen so as to cope with the interactions $G_m(s)$.

A general method for choosing $P(s)$ is not available yet, however interesting results have been obtained for the case when

$$P(s) = p_i(s)I \qquad (13)$$

with identical entries. So $p_i(s)$ will be choose equal to one of the m characteristic functions $g_i, i = 1,2,...m$ of $[-G_m(s)]$.

$$p_i(s) = -g_k, k \in \{1,2,...,m\} \qquad (14)$$

3.3 Robust controller design procedure

1. Calculation of nominal model $G(s)$;
2. Calculation of $U(s)$ according (6, 9);
3. Set of M_t as minimal value of $U(s)$;
4. Minimal phase margin calculation according (10);
5. Equivalent model $G^{eq}(s)$ calculation according (14, 13, 11);
6. Controllers design for equivalent model subsystems using SISO method ensuring phase margin greater than calculated in point 4.;

Note: If controlled system is SISO point no.5 is omitted and controller is designed for nominal model $G(s)$ **.**

4. CASE STUDY

Consider two tanks process with two inputs (pumps voltage) and two outputs (water level) depicted in figure 2.

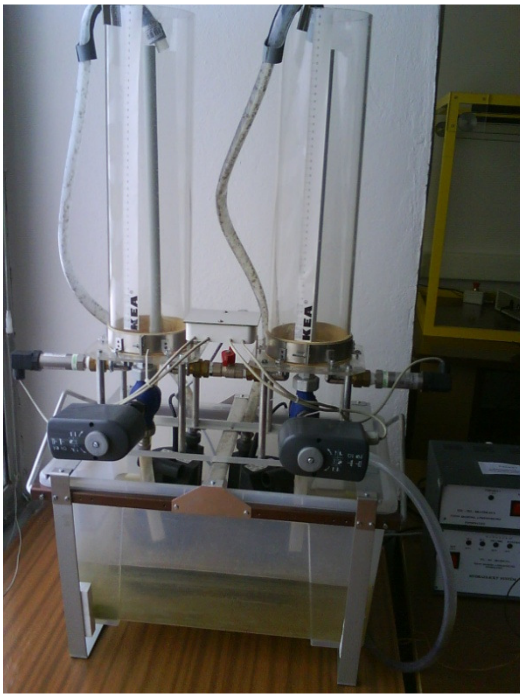


Fig.2 Two tanks process

Process contains also valves which are used for decreasing water level in tanks. Valves voltage can be changed in range (0-10V). Different operating points were obtained using identification with valves at 7.5V, 8.5V and 9.5V. This system has normally no interactions so software interactions were added according to fig. 3.

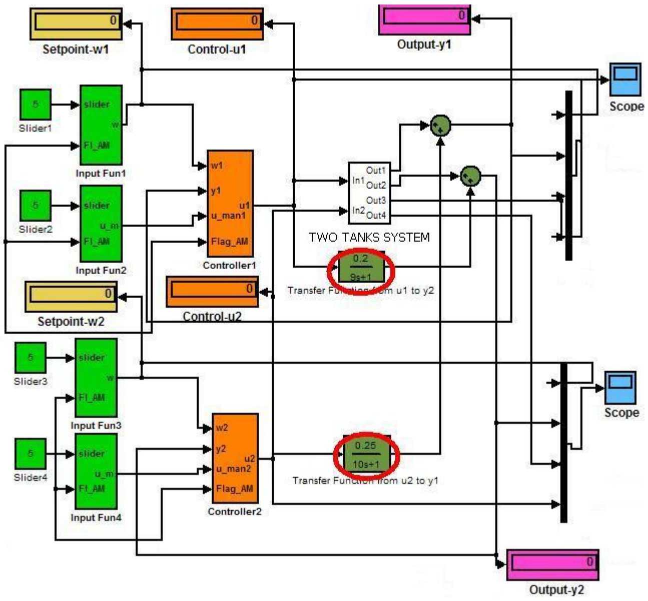


Fig. 3 Simulation diagram with interactions

$$\tilde{G}_1(s)=\begin{bmatrix}\dfrac{0.049s+0.98}{109.2s^2+11.83s+1} & \dfrac{0.25}{10s+1}\\ \dfrac{0.2}{9s+1} & \dfrac{0.059s+1.19}{27.78s^2+10.91s+1}\end{bmatrix}$$

$$\tilde{G}_2(s)=\begin{bmatrix}\dfrac{0.049s+0.97}{46.5s^2+8.76s+1} & \dfrac{0.25}{10s+1}\\ \dfrac{0.2}{9s+1} & \dfrac{0.048s+0.96}{61.06s^2+11.41s+1}\end{bmatrix} \qquad (15)$$

$$\tilde{G}_3(s)=\begin{bmatrix}\dfrac{0.047s+0.94}{38.46s^2+8.21s+1} & \dfrac{0.25}{10s+1}\\ \dfrac{0.2}{9s+1} & \dfrac{0.044s+0.88}{31.2s^2+5.15s+1}\end{bmatrix}$$

Nominal model of this plant was calculated as average value of $\tilde{G}_1(s)$, $\tilde{G}_2(s)$, $\tilde{G}_3(s)$.

$$G(s)=\begin{bmatrix}\dfrac{0.048s+0.96}{64.71s^2+9.6s+1} & \dfrac{0.2}{9s+1}\\ \dfrac{0.25}{10s+1} & \dfrac{0.05s+1}{40.01s^2+9.16s+1}\end{bmatrix} \qquad (16)$$

In this example additive uncertainty will be used so $U(s)$ calculates according (9) and it is depicted in fig. 4.

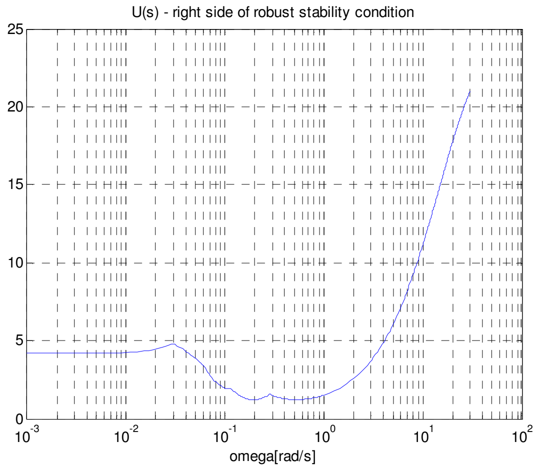


Fig.4 Behavior of $U(s)$

Minimal value of $U(s)$, $U_{\min}(s)=1.205$ is set as M_t which can be recalculate according (10) into minimal phase margin $PM_{\min}=49$ which presumably ensures robust stability.

From nominal model equivalent model will be calculated. At first characteristic functions are calculated from $G_m(s)$ (Fig. 5).

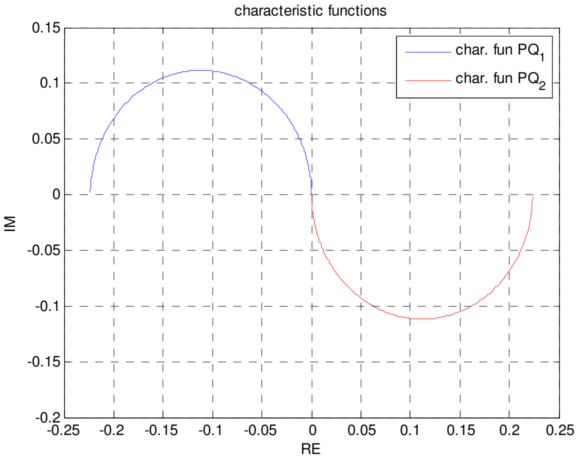


Fig.5 Characteristic functions PQ_1 and PQ_2

For equivalent subsystems calculation according (14, 13 and 11) we use PQ_2. Equivalent subsystems are depicted in fig. 6,7.

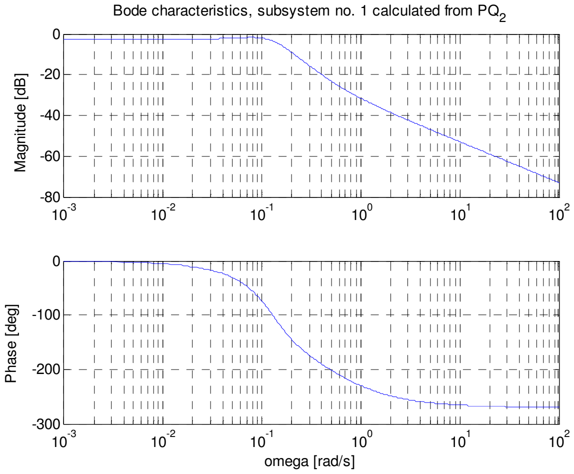


Fig. 6 Bode plot of equivalent subsystem no.1

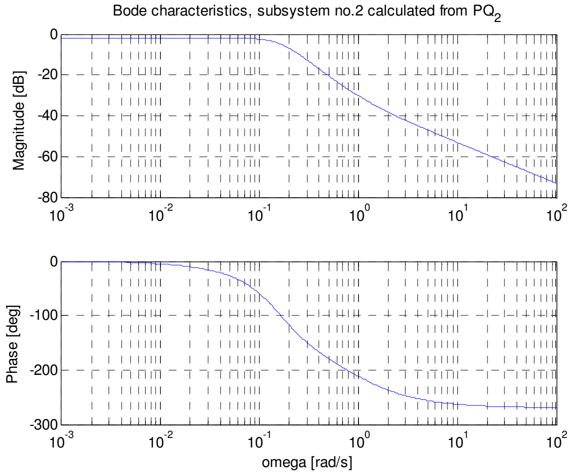


Fig. 7 Bode plot of equivalent subsystem no.2

For each subsystem controller will be calculated. Aim of the controllers design is to fulfill robust conditions and for nominal model have overshoot less than 15%.

Phase margin corresponding to overshoot 15% (Fig. 8) is approximately $PM = 60° > PM_{\min}$.

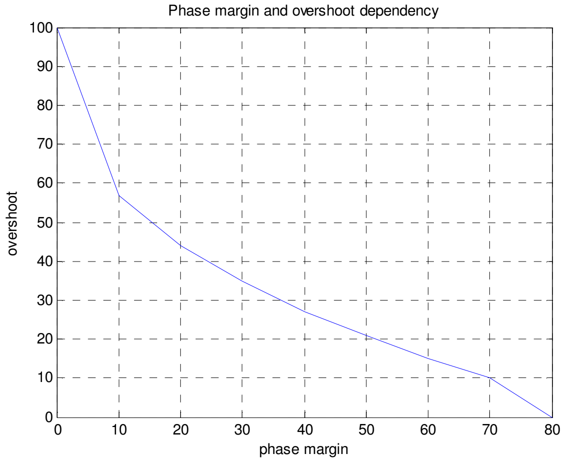


Fig. 8 Dependency of phase margin and overshoot

For subsystem 1 was designed controller with following parameters,

$$r_{11}(s) = 0.558 + \frac{0.09}{s} + 0.284s \tag{17}$$

and for subsystem 2 controller with parameters:

$$r_{22}(s) = 1.03 + \frac{0.13}{s} + 0.41s \tag{18}$$

Decentralized controller $R(s)$ consists of controllers for both subsystems.

$$R(s) = \begin{bmatrix} r_{11} & 0 \\ 0 & r_{22} \end{bmatrix} \tag{19}$$

Reaching of desired crossover frequency and phase margin for both subsystems proofs Fig. 9,10.

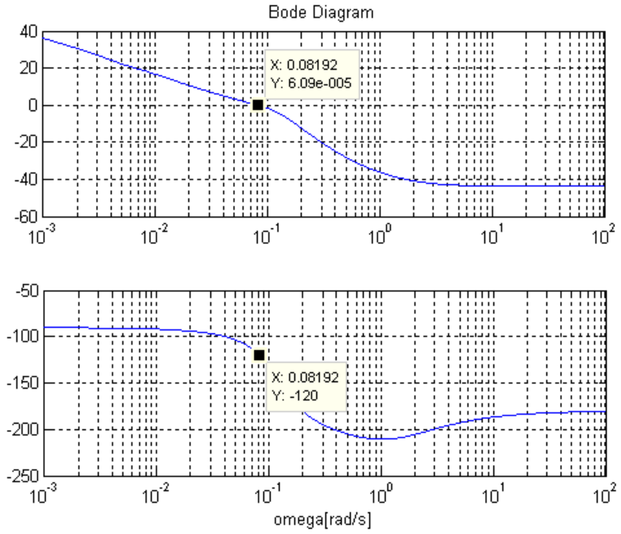


Fig. 9 Bode characteristics for subsystem 1 with PID

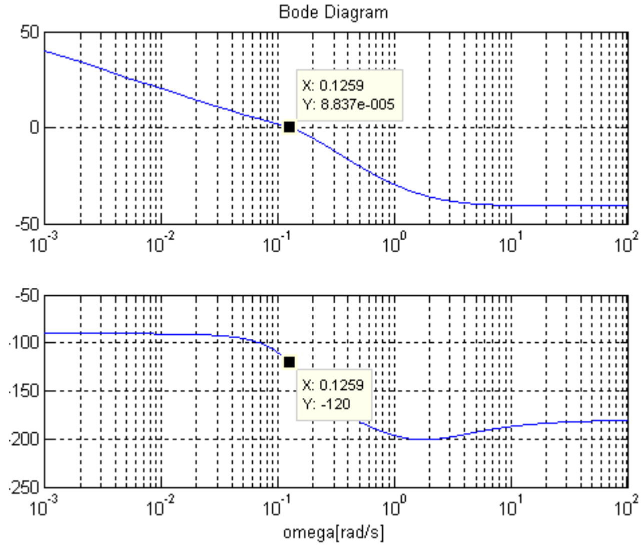


Fig. 10 Bode characteristics for subsystem 2 with PID

Matlab simulation in nominal model (Fig. 11) show that system has overshoot less than 15% by step change in both outputs. So we can see that using controllers with desired phase margin for equivalent subsystems it is possible to reach desired overshoot for nominal model.

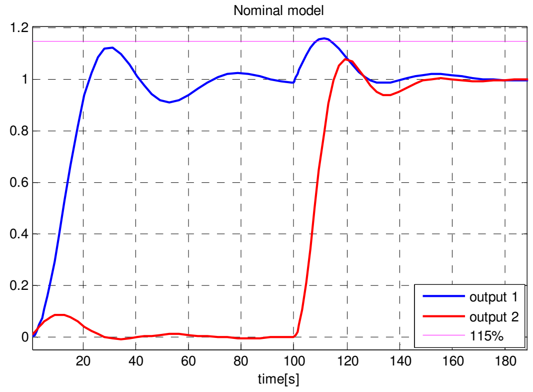


Fig. 11 Nominal model simulation

Because our process is stable, negative roots in operating points (20) proofs stability in this points and robust stability condition (Fig. 12) show that process with designed controller is also robust stable.

$$\Lambda_1 = \{-0{,}14 \pm 0.15i; -0.2 \pm 0.1i; -0.12; -0.07\}$$
$$\Lambda_2 = \{-0.06 \pm 0.15i; -0.05 \pm 0.14i; -0.1 \pm 0.015i; -0.064\}$$
$$\Lambda_3 = \{-0.055 \pm 0.22i; -0.07 \pm 0.15i; -0.09 \pm 0.015i; -0.053\} \quad (20)$$

Finally designed controllers were set on the real process. Step responses in all operating points are depicted in fig. 13.

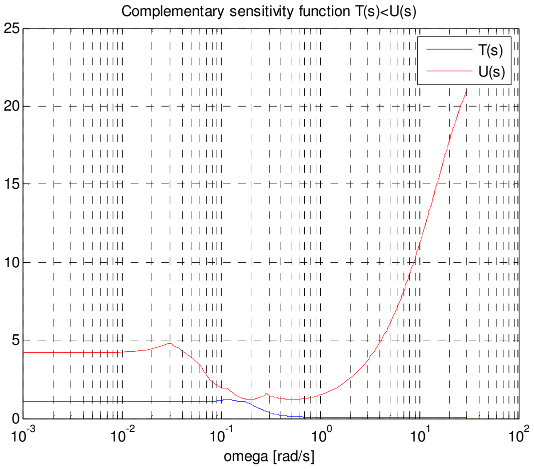


Fig. 12 Robust stability condition

Real process experiments shows that overshoot is really less than 15%, but the step response is different from Matlab simulation. It is due to inaccuracy of models obtained by identification.

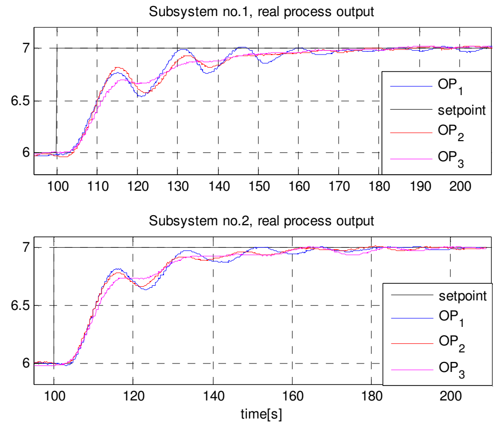


Fig. 13 Step responses of real process

5. CONCLUSION

The paper deals with the robust decentralized controller design in the frequency domain for stable and unstable plants. Equivalent subsystem method was used to simplify the full nominal model matrix into diagonal equivalent one. Controllers were designed for subsystems of equivalent matrix independently, so that desired phase margin in equivalent subsystems guaranteed overshoot for outputs in nominal model. Controller design process was illustrated on two tanks example.

ACKNOWLEDGMENTS

This work was supported by the Scientific Grant Agency of the Ministry of Education of the Slovak Republic Grants No. 1/0544/09 and APVV -0211/10.

REFERENCES

Kozáková A., Veselý V. (2005) *A frequency domain design technique for robust decentralized controllers*. Preprints of the 16th IFAC World Congress. Prague, Czech Republic July 3-8, p. Mo-E21-TO/6.

Kozaková, A., Veselý, V., Osuský, J. (2009). A new Nyquist- based technique for tuning robust decentralized controllers, *Kybernetika*, vol.45, No 1., pp. 63-83

Skogestad, S., Postlethwaite, I. (1997): Multivariable Feedback Control: Analysis and Design. *John Wiley and Sons*

Šulc B., Vítečková, M., (2004) *Theory and practice of control loops design*. ČVUT, Praha (in Czech)

SENSITIVITY ANALYSIS OF HYPERBOLIC OPTIMAL CONTROL SYSTEMS WITH BOUNDARY CONDITIONS INVOLVING TIME DELAYS

A. Kowalewski * and **J. Sokołowski** **

* *Institute of Automatics*
AGH University of Scince and Technology
al. Mickiewicza 30, 30-059 Cracow, Poland
fax: +48 -12 -6341568, e-mail: ako@ia.agh.edu.pl

** *Institut Elie Cartan, Laboratoire de Mathématiques*
Université Henri Poincaré Nancy I
B.P. 239, 54506 Vandoeuvre lés Nancy Cedex, France
e-mail: Jan.Sokolowski@iecn.u-nancy.fr
and
Systems Research Institute of the Polish Academy of Sciences
ul. Newelska 6, 01-447 Warszawa, Poland

Abstract: In the paper the first order sensitivity analysis is performed for a class of optimal control problems for hyperbolic equations with the Neumann boundary conditions involving constant time delays. A singular perturbation of geometrical domain of integration is introduced in the form of a circular hole. The Steklov-Poincaré operator on a circle is defined in order to reduce the problem to regular perturbations in the truncated domain. The optimality system is differentiated with respect to the small parameter and the directional derivative of the optimal control is obtained as a solution to an auxiliary optimal control problem.

Keywords: Sensitivity analysis, hyperbolic system, Neumann boundary condition, time delay.

1. INTRODUCTION

We consider an optimal control problem in the domain with small geometrical defect. The size of the defect is measured by small parameter $\rho > 0$. The presence of the defect results in the singular perturbation of the hyperbolic state equation. Such a perturbation is transformed to the regular perturbation in the truncated domain Ω_R for any $R > \rho > 0$. We perform the sensitivity analysis in the truncated domain using the Steklov-Poincaré operator defined on the circle Γ_R.

The problems of the sensitivity analysis for regular perturbations of optimal control problems were studied in Lasiecka and Sokołowski (1991); Malanowski and Sokołowski (1986); Malanowski (2001); Rao and Sokołowski (2000); Sokołowski (1985 1987 1988); Sokołowski and Zolesio (1992). Singular perturbations of geometrical domains are analysed in Jackowska et al. (2002 2003); Maz'ya et al. (2000); Nazarov (1999); Nazarov and Sokołowski (2004 2003acb); Nazarov et al. (2004); Sokołowski and Żochowski (1999abc 2001 2003). The construction of asymptotic approximation for the Steklov-Poincaré operator is given in Sokołowski and Żochowski (2005). In particular, in Kowalewski et al. (2010) the sensivity analysis of optimal control problems defined for the wave equation is performed. The small parameter describes the size of an imperfection in the form of a small hole or cavity in the geometrical domain of integration. The initial state equation in the singularly perturbed domain is replaced by the equation in a smooth domain. The imperfection is replaced by its approximation defined by a suitable Steklov's type differential operator. For approximate optimal control problems the well-posedness is shown. One term asymptotics of optimal control are derived and justified for the approximate model. The key role in the arguments is played by the so called "hidden regularity" of boundary traces generated by hyperbolic solutions.

The idea of "hidden regularity" regalarization has been used in the past successfully for boundary control problems, particulary in the context of numerical approximations (Hendrickson and Lasiecka (1993 1995); Lagnese and Leugering (2004); Lasiecka and Triggiani (2000)). Regularizing parameter allows to obtain smooth on the boundary approximations, which can be then taken to appropriate limits. The property of "hidden regularity" is displayed by hyperbolic flows which satisfy the Lopatinski condition (Harmander (1985); Lasiecka et al. (1986); Lasiecka and

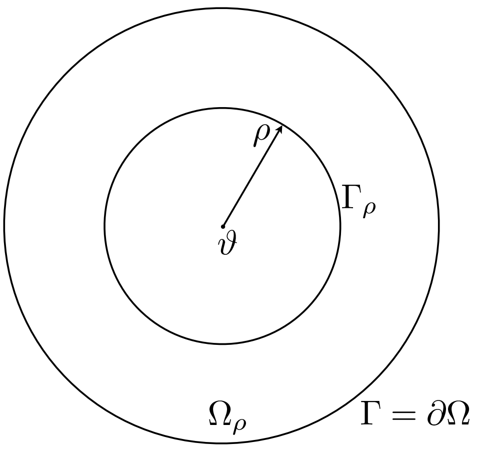


Fig. 1. The domain Ω_ρ in two spatial dimensions.

Triggiani (1990 1991); Sakamoto (1982)). The method of "hidden regularity" regularization has been also applied in domain decomposition procedures introduced and described in Lagnese and Leugering (2004).

In the present paper an optimal control problem in singularly perturbed geometrical domain Ω_ρ is analysed with respect to small parameter $\rho > 0$. We derive the one-term asymptotic expansion of optimal controls. The first term of the expansion, of the order ρ^2 is uniquely determined as an optimal solution to the auxiliary optimal control problem. The control constraints for the auxiliary problem are obtained by an application of the conical differentiability of metric projection in L^2 spaces. Our method is constructive and can lead to numerical procedures for determination of the first order approximations of the optimal controls.

2. PRELIMINARIES

Consider now the distributed parameter system described by the following time delay hyperbolic equation

$$\left.\begin{array}{ll} \dfrac{\partial^2 y}{\partial t^2} - \Delta y = f & \text{in } \ \Omega_\rho \times (0,T), \\ \dfrac{\partial y}{\partial \eta} = y(x, t-h) + Gv & \text{on } \ \Gamma \times (0,T), \\ \dfrac{\partial y}{\partial \eta} = 0 & \text{on } \ \Gamma_\rho \times (0,T), \\ y(x,0) = y_0(x) & \text{in } \ \Omega_\rho, \\ \dfrac{\partial y}{\partial t}(x,0) = y_I(x) & \text{in } \ \Omega_\rho, \\ y(x,t') = \Psi_0(x,t') & \text{in } \ \Gamma \times [-h,0), \end{array}\right\} \tag{1}$$

where:

$$\Delta = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}, \ \ G \in \mathcal{L}(L^2(\Sigma), H^{-5/2}\Xi^{-5/2}(\Sigma)),$$

h is a specified positive number representing a time delay, Ψ_0 is an initial function defined on $\Gamma \times [-h, 0)$, $\partial/\partial\eta$ is a normal derivative at $\Gamma\rho$ directed towards the exterior of $\Omega\rho$, $\Omega\rho$ is presented on the Fig. 1.

We denote by

$$\Omega_\rho = \Omega \setminus \overline{B(\rho)} \subset R^2, \quad \partial\, \Omega_\rho = \Gamma \cup \Gamma_\rho, \tag{2}$$

where: Ω is a domain on the plane R^2 with a smooth boundary $\partial\, \Omega$ and

$$B_\rho = \{x : |x - \vartheta| < \rho\} \tag{3}$$

with a smooth boundary $\Gamma\rho$.

First we shall present sufficient conditions for the existence of a unique solution of the problem (1) for the case where the boundary control $v \in L^2(\Sigma)$.

For this purpose, we introduce the space $\mathcal{D}^{-1}_{A+\mathcal{D}_t^2}(Q)$ (Lions and Magenes (1972), vol. 2, p.131) defined by

$$\begin{array}{l} \mathcal{D}^{-1}_{A+\mathcal{D}_t^2}(Q) \overset{df}{=} \\ \{y | y \in H^{-1,-2}(Q), y'' + Ay \in \Xi^{-3,-3}(Q)\}, \end{array} \tag{4}$$

where: the spaces $H^{-1,-2}(Q)$ and $\Xi^{-3,-3}(Q)$ are defined by (9.5) and (10.4) of Chapter 5 in (Lions and Magenes (1972), vol. 2) respectively. Under the norm of the graph $\mathcal{D}^{-1}_{A+\mathcal{D}_t^2}(Q)$ is a Hilbert space.

The existence of a unique solution for the mixed initial-boundary value problem (1) on the cylinder Q can be proved using a constructive method, i.e. first solving (1) on the subcylinder Q_1 and in turn on Q_2 etc., until the procedure covers the whole cylinder Q. In this way the solution in the previous step determines the next one.

For simplicity, we introduce the following notations:

$$\left.\begin{array}{l} Q = \Omega_\rho \times (0,T) \\ \Sigma = \Gamma \times (0,T) \\ E_j \triangleq ((j-1)h, jh) \\ Q_j = \Omega_\rho \times E_j \\ \Sigma_j = \Gamma \times E_j \\ \Sigma_0 = \Gamma \times [-h, 0) \end{array}\right\} \text{ for } \quad j = 1, ..., K. \tag{5}$$

Using Theorem 10.1 of (Lions and Magenes (1972), vol. 2, p. 132) we can prove the following result.

Theorem 1. Let y_0, y_I, Ψ_0, v and f be given with $y_0 \in \Xi^{-3/2}(\Omega)$, $y_I \in \Xi^{-5/2}(\Omega)$,
$\Psi_0 \in H^{-5/2}\Xi^{-5/2}(\Sigma_0)$, $v \in L^2(\Sigma)$ and $f \in \Xi^{-3,3}(Q)$. Then there exists a unique solution $y \in \mathcal{D}^{-1}_{A+\mathcal{D}_t^2}(Q)$ for the problem (1). Moreover, $y(\cdot, jh) \in \Xi^{-3/2}(\Omega)$ and $\dfrac{\partial y}{\partial t}(\cdot, jh) \in \Xi^{-5/2}(\Omega)$ for $j = 1, ...K$.

The spaces appearing in the Theorem 1 are defined in Lions and Magenes (1972).

Let us surround Γ_ρ by the circle Γ_R such that $R > \rho > 0$ (Fig. 2) .

Consequently, we denote

$$\Omega_R = \Omega \setminus \overline{B(R)}, \tag{6}$$

where:

$$B(R) = \{x : |x - \vartheta| < R\}. \tag{7}$$

We set the non-local Neumann boundary condition on Γ_R:

$$\frac{\partial y}{\partial \eta} = A_\rho(y) \text{ on } \Gamma_R, \tag{8}$$

where: A_ρ is a Steklov-Poincare operator defined in the domain $C(R,\rho) = B(R) \setminus \overline{B(\rho)}$.The operator A_ρ is a

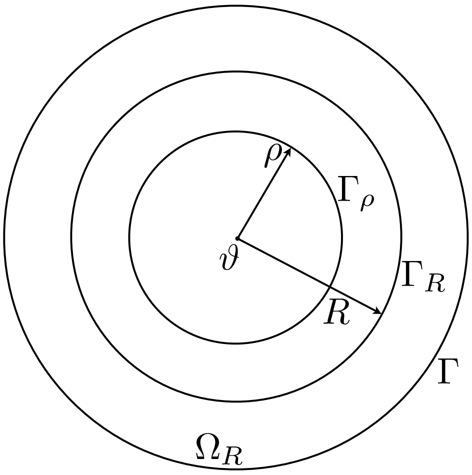


Fig. 2. The domain Ω_R.

mapping of $H^{1/2}(\Gamma_R) \to H^{-1/2}(\Gamma_R)$. Consequently, we consider in $\Omega_R \times (0,T)$ the following time delay hyperbolic equation:

$$\left.\begin{array}{lll} \dfrac{\partial^2 y}{\partial t^2} - \Delta y = f & \text{in} & \Omega_R \times (0,T), \\ \dfrac{\partial y}{\partial \eta} = y(x,t-h) + Gv & \text{on} & \Gamma \times (0,T), \\ \dfrac{\partial y}{\partial \eta} = A_\rho(y) & \text{on} & \Gamma_R \times (0,T), \\ y(x,0) = y_0(x) & \text{in} & \Omega_R. \\ \dfrac{\partial y}{\partial t}(x,0) = y_I(x) & \text{in} & \Omega_R, \\ y(x,t') = \Psi_0(x,t') & \text{in} & \Gamma \times [-h,0), \end{array}\right\} \quad (9)$$

We shall investigate the dependence of optimal solutions on the small parameter $\rho > 0$.

The small hole $B(\rho)$ is a singular perturbation in the domain $\Omega\rho$. Consequently, the same small hole constitutes regular perturbation in the domain Ω_R.

Using the results of Sokołowski and Żochowski (2005) we obtain the following expansion for the operator A_ρ:

$$\begin{array}{l} A_\rho = A_0 + \rho^2 B + O(\rho^4) \\ \text{in the operator norm} \\ \mathcal{L}(H^{1/2}(\Gamma_R), H^{-1/2}(\Gamma_R)), \end{array} \quad (10)$$

where: the remainder $O(\rho^4)$ is uniformly bounded on bounded sets in the space $H^{1/2}(\Gamma_R)$.

Corollary 1. In the space $\mathcal{D}^{-1}_{A+\mathcal{D}^2_t}(Q)$ the solution of the hyperbolic equation (for $\rho = 0$) can be represented as

$$\left.\begin{array}{lll} \dfrac{\partial^2 y^0}{\partial t^2} - \Delta y^0 = f & \text{in} & \Omega_R \times (0,T), \\ \dfrac{\partial y^0}{\partial \eta} = y^0(x,t-h) + Gv & \text{on} & \Gamma \times (0,T), \\ \dfrac{\partial y^0}{\partial \eta} = A_0(y^0) & \text{on} & \Gamma_R \times (0,T), \\ y^0(x,0) = y_0(x) & \text{in} & \Omega_R, \\ \dfrac{\partial y^0}{\partial t}(x,0) = y_I(x) & \text{in} & \Omega_R, \\ y^0(x,t') = \Psi_0(x,t') & \text{in} & \Gamma \times [-h,0). \end{array}\right\} \quad (11)$$

We shall look the expansion of the solution y^ρ in $\Omega_R \times (0,T)$:

$$\begin{aligned} y^\rho &= y^0 + \rho^2 y^1 + \tilde{y} = \\ &= y^0 + \rho^2 y^1 + \rho^4 \hat{y} \end{aligned} \quad (12)$$

Consequently, the Neumann boundary condition in (9) can be rewritten as

$$\begin{aligned} \frac{\partial y^\rho}{\partial \eta} &= A_\rho(y^\rho) = \\ &= A_0(y^\rho) + \rho^2 B(y^\rho) + \rho^4 \tilde{A}(y^\rho) \end{aligned} \quad (13)$$

Substituting (12) into (13) we obtain

$$\begin{aligned} &\frac{\partial y^0}{\partial \eta} + \rho^2 B \frac{\partial y^1}{\partial \eta} + \frac{\partial \tilde{y}}{\partial \eta} = \\ &= A_0(y^0 + \rho^2 y^1 + \tilde{y}) + \\ &+ \rho^2 B(y^0 + \rho^2 y^1 + \tilde{y}) + \rho^4 \tilde{A}(y^\rho) \end{aligned} \quad (14)$$

Comparing components with the same powers we get

$$\left.\begin{array}{l} \rho^0 : \dfrac{\partial y^0}{\partial \eta} = A_0(y^0) \\ \rho^2 : \rho^2 \dfrac{\partial y^1}{\partial \eta} = \rho^2 [A_0 y^1 + B y^0] \end{array}\right\} \quad (15)$$

Hence it follows the following expansion of solutions:

Let us denote by y^0 the solution of the problem (11) corresponding to a given parameter $\rho = 0$.

Subsequently, y^1 corresponding to a given parameter ρ^2 is a solution of the following equation:

$$\left.\begin{array}{lll} \dfrac{\partial^2 y^1}{\partial t^2} - \Delta y^1 = 0 & \text{in} & \Omega_R \times (0,T), \\ \dfrac{\partial y^1}{\partial \eta} = y^1(x,t-h) + Gv & \text{on} & \Gamma \times (0,T), \\ \dfrac{\partial y^1}{\partial \eta} = A_0(y^1) + B(y^0) & \text{on} & \Gamma_R \times (0,T), \\ y^1(x,0) = 0 & \text{in} & \Omega_R, \\ \dfrac{\partial y^1}{\partial t}(x,0) = 0 & \text{in} & \Omega_R, \\ y^1(x,t') = \Psi_0(x,t') & \text{in} & \Gamma \times [-h,0). \end{array}\right\} \quad (16)$$

3. PROBLEM FORMULATION. OPTIMIZATION THEOREM.

We shall now consider the optimal boundary control problem in domains Ω_ρ and Ω_R respectively. Let us denote by $U = L^2(\Gamma \times (0,T))$ the space of controls. The time horizon T is fixed in our problem.

Let us consider in $\Omega_\rho \times (0,T)$ the following time delay hyperbolic equation

$$\left.\begin{array}{lll}
\dfrac{\partial^2 y}{\partial t^2} - \Delta y = f & \text{in} & \Omega_\rho \times (0,T), \\
 & \text{supp} & f \subset \Omega_R \times (0,T), \\
\dfrac{\partial y}{\partial \eta} = y(x,t-h) + Gv & \text{on} & \Gamma \times (0,T), \\
\dfrac{\partial y}{\partial \eta} = 0 & \text{on} & \Gamma_\rho \times (0,T), \\
y(x,0) = y_0(x) & \text{in} & \Omega_\rho \\
 & \text{supp} & y_0 \subset \Omega_R, \\
\dfrac{\partial y}{\partial t}(x,0) = y_I(x) & \text{in} & \Omega_\rho, \\
 & \text{supp} & y_I \subset \Omega_R, \\
y(x,t') = \Psi_0(x,t') & \text{in} & \Gamma \times [-h,0).
\end{array}\right\} \tag{17}$$

The performance functional is given by

$$\begin{aligned} I(v) = & \frac{1}{2}\Big\| \; y(v) - z_d \; \Big\|^2_{H^{-1,-2}(\Omega_R \times (0,T))} \\ & + \frac{\alpha}{2}\Big\| \; v \; \Big\|^2_{L^2(\Gamma\times(0,T))}. \end{aligned} \tag{18}$$

Finally, we assume the following constraints on the control $v \in U_{ad}$:

$$U_{ad} = \{v \in L^2(\Gamma \times (0,T)), 0 \le v(x,t) \le 1\}. \tag{19}$$

Subsequently, we consider in $\Omega_R \times (0,T)$ the following hyperbolic time delay equation

$$\left.\begin{array}{lll}
\dfrac{\partial^2 y}{\partial t^2} - \Delta y = f & \text{in} & \Omega_R \times (0,T), \\
\dfrac{\partial y}{\partial \eta} = y(x,t-h) + Gv & \text{on} & \Gamma \times (0,T), \\
\dfrac{\partial y}{\partial \eta} = A_\rho(y) & \text{on} & \Gamma_R \times (0,T), \\
y(x,0) = y_0(x) & \text{in} & \Omega_R, \\
\dfrac{\partial y}{\partial t}(x,0) = y_I(x) & \text{in} & \Omega_R, \\
y(x,t') = \Psi_0(x,t') & \text{in} & \Gamma \times [-h,0).
\end{array}\right\} \tag{20}$$

The performance functional and constraints on the control are given by (18) and (19).

Result: The Solution of the problem (20) (in the domain Ω_R) is a restriction of the solution of the problem (17) (in the domain Ω_ρ) to Ω_R. Hence, we have the possibility of replacing the singular perturbation of the domain $B(\rho)$ by the regular perturbation on the boundary Γ_R in a smaller domain Ω_R. Consequently, we shall analyse the optimal boundary control problem (18)-(20) in the domain Ω_R. Moreover, we assume the fixed parameter $\rho > 0$. The solving of the formulated optimal control problem is equivalent to seeking a $v_0 \in U_{ad}$ such that $I(v_0) \le I(v)\ \forall v \in U_{ad}$.

From Lions' scheme (Theorem 1.3 Lions (1971), p. 10) it follows that for $\alpha > 0$ a unique optimal control v_0 is characterized by the following condition

$$I'(v_0)(v - v_0) \ge 0 \quad \forall v \in U_{ad}. \tag{21}$$

Using the form of the performance functional (18) we can express (21) in the following form:

$$\begin{aligned} & \Big\langle (y(v_0) - z_d, y(v) - y(v_0) \Big\rangle_{H^{-1,-2}(\Omega_R\times(0,T))} \\ & + \alpha \Big\langle v_0, v - v_0 \Big\rangle_{L^2(\Gamma\times(0,T))} \ge 0 \quad \forall v \in U_{ad}. \end{aligned} \tag{22}$$

To simplify (22), we introduce the adjoint equation and for every $v \in U_{ad}$. we define the adjoint variable $p = p(v) = p(x,t;v)$ as the solution of the following equation

$$\left.\begin{array}{ll}
\dfrac{\partial^2 p}{\partial t^2} - \Delta p = y(v) - \; z_d & \text{in } \Omega_R \times (0,T), \\
\dfrac{\partial p}{\partial \eta} = p(x,t+h) & \text{on } \Gamma \times (0, T-h), \\
\dfrac{\partial p}{\partial \eta} = 0 & \text{on } \Gamma \times (T-h, T), \\
\dfrac{\partial p}{\partial \eta} = A_\rho(p) & \text{on } \Gamma_R \times (0,T), \\
p(x,T;v) = 0 & \text{in } \Omega_R, \\
p'(x,T;v) = 0 & \text{in } \Omega_R.
\end{array}\right\} \tag{23}$$

Theorem 2. Let the hypothesis of Theorem 1 be satisfied. Then for given $z_d \in H^{-1,-2}(\Omega_R \times (0,T))$ and any $v_0 \in L^2(\Sigma)$, there exists a unique solution $p(v_0) \in H^{3,3}(\Omega_R \times (0,T)) \subset \Xi^{3,3}(\Omega_R \times (0,T))$ for the problem (23).

We simplify (22) using the adjoint equation (23). Consequently, after transformations we obtain the following formula

$$\begin{aligned} \Big\langle G^* p + \alpha\, v_0, v - v_0 \Big\rangle_{L^2(\Gamma\times(0,T))} \ge 0 \\ \forall v \in U_{ad}. \end{aligned} \tag{24}$$

Theorem 3. For the problem (20) with the performance functional (18) with $\alpha > 0$, and with constraints on the control (19), there exists a unique optimal control v_0 which satisfies the maximum condition (24). Moreover, $v_0 = P_{U_{ad}}\left(-\dfrac{1}{\alpha} G^* p\right)$ where $P_{U_{ad}}$ is a projective operator.

4. THE SENSITIVITY OF OPTIMAL CONTROLS

Theorem 4. We have the following expansion of the optimal control in $L^2(\Gamma \times (0,T))$, with respect to the small parameter,

$$v_\rho = v_0 + \rho^2 q + o(\rho^2) \tag{25}$$

for $\rho > 0$.

Moreover, we assume that ρ is a sufficiently small. The function q in (25) is a optimal solution of the following optimal control problem:

The state equation

$$\left.\begin{array}{ll} \dfrac{\partial^2 w}{\partial t^2} - \Delta w = 0 & \text{in } \Omega_R \times (0,T), \\ \dfrac{\partial w}{\partial \eta} = w(x,t-h) + Gq & \text{on } \Gamma \times (0,T), \\ \dfrac{\partial w}{\partial \eta} = A_0(w) + B(y^0) & \text{on } \Gamma_R \times (0,T), \\ w(x,0) = 0 & \text{in } \Omega_R, \\ \dfrac{\partial w}{\partial t}(x,0) = 0 & \text{in } \Omega_R, \\ w(x,t') = \Psi_0(x,t') & \text{on } \Gamma \times [-h,0), \end{array}\right\} \quad (26)$$

where: $w = y^1$.

The performance functional

$$\begin{aligned} I(u) = {} & \frac{1}{2}\Big\| w(q) \Big\|^2_{H^{-1,-2}(\Omega_R \times (0,T))} \\ & + \frac{\alpha}{2}\Big\| u \Big\|^2_{L^2(\Gamma \times (0,T))}. \end{aligned} \quad (27)$$

The adjoint equation

$$\left.\begin{array}{ll} \dfrac{\partial^2 z}{\partial t^2} - \Delta z = w(q) & \text{in } \Omega_R \times (0,T), \\ \dfrac{\partial z}{\partial \eta} = z(x,t+h) & \text{on } \Gamma \times (0,T-h), \\ \dfrac{\partial z}{\partial \eta} = 0 & \text{on } \Gamma \times (T-h,T), \\ \dfrac{\partial z}{\partial \eta} = A_0(z) + B(p^0) & \text{on } \Gamma_R \times (0,T), \\ z(x,T) = 0 & \text{in } \Omega_R, \\ z'(x,T) = 0 & \text{in } \Omega_R, \end{array}\right\} \quad (28)$$

where: $z = p^1$.

Then, the optimal control q is characterized by

$$\begin{aligned} & \Big\langle w(q), w(u) - w(q) \Big\rangle_{H^{-1,-2}(\Omega_R \times (0,T))} \\ & + \alpha \Big\langle q, u - q \Big\rangle_{L^2(\Gamma \times (0,T))} \geq 0 \quad \forall v \in U_{ad}, \end{aligned} \quad (29)$$

where: S_{ad} is a set of admissible controls such that

$$\begin{aligned} S_{ad} = \Big\{ & u \in L^2(\Gamma \times (0,T)) \Big| \\ & u(x,t) \geq 0 \text{ on the set} \\ & \qquad E_0 = \{(x,t) | v_0(x,t) = 0\}, \\ & u(x,t) < 0 \text{ on the set} \\ & \qquad E_1 = \{(x,t) | v_0(x,t) = 1\}, \\ & \Big\langle G^* p_0 + \alpha\, v_0, u \Big\rangle_{L^2(\Gamma \times (0,T))} = 0 \}, \end{aligned} \quad (30)$$

where:

p_0 is a adjoint state for $\rho = 0$,

v_0 is a optimal solution for $\rho = 0$ such that

$0 \leq v_0(x,t) \leq 1$.

We simplify (29) using the adjoint equation (28). After transformations we obtain the following maximum condition

$$\begin{aligned} \Big\langle G^* z + \alpha\, q, u - q \Big\rangle_{L^2(\Gamma \times (0,T))} & \geq 0 \\ \forall u & \in S_{ad}. \end{aligned} \quad (31)$$

Theorem 5. For the time delay hyperbolic problem

$$\left.\begin{array}{ll} \dfrac{\partial^2 w}{\partial t^2} - \Delta w = 0 & \text{in } \Omega_R \times (0,T), \\ \dfrac{\partial w}{\partial \eta} = w(x,t-h) + Gu & \text{on } \Gamma \times (0,T), \\ \dfrac{\partial w}{\partial \eta} = A_0(w) + B(y^0) & \text{on } \Gamma_R \times (0,T), \\ w(x,0) = 0 & \text{in } \Omega_R, \\ \dfrac{\partial w}{\partial t}(x,0) = 0 & \text{in } \Omega_R, \\ w(x,t') = \Psi_0(x,t') & \text{in } \Gamma \times [-h,0), \end{array}\right\} \quad (32)$$

with the performance functional (27) with $\alpha > 0$, and with constraints on the control (30), there exists a unique optimal control q which satisfies the maximum condition (31).

5. CONCLUSIONS

The results presented in the paper can be treated as a generalization of the results obtained in Sokołowski and Żochowski (2005) onto the case of hyperbolic systems with boundary condition involving time delays.

In this paper we have considered the mixed initial boundary value problems of hyperbolic type.

We can also consider similar optimal control problems for parabolic-hyperbolic systems.

The ideas mentioned above will be developed in forthcoming papers.

ACKNOWLEDGEMENTS

The research presented here was carried out within the research programme AGH University of Science and Technology, No. 11.11.120.768.

REFERENCES

Harmander, L. (1985). *The Analysis of Linear Partial Differential Operators - Vol.III.* Springer-Verlag, Berlin-Heidelberg.

Hendrickson, E. and Lasiecka, I. (1993). Numerical approximations and regularizations of Ricati equations arising in hyperbolic dynamics with unbounded control operators. *Computational Optimization and Applications*, 2, 343–390.

Hendrickson, E. and Lasiecka, I. (1995). Finite dimensional approximations of boundary control problems arising in partially observed hyperbolic systems. *Dynamics of Continuous Discrete and Impulsive Systems*, 1, 101–142.

Jackowska, L., Sokołowski, J., and Żochowski, A. (2003). Topological optimization and inverse problems. *Computer Assisted Mechanics and Engineering Sciences*, 10, 163–176.

Jackowska, L., Sokołowski, J., Żochowski, A., and Henrot, A. (2002). On numerical solution of shape inverse problems. *Computational Optimization and Applications*, 23, 231–255.

Kowalewski, A., Lasiecka, I., and Sokołowski, J. (2010). Sensitivity analysis of hyperbolic optimal control problems (published on-line: 20 November 2010). *Computational Optimization and Applications (*to appear).

Lagnese, J. and Leugering, G. (2004). *Domain Decomposition Methods in Optimal Control of Partial Differential Equations.* Birkhäuser, Basel.

Lasiecka, I., Lions, J., and Triggiani, R. (1986). Nonhomogeneous boundary value problems for second order hyperbolic operators. *Journal de Mathematiques Pures et Appliquees*, 65, 149–192.

Lasiecka, I. and Sokołowski, J. (1991). Sensitivity analysis of constrained optimal control problem for wave equation. *SIAM Journal on Control and Optimization*, 29, 1128–1149.

Lasiecka, I. and Triggiani, R. (1990). Sharp regularity results for second order hyperbolic equations of Neumann type. *Annali di Matematica Pura ed Applicata*, CLVII, 1128–1149.

Lasiecka, I. and Triggiani, R. (1991). Regularity theory of hyperbolic equations with non-homogeneous Neumann boundary conditions. *Journal of Differetial Equations*, 94, 112–164.

Lasiecka, I. and Triggiani, R. (2000). *Control Theory for Partial Differential Equations.* Cambridge University Press, Cambridge.

Lions, J. (1971). *Optimal Control of Systems Governed by Partial Differential Equations.* Springer-Verlag, Berlin-Heidelberg.

Lions, J. and Magenes, E. (1972). *Non-Homogeneous Boundary Value Problems and Applications –Vols. 1 adn. 2.* Springer-Verlag, Berlin-Heidelberg.

Malanowski, K. (2001). *Stability and sensitivity analysis for optimal control problems with control-state constraints.* Disertations Math. (Rozprawy Mat.), Warsaw.

Malanowski, K. and Sokołowski, J. (1986). Sensitivity of solutions to convex, control constrained optimal control problems for distributed parameter systems. *Journal of Mathematical Analysis and Applications*, 120, 240–263.

Maz'ya, V., Nazarov, S., and Plamenevskij, B. (2000). *Asymptotic theory of elliptic boundary value problems in singularly perturbed domains -vol.1.* Birkhäuser Verlag, Basel.

Nazarov, S.A. (1999). Asymptotic conditions at a point, self adjoint extensions of operators and the method of matched asymptotic expansions. *American Mathematical Society Translations (2)*, 198, 77–125.

Nazarov, S.A., Slutskij, S.A., and Sokołowski, J. (2004). Topological derivative of the energy functional due to formation of a thin ligament on a spatial body. *Les prépublications de l'Institut Élie Cartan*, 14.

Nazarov, S.A. and Sokołowski, J. (2003a). Asymptotic analysis of shape functionals. *Journal de Mathématiques pures et appliquées*, 82, 125–196.

Nazarov, S.A. and Sokołowski, J. (2003b). Self adjoint extensions for the neumann laplacian in application to shape optimization. *Les prépublications de l'Institut Élie Cartan*, 9.

Nazarov, S.A. and Sokołowski, J. (2003c). Self adjoint extensions of differential operators in application to shape optimization. *Comptes Rendus Mecanique*, 331, 667–672.

Nazarov, S.A. and Sokołowski, J. (2004). The topological derivative of the dirichlet integral due to formation of a thin ligament. *Siberian Math. J.*, 45, 341–355.

Rao, M. and Sokołowski, J. (2000). Tangent sets in banach spaces and applications to variational inequalities. *Les prépublications de l'Institut Élie Cartan*, 42.

Sakamoto, R. (1982). *Hyperbolic Boundary Value Problems.* Cambridge University Press, Cambridge.

Sokołowski, J. (1985). Differential stability of solutions to constrained optimization problems. *Appl. Math. Optim.*, 13, 97–115.

Sokołowski, J. (1987). Sensitivity analysis of control constrained optimal control problems for distributed parameter systems. *SIAM J. Control and Optimization*, 25, 1542–1556.

Sokołowski, J. (1988). Shape sensitivity analysis of boundary optimal control problems for parabolic systems. *SIAM J. Control and Optimization*, 26, 763–787.

Sokołowski, J. and Żochowski, A. (1999a). On topological derivative in shape optimization. *SIAM J. Control and Optimization*, 37, 1251–1272.

Sokołowski, J. and Żochowski, A. (1999b). Topological derivative for optimal control problems. *Control and Cybernetics*, 28, 611–626.

Sokołowski, J. and Żochowski, A. (1999c). Topological derivatives for elliptic problems. *Inverse Problems*, 1, 123–134.

Sokołowski, J. and Żochowski, A. (2001). Topological derivatives of shape functionals for elasticity systems. *Mechanics of Structures and Machines*, 29, 333–351.

Sokołowski, J. and Żochowski, A. (2003). Optimality conditions for simultaneous topology and shape optimization. *SIAM J. Control and Optimization*, 42, 1198–1221.

Sokołowski, J. and Żochowski, A. (2005). Topological derivatives for obstacle problems. *Les prépublications de l'Institut Élie Cartan*, 12.

Sokołowski, J. and Zolesio, J.P. (1992). *Introduction to Shape Optimization. Shape Sensitivity Analysis.* Springer Verlag, Berlin-Heidelberg.

Tighter Convex Relaxations for Global Optimization Using αBB Based Approach

R. Paulen * **and M. Fikar** *

* *Institute of Information Engineering, Automation and Mathematics*
Slovak University of Technology in Bratislava
e-mail: {radoslav.paulen, miroslav.fikar}@stuba.sk
url: http://www.kirp.chtf.stuba.sk

Abstract: This paper is devoted to investigation of certain issues that appear in solving of deterministic global optimization problems (GOPs). Basically, we focus ourselves on introducing a procedure which may serve to establish tighter convex relaxations for a certain class of non-convex optimization problems. Tightness of these convex relaxations plays important role in speeding of the convergence of branch-and-bound algorithm which is used as a basic framework of solving GOPs in this study. Two case studies are solved where it is shown how significant improvement can be achieved by considering proposed framework.

Keywords: Global Optimization, Convex Relaxation, αBB Method

1. INTRODUCTION

Global optimization (GO) represents a set of methods which aim to find (global) solution of non-convex optimization problems which may possess multiple suboptimal (local) solutions and are typically encountered in many engineering fields, including chemical engineering, process design, computational biology, and many others. Over past two decades, there was a significant effort dedicated to deterministic GO by many scientists. Efficient algorithms and methods were developed, many new interesting applications were introduced and lot of existing non-convex optimization problems were solved to global optimality. Essence of these can be found in works Floudas and Visweswaran (1990); Adjiman and Floudas (1996); Singer and Barton (2001); Papamichail and Adjiman (2002); Chachuat et al. (2003); Čižniar et al. (2009).

There is a big range of problems addressed by global optimization. Basically non-convex non-linear programs (NLPs) are considered. However, it is popular to convert mixed-integer linear programs (MILPs) and mixed-integer non-linear (MINLPs) to non-convex NLPs. Also the problems of dynamic optimization are usually discretized into NLPs, e.g. by using the method of orthogonal collocation (Biegler, 1984). In all these problems GO plays significant role. The problems are addressed using either stochastic approaches, such as simulated annealing, particle swarm and genetic algorithms, or deterministic ones, such as branch-and-bound or interval analysis methods.

Branch-and-bound (BB) methods are the most popular deterministic GO frameworks. These methods successively partition solution space on which optimization problem is defined into smaller regions. In each region, the upper and lower bounds to the objective function value are generated, by solving the original (non-convex) problem together with its convex relaxation. According to these bounds it is decided whether region is going to be explored further or whether it should be fathomed out of BB tree. Global solution is then obtained once current best (lowest) upper bound (UB) value is close to current best (highest) lower bound (LB) value within specified tolerance ε.

Problems of GO are typically defined over a quite large region of decision variables. However, each deterministic GO algorithm investigates the whole solution space in some manner. This is a critical issue and it is a reason why very tight convex relaxations of non-convex problems are needed. It is the aim of this study to present a technique which involves simple algebraic manipulations but results in a considerable improvement in terms of number of GO algorithm iterations and algorithm run time.

The paper is organized as follows. Section 1 gives the mathematical formulation of the problem, it shows how the solution can be found and points out to some issues which are motivating our research. Section 2 gives procedure of how certain issues revealed in Section 1 may be avoided. Finally in Section 3 selected case studies are solved to prove efficacy of the proposed approach.

2. GLOBAL OPTIMIZATION PROCEDURE

In this section, general procedure is described for solving of non-convex optimization problems to global optimality.

2.1 Problem Formulation

We address an optimization problem in following form

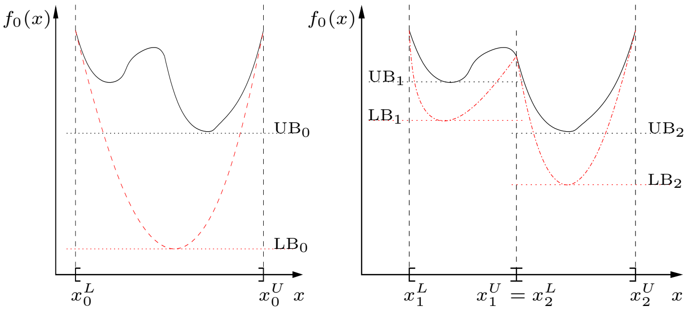
$$\min_{x} \; f_0(x) \tag{1a}$$

$$\text{s.t.} \; f_i(x) \le 0, \qquad i = 1, \dots, n_i \tag{1b}$$

$$f_j(x) = 0, \qquad j = 1, \dots, n_e \tag{1c}$$

$$x \in [x^L, x^U] \tag{1d}$$

where $x \in \mathbb{R}^n$ is a vector of decision variables which values are initially bounded by box constraints (1d). According



(a) node 0

(b) nodes 1&2

Fig. 1. Illustration of branch-and-bound procedure.

to Boyd and Vandenberghe (2004), this is a non-convex optimization problem with n_i inequality and n_e equality constraints, if any of functions in (1a) and (1b) is non-convex, or any of functions in (1c) is not affine. We assume that functions f_k ($k = 0, \ldots, n_i + n_e$) are twice continuously differentiable ($f_k \in \mathbb{C}^2$) and real-valued ($f_k : \mathbb{R}^n \to \mathbb{R}$). Solution to problem (1) provides an upper bound for BB algorithm. Lower bound is found by solving a convex relaxation of (1).

Global minimum of the optimization problem is found employing branch-and-bound framework (Horst and Tuy, 1990). At each branching node, original problem is solved together with its convex relaxation. This is shown in Fig. 1 where first two stages of illustrative BB procedure are shown. In node 0, convex relaxation (red line) of a non-convex problem (black line) is found on a given interval of decision variable x. This interval is then branched creating the first and the second node. Formation of convex relaxations of original problem on this branches follows. It is clear that lower bound in the first node (LB_1) is higher than upper bound in the second node (UB_2) and so the first node is not considered further since global solution does not lie there obviously. The ε-global optimum is found once the UB_i and LB_i meet within specified tolerance ε.

2.2 Convex Relaxation of the Problem

According to Adjiman et al. (1998b), formulation of convex relaxation of original (non-convex) problem requires decomposition of each function f_k ($k = 0, \ldots, n_i + n_e$) in (1) to a sum of linear, convex, special non-convex (univariate concave $\mathrm{UT}(x)$, bilinear $\mathrm{BT}(x)$, ...), and arbitrary non-convex ($\mathrm{NT}(x)$) terms. These terms are then convexly relaxed separately.

Linear terms do not require any convex relaxation since they are already convex. The same applies for convex terms unless they appear in equality constraint functions. In fact, any non-linear equality constraint has to be rewritten such that

$$f_j(x) = 0 \Leftrightarrow \begin{cases} f_j(x) \leq 0 \\ -f_j(x) \leq 0 \end{cases} \quad j = 0, \ldots, n_e \tag{2}$$

For this inequality form, the original decomposition into generic terms should be reconsidered. Hence that if f_j is convex then $-f_j$ is concave and vice versa.

Convex relaxation of univariate concave terms $\mathrm{UT}(x)$ is done by linearization around the lower bound of the variable range. Then, every occurrence of such term is replaced by following expression

$$\mathrm{UT}(x^L) + \frac{\mathrm{UT}(x^U) - \mathrm{UT}(x^L)}{x^U - x^L}(x - x^L) \tag{3}$$

Addition of a relaxation function to the arbitrary non-convex term $\mathrm{NT}(x)$ establishes convex relaxation function for such term in a form

$$\mathrm{NT}(x) + \sum_{i=1}^{n} \alpha_i (x_i^L - x_i)(x_i^U - x_i) \tag{4}$$

where values of α_i's are non-negative scalars found such that

$$\alpha_i \geq \max \left\{ 0, -\frac{1}{2} \min_{x_i} \lambda \left(\nabla_{x_i}^2 \mathrm{NT}(x) \right) \right\} \tag{5}$$

where λ is eigenvalue of Hessian matrix of non-convex term. Another option is just to concentrate on finding the overall valid α which will guarantee convexity of function

$$\mathrm{NT}(x) + \alpha(x^L - x)^T(x^U - x) \tag{6}$$

Then the value of α is computed such that

$$\alpha \geq \max \left\{ 0, -\frac{1}{2} \min_{x} \lambda \left(\nabla_{x}^2 \mathrm{NT}(x) \right) \right\} \tag{7}$$

Problem of minimization of eigenvalue appearing in Eq. (5) and (7) requires solution of non-convex problem in most cases. To avoid this, an interval arithmetic methods can be exploited, e.g. Gerschgorin's theorem for interval matrices (Floudas, 2000). Then, problem of calculation of α_i's values boils down to finding of a minimal eigenvalue of interval family Hessian matrix $[\nabla^2 \mathrm{NT}(x)]$. Interval approaches which can be adopted for this purpose are discussed in Adjiman et al. (1998b) in detail.

As showed in Kearfott (1996), if interval arithmetic operations (multiplication, division, addition, etc.) are composed the interval arithmetic calculations overestimate the range of resulting interval. For example (taken from Kearfott (1996)), if interval function $f(x) = x^2 - x$ over the interval $x = [0, 1]$ is considered, resulting interval calculation is done such that

$$[0, 1]^2 - [0, 1] = [0, 1] - [0, 1] = [-1, 1] \tag{8}$$

This effect is illustrated in Fig. 2 which shows a plot of considered interval function together with its real-valued function equivalent. It can be observed that real-valued function takes values from -0.25 to 0, while its interval extension overestimates this values as it is shown in Eq. (8).

Illustrative Example. Let us consider a simple example to show the effect of the range overestimation of interval arithmetic calculations on convex relaxation of non-convex functions. Here, it is desired to find a convex relaxation of a function $f(x, y) = \cos(x)\sin(y)$ on the interval $x \times y = [-1, 2] \times [-1, 1]$. This convex relaxation is found in form (6). Using of (4) gives the same result. Value of α is found by computing eigenvalues of Hessian matrix

$$\nabla^2 f(x, y) = \begin{pmatrix} -\cos(x)\sin(y) & -\sin(x)\cos(y) \\ -\sin(x)\cos(y) & -\cos(x)\sin(y) \end{pmatrix} \tag{9}$$

Using Gerschgorin's theorem for interval matrices (currently implemented in INTLAB toolbox by Rump (1999)), the value of α was computed according to Eq. (7) to be greater than or equal to 0.92. Exact eigenvalue calculation

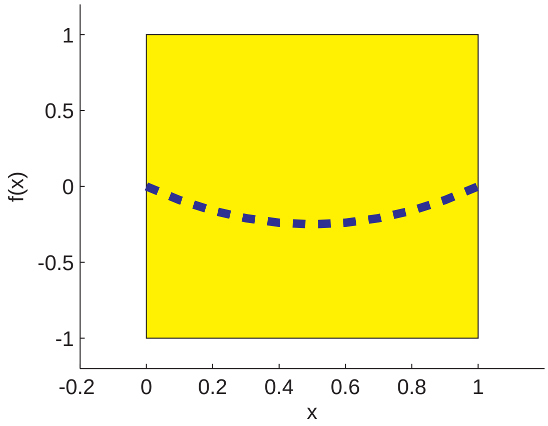
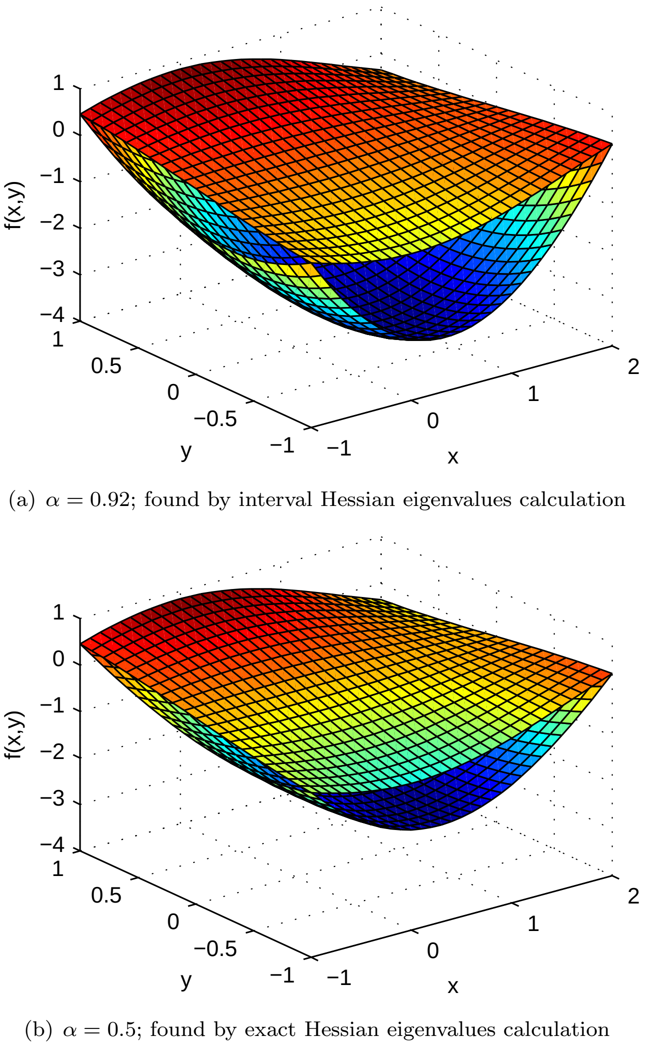


Fig. 2. Real-valued function $f(x)$ (dashed blue line) and its interval function equivalent (yellow box).



(a) $\alpha = 0.92$; found by interval Hessian eigenvalues calculation

(b) $\alpha = 0.5$; found by exact Hessian eigenvalues calculation

Fig. 3. Original function and its convex underestimators.

of Hessian (9) found minimal value of α that guarantee convexity of underestimator (relaxed function) to be 0.5. Fig. 3 compares two convex underestimators obtained by evaluating Eq. (6) with previously computed α values. It is clearly seen that underestimator generated using value of α obtained by an exact Hessian calculation produces convex underestimator tighter almost twice compared to the other one.

Throughout the BB algorithm run, a possibly large number of nodes may appear in BB tree. This happens if loose lower bounds are provided and it results in keeping many nodes where only suboptimal solutions lie. Then, BB algorithm spends a fair amount of time exploring these nodes which is an unwanted feature. It is then straightforward that tighter convex relaxation will result in less iterations needed for a convergence of BB optimization algorithm and less running time of the algorithm as well. In next, we will show how a simple algebraic manipulation can lead to a significant benefit in terms of more efficient algorithm.

3. PROPOSED REFORMULATION PROCEDURE

In previous section we showed how composition of arithmetic operations in interval calculus may result in large overestimation of resulting interval (function). This may be a certain issue if tight convex relaxation functions are wanted to be established for arbitrary non-convex terms. Addition and subtraction operations play just marginal role here since the non-convex term where addition (subtraction) occurs can be rearranged to more non-convex (some possibly convex) terms with no addition (subtraction) operation occurring. Multiplication operations can be decomposed using a simple algebraic transform (Williams, 1993). Suppose that non-convex term in any of functions in (1) is such that $\mathrm{NT}(x) = f_1(x)f_2(x)$. This can be rewritten as

$$f_1(x)f_2(x) = \frac{1}{4}(f_1(x)+f_2(x))^2 - \frac{1}{4}(f_1(x)-f_2(x))^2 \quad (10)$$

Equation (10) can be simplified by considering two new (decision) variables with two equality constraints.

$$f_1(x)f_2(x) = \frac{1}{4}u_1^2 - \frac{1}{4}u_2^2 \quad (11a)$$
$$f_1(x)+f_2(x) = u_1 \quad (11b)$$
$$f_1(x)-f_2(x) = u_2 \quad (11c)$$

Convex relaxation of this rewritten function is now found by the convex relaxation of concave term $-u_2^2$ appearing in Eq. (11a). This is done by a replacement of concave term using (3). Convex relaxation of constraint functions follow in the same manner as described in previous section by rewriting (11b) and (11c) into inequality form and then founding convex relaxations of terms $f_1(x)$, $-f_1(x)$, $f_2(x)$ and $-f_2(x)$. These convex relaxations then produce tighter convex relaxation of problem (1). However, it is needed to provide bounds (box constraints) on new added optimized variables. There are two possibilities. One, is to use interval arithmetic calculations such that u_1^L=min$[f_1(x)+f_2(x)]$ and so on. The second alternative is to consider an optimization problem which minimizes/maximizes u_i having the same constraints as convex relaxation of (1). This approach is similar to variable bound updates approach presented in Adjiman et al. (1998a).

Illustrative Example (Continued) We continue here with illustrative example considered previously. Non-convex term $\cos(x)\sin(y)$ is rewritten to following final form

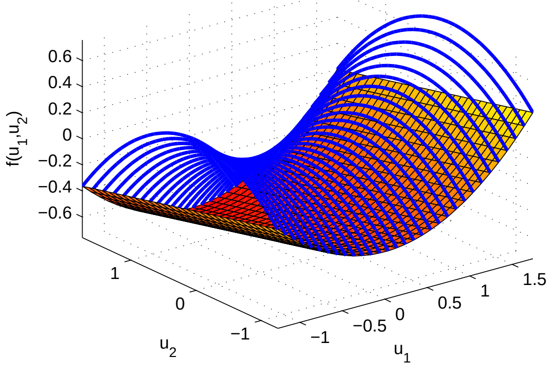


Fig. 4. Function $f(u_1, u_2)$ (depicted in blue lines) and its convex underestimator (red-orange surface).

$$\cos(x)\sin(y) = \frac{1}{4}u_1^2 - \frac{1}{4}u_2^2 \tag{12a}$$

$$\cos(x) + \sin(y) - u_1 \leq 0 \tag{12b}$$

$$-\cos(x) - \sin(y) + u_1 \leq 0 \tag{12c}$$

$$\cos(x) - \sin(y) - u_2 \leq 0 \tag{12d}$$

$$-\cos(x) + \sin(y) + u_2 \leq 0 \tag{12e}$$

Convex relaxation is done by underestimation of terms $-u_2^2$, $\cos(x)$, $-\cos(x)$, $\sin(y)$ and $-\sin(y)$. Box constraints used to bound new added decision variables are found using interval arithmetic calculations. Resulting convex relaxation of the function $f(u_1, u_2)$ is shown in Fig. 4. By transforming this function into original coordinates (by inverting the reformulation) it can be seen that convex relaxation of $\cos(x)\sin(y)$ term using proposed transform is clearly tighter than any of relaxations illustrated in Fig. 3.

4. CASE STUDIES

The global optimization algorithm taken from Papamichail and Adjiman (2004) was implemented in MATLAB 7.11. Solution of NLP problems was found using MATLAB NLP solver *fmincon*. It is an implementation of a general NLP solver, provided by the Optimization Toolbox, uses either a subspace trust region method, based on the interior-reflective Newton method, or a sequential quadratic programming method. The interval calculations needed were performed using INTLAB toolbox by Rump (1999). This toolbox finds the eigenvalues of interval family matrices using Gerschgorin's theorem for interval matrices. All case studies were solved on a workstation with 4.0 GHz Intel® Core™ 2 Duo Processor E8400 with 4GB RAM.

4.1 One dimensional non-convex problem

The first case study considers the problem of minimizing univariate non-convex function over a box constraint of decision variables. This problem was introduced in Pintér (2002) as test problem for GO algorithms. Its objective function is depicted in Fig. 5. Problem takes the form

$$\min_x\ 0.05(x - x_1)^2 + \sin^2(x - x_1) + $$
$$+ \sin^2\left((x - x_1)^2 + (x - x_1)\right) \tag{13a}$$

$$\text{s.t.}\ -10 \leq x \leq 10 \tag{13b}$$

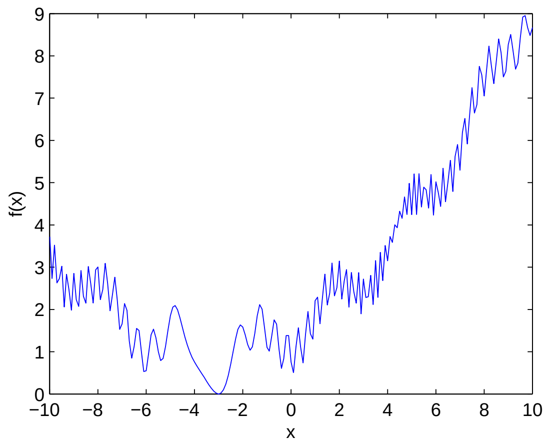


Fig. 5. Plot of the objective function in first case study.

Table 1. Results of the αBB algorithm run with different ranges of box constraints for reformulated problem.

N	No. of iterations	CPU time [s]
1	4	1
10	6	1
100	10	2

Table 2. Results of the αBB algorithm run with different ranges of box constraints for non-reformulated problem.

N	No. of iterations	CPU time [s]
1	41	5
10	497	60
100	8110	1063

where x_1 is a value of minimizer which can be chosen arbitrarily. We choose a minimizer value equal to -3. The first term present in objective function is convex and needs no convex relaxation. The second and the third term are non-convex due to periodicity of a sinus function. Using the proposed procedure, we avoided the squaring of sinus function in both of these terms by introduction of four new decision variables and corresponding eight inequality constraints.

Domain of the problem as it is introduced is arbitrarily set to $[-10, 10]$. In our computations we allow problem domain to be enlarged by a multiplication of box constraint by factor N. If N is equal to 1, originally proposed domain $[-10, 10]$ is considered. When N is set to 10, domain of the problem becomes $[-100, 100]$. This is done to investigate how proposed procedure performs with expanded size of box constraints. Resulting problems are solved to relative global optimality $\varepsilon = 1 \times 10^{-3}$. Results for different values of N are shown in Tab. 1. Comparison with performance of non-reformulated problem αBB algorithm is shown in Tab. 2.

It can be observed that αBB algorithm which exploits reformulation introduced in Section 3 performs significantly better. This feature is most evident if the largest box constraint ($N = 100$) is considered.

Table 3. Results of the αBB algorithm run with different ranges of box constraints for reformulated problem.

N	No. of iterations	CPU time [s]
1	12	16
10	78	90
100	835	929

Table 4. Results of the αBB algorithm run with different ranges of box constraints for non-reformulated problem.

N	No. of iterations	CPU time [s]
1	20	9
10	305	111
100	3083	1243

4.2 Two-dimensional non-convex problem

In this case study, non-convex term $\cos(x)\sin(y)$ appears which was used as an illustrative example for the whole proposed procedure. This optimization problem appears in Adjiman et al. (1996) where it was used as tutorial example to show how the α-based convexification procedure works. The problem is as follows

$$\min_{x,y} \ \cos(x)\sin(y) - \frac{x}{y^2+1} \tag{14a}$$

$$\text{s.t.} \ -1 \le x \le 2 \tag{14b}$$

$$-1 \le y \le 1 \tag{14c}$$

Second non-convex term present in objective function is rewritten in a similar manner to avoid a multiplication between the terms x and $1/(y^2+1)$. By a reformulation procedure, four new decision variables and eight inequality constraints are introduced into a problem. Bounds on new decision variables are found using interval arithmetic calculations. To find out how reformulation procedure performs, we consider not only the original range of decision variables but we multiply the box constraints with factor N equal to 10 and 100. These problems are again solved to relative global optimality $\varepsilon = 1 \times 10^{-3}$. Results are summarized in Tabs. 3 and 4 for the proposed and original approaches, respectively.

This case study again shows that the GO of reformulated problem performs better. However, the overall improvement is not as significant as it is for the first case study. The reduction in number of iterations is satisfying. However, there is a room for improvement if tighter bounds on new added optimized variables are provided. These bounds can be obtained by considering variable bound updates approach (Adjiman et al., 1998a). It can be observed that computational time needed for a single iteration of reformulated problem is almost double than CPU time needed for a single iteration of non-reformulated problem. This can be attributed to a greater size of the reformulated problem.

5. CONCLUSIONS

In this paper, we focused on a problem of finding of a global solution to non-convex non-linear problems. We considered utilization of αBB procedure to solve GO problems in deterministic fashion. Aim of this study was to introduce a simple algebraic reformulation of the non-convex problem to enhance the performance of αBB procedure. Chosen case studies showed that proposed reformulation technique resulted in significant improvement of αBB algorithm. Particularly this was significant if problems were defined over large region of decision variables.

Some issues of this approach were discussed, where one of these is linked to increasing of order of original problem by introducing new optimized variables into it. There is also a possible problem of finding of tight bounds for optimization variables added into reformulated problem. These are the main problems which will be critically addressed in a future work on this promising concept.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09, and the Slovak Research and Development Agency under the project APVV-0029-07. The paper is also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Adjiman, C.S., Androulakis, I.P., and Floudas, C.A. (1998a). A global optimization method, αBB, for general twice-differentiable constrained NLPs - II. Implementation and computational results. *Computers chem. Engng.*, 22(9), 1159–1179.

Adjiman, C.S., Androulakis, I.P., Maranas, C.D., and Floudas, C.A. (1996). A global optimization method, αBB for process design. *Computers chem. Engng.*, 20, 419–424.

Adjiman, C.S., Dallwig, S., Floudas, C.A., and Neumaier, A. (1998b). A global optimization method, αBB, for general twice-differentiable constrained NLPs - I. Theoretical advances. *Computers chem. Engng.*, 22(9), 1137–1158.

Adjiman, C.S. and Floudas, C.A. (1996). Rigorous convex underestimators for general twice-differentiable problems. *J. Glob. Optimization*, 9, 23–40.

Biegler, L.T. (1984). Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers chem. Engng.*, 8(3/4), 243–248.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Chachuat, B., Latifi, M.A., and Roche, N. (2003). A new approach in deterministic global optimisation of problems with ordinary differential equations. In C.A. Floudas and P.M. Pardalos (eds.), *Frontiers in Global Optimization*, 83–108. Kluwer Academic Publishers.

Floudas, C.A. (2000). *Deterministic Global Optimization: Theory, Methods and Applications.* Nonconvex Optimization and Its Applications. Kluwer Academic Publishers.

Floudas, C. and Visweswaran, V. (1990). A global optimization algorithm (GOP) for certain classes of nonconvex NLPs – I theory. *Computers chem. Engng.*, 14, 1397–1417.

Horst, R. and Tuy, H. (1990). *Global optimization. Deterministic approaches (3rd ed.).* Springer Verlag, Berlin, Germany.

Kearfott, R.B. (1996). *Rigorous Global Search: Continuous Problems*, volume 13 of *Nonconvex Optimization and Its Applications.* Kluwer Academic Publishers.

Papamichail, I. and Adjiman, C.S. (2002). A rigorous global optimization algorithm for problems with ordinary differential equations. *J. Glob. Optimization*, 24, 1–33.

Papamichail, I. and Adjiman, C.S. (2004). Global optimization of dynamic systems. *Computers chem. Engng.*, 28, 403–415.

Pintér, J.D. (2002). Global optimization: Software, test problems, and applications. In P.M. Pardalos and H.E. Romeijn (eds.), *Handbook of Global Optimization, Volume 2: Heuristic Approaches*, 515–569. Kluwer Academic Publishers.

Rump, S.M. (1999). INTLAB - INTerval LABoratory. In T. Csendes (ed.), *Developments in Reliable Computing*, 77–104. Kluwer Academic Publishers, Dordrecht.

Singer, A.B. and Barton, P.I. (2001). Global solution of linear dynamic embedded optimization problems. *Journal of Optimization Theory and Applications*, 121(3), 613–646.

Čižniar, M., Podmajerský, M., Hirmajer, T., Fikar, M., and Latifi, M.A. (2009). Global optimization for the parameter estimation of differential-algebraic systems. *Chemical Papers*, 63(3), 274–283.

Williams, H.P. (1993). *Model Building in Mathematical Programming.* John Wiley & Sons, third edition.

Real-time Dynamic Optimisation by Integrated Two-Time-Scale Scheme

Marián Podmajerský * **Miroslav Fikar** *

* *Institute of Information Engineering, Automation and Mathematics,
STU in Bratislava, Slovakia
{marian.podmajersky,miroslav.fikar}@stuba.sk*

Abstract: This paper deals with the problem of uncertainties in optimal control of real process. The measurement-based optimisation is used to treat variations in terminal constraints, model mismatch and process disturbances. It is assumed that this process will be carried out several times in a row and so that run-to-run optimisation can be performed. The paper presents an integrated two-time-scale control where constraints in optimisation problem are adopted between runs and the pre-computed optimal inputs are corrected according to the on-line output measurements during the run. Moreover, the proposed control approach has been implemented to control level transition in two connected tanks with liquid interaction. The results uncover better convergence properties with the resulting control scheme than individual schemes dealing either with run-to-run adaptation or with neighbouring extremal corrections inside the run.

Keywords: dynamic optimisation, neighbouring-extremal control, optimal control, integrated control scheme

1. INTRODUCTION

The processes in general are subject to substantial uncertainty during their operation what lowers quality of the performance and quantity along with operational constraint violations. Common sources of uncertainty include measurement noise, inaccurate model, perturbations in initial conditions, and disturbances during a run-time. Model-based and measurement-based optimisation of dynamic processes are established frameworks that have the ability to mitigate the effect of uncertainty on process performance, especially in the presence of constraints (Kadam and Marquardt, 2007).

In current literature, various optimisation techniques can be found that improve process operation and deal with the influence of uncertainty during the process operation. The most straightforward approach is to implement the optimal inputs obtained as off-line solution of dynamic optimisation problem for the process model and then to track the optimal trajectory on-line. However, in reality, the presence of model mismatch shifts the precomputed optimum what requires on-line corrections of the nominal trajectory in order to ensure optimal operation policy. Different strategy, model predictive control (MPC) (Allgöwer and Zheng, 2000; Garcia et al., 1989; Maciejowski, 2002), implements a re-optimisation of the problem and uses the measurements to update the current state of the model. This strategy suffers two important deficiencies: i) the presence of constraints may result in an infeasible solution; ii) the re-optimisations may not be tractable in real-time. Clearly, the time needed to re-optimise the system depends on both the problem complexity and the computing performance. Long computational time may lead to performance loss, or worse constraint violations, especially in processes with fast process dynamic. Some efficient implementations have been proposed by Biegler (2000); Diehl et al. (2002); Cannon et al. (2001). Explicit MPC approach (Bemporad et al., 2002), multi-parametric programming is used to pre-compute off-line all possible control actions for a given range of the state variables. The control inputs are then adjusted by simply selecting the control law that corresponds to the actual state of the process, as given by the latest measurements. Although this method can accommodate fast sampling times, its foremost limitation comes from the curse of dimensionality and from the quality of the linearisations. This currently limits the application of explicit MPC to problems having no more than a few state variables as well as piece-wise linear dynamics.

In the literature, another control strategy that reduces the online computational effort can be found. Neighbouring extremal (NE) control provides fast suboptimal solutions by not re-solving optimisation problem thus it reduce computational expense. This method was introduced in seventies and eighties by Bryson and Ho (1975); Pesch (1989). In neighbouring extremal control an optimum feedback law is applied to compute control corrections for small variations in state vector. The feedback law is derived around nominal control trajectory obtained from offline solution of dynamic optimisation problem. The optimal control problem and the approximation of the solution for perturbed process are subject to boundary value problem. The derivation of the boundary problem is not straightforward and the good initial guess are required to estimate adjoint variables. Further drawback is a suboptimality of the solution as the feedback law is derived for the small variations in states in the neighbourhood of the nominal solution. Also, the neighbouring extremal control exhibits lower performance when applied to very non-linear processes, e.g. chemical processes. Closely related real-time strategy has been proposed by Kadam and Marquardt (2007) for computation of neighbouring extremal solution using direct optimisation methods. With this strategy neighbouring extremal is computed through sensitivity information of the discretised optimal solution, instead of deriving an optimum feedback law.

This paper presents a two-time-scale approach, whereby a run-to-run adaptation strategy (Bonvin et al., 2006) is implemented at the slow time scale (outer loop) and is integrated with a (constrained) neighbouring extremal controller (Bryson and Ho, 1975) that operates at the fast time scale (inner loop). More specifically, run-to-run adaptation of the terminal constraints (Marchetti et al., 2007) is considered for the outer loop. In its original form, this scheme proceeds by re-optimising the batch operation between each run and adapting the terminal constraints based on the mismatch between their predicted and measured values; but no adaptation is made within a run. In order to reject disturbances within each run and at the same time promote feasibility and optimality, a NE controller is here considered as the inner loop. The NE control which avoids the costly re-optimisation of dynamic systems and approximates the optimal solution of perturbed system, is well-suited for this purpose. The integration between the outer- and inner-loops occurs naturally since the NE controllers are recalculated after each run based on the solution to the outer-loop optimization problem. The resulting integrated two-time-scale optimization scheme thus offers promise to enhance performance and tractability.

The paper is structured as follows. Section 2 defines dynamic optimisation problem and necessary conditions of optimality, Next, Section 3 provides theoretical background on NE control and its numerical computation. Run-to-run constraint optimisation is outlined in Section 4. The proposed integrated two-times-scale optimisation scheme is closely described in Section 5. The performance of the proposed approach is demonstrated on level control of two connected tanks with liquid interaction, in Section 6. Finally, Section 7 concludes the paper.

2. DYNAMIC OPTIMISATION PROBLEM

2.1 Problem Formulation

Throughout the paper, the following dynamic optimisation problem with control and terminal bound constraints is considered:

$$\min_{\boldsymbol{u}} \; J = \phi(\boldsymbol{x}(t_f)) + \int_0^{t_f} L(\boldsymbol{x}(t), \boldsymbol{u}(t))\mathrm{d}t \qquad (1)$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad 0 \le t \le t_f \qquad (2)$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0 \qquad (3)$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f), t_f) \le \boldsymbol{\psi}_{ref} \qquad (4)$$

$$\boldsymbol{u}^L \le \boldsymbol{u}(t) \le \boldsymbol{u}_U. \qquad (5)$$

In (1)–(5), $t \ge 0$ denotes the time variable, with t_f the final time; $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the control vector; $\boldsymbol{x} \in \mathbb{R}^{n_x}$ the state vector, with initial value $\boldsymbol{x}_0$; J, ϕ and L the scalar cost, terminal cost, and integral cost, respectively; and $\boldsymbol{\psi}$ the vector of n_ψ terminal constraints. All the functions in (1)–(5) are assumed to be continuously differentiable with respect to all their arguments.

2.2 Necessary Conditions for Optimality

Following Bryson and Ho (1975), the Hamiltonian function H is defined as follows:

$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U) = L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})^T\boldsymbol{\lambda} + \qquad (6)$$

$$+ \boldsymbol{\mu}^L(\boldsymbol{u}^L - \boldsymbol{u}) + \boldsymbol{\mu}^U(\boldsymbol{u} - \boldsymbol{u}^U), \qquad (7)$$

$\boldsymbol{\lambda} \in \mathbb{R}^{n_x}$ denotes the so-called adjoint (or costate) vector which satisfies

$$\dot{\boldsymbol{\lambda}} = -\boldsymbol{H}_{\boldsymbol{x}} = -\boldsymbol{F}_{\boldsymbol{x}}^T\boldsymbol{\lambda} - \boldsymbol{L}_{\boldsymbol{x}}, \; 0 \le t \le t_f, \qquad (8)$$

with the terminal conditions given by

$$\boldsymbol{\lambda}(t_f) = \left[\boldsymbol{\phi}_{\boldsymbol{x}} + \boldsymbol{\nu}^T\boldsymbol{\psi}_{\boldsymbol{x}}\right]_{t=t_f}, \qquad (9)$$

$\boldsymbol{\mu}^L(t), \boldsymbol{\mu}^U(t) \in \mathbb{R}^{n_u}$ are Lagrange multiplier vector functions satisfying

$$\boldsymbol{\mu}^{L^T}(\boldsymbol{u}^L - \boldsymbol{u}) = \boldsymbol{0}; \quad \boldsymbol{\mu}^L \ge \boldsymbol{0} \qquad (10)$$

$$\boldsymbol{\mu}^{U^T}(\boldsymbol{u} - \boldsymbol{u}^U) = \boldsymbol{0}; \quad \boldsymbol{\mu}^U \ge \boldsymbol{0}, \quad 0 \le t \le t_f. \qquad (11)$$

and $\boldsymbol{\nu} \in \mathbb{R}^{n_\psi}$ are Lagrange multipliers for the terminal constraints such that

$$\boldsymbol{0} = \nu_k\psi_k, \quad \nu_k \ge 0, \quad \text{for each } k = 1, \ldots, n_\psi. \qquad (12)$$

Provided that the optimal control problem is not abnormal, the first- and second- order necessary conditions for optimality (NCO) read:

$$H_{\boldsymbol{u}} = L_{\boldsymbol{u}} + \boldsymbol{F}_{\boldsymbol{u}}^T\boldsymbol{\lambda} - \boldsymbol{\mu}^L - \boldsymbol{\mu}^U = \boldsymbol{0} \qquad (13)$$

$$H_{\boldsymbol{uu}} \ge 0 \qquad (14)$$

This latter determines the set of active terminal constraints at the optimum, which is denoted by the vector $\bar{\boldsymbol{\psi}}$ of dimension $n_{\bar{\psi}}$ and by complementary multiplier $\bar{\boldsymbol{\nu}}^*$. The constraints are inactive when the crresponding Lagrange multiplier is equal to zero. (The subscript such as $_y$ for a given variable denotes partial derivatives of that variable with respect to y.)

3. NEIGHBOURING-EXTREMAL CONTROL

3.1 Neighbouring Control

Let's assume that the optimal control trajectory $\boldsymbol{u}^*(t)$ for the optimisation problem (1)–(5) consists of a sequence of constrained and unconstrained arcs. The optimal solution then comprises $\boldsymbol{x}^*(t)$, $\boldsymbol{\lambda}^*(t), \bar{\boldsymbol{\nu}}^*, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U, 0 \le t \le t_f$. For the control sequence, it is also assumed that the uncertainty is sufficiently small for the perturbed optimal control to have the same sequence of constrained and unconstrained arcs as the nominal solution.

The constrained optimal control problem obtained with a small variation in the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$ and in active terminal constraints $\bar{\boldsymbol{\psi}}(\boldsymbol{x}(t_f), t_f) = \delta\bar{\boldsymbol{\psi}}$ produces variations in optimal control vector $\delta\boldsymbol{u}(t)$, state vector $\delta\boldsymbol{x}(t)$, adjoint vector $\delta\boldsymbol{\lambda}(t)$ and Lagrange multiplier vector $\delta\bar{\boldsymbol{\nu}}$ (for the active terminal constraints $\bar{\boldsymbol{\psi}}$). Along unconstrained arcs, these variations can be calculated from the linearisation of the first-order NCO (10)–(12) around the extremal path (Bryson and Ho, 1975):

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}_{\boldsymbol{x}}^*\delta\boldsymbol{x} + \boldsymbol{F}_{\boldsymbol{u}}^*\delta\boldsymbol{u} \qquad (15)$$

$$\delta\dot{\boldsymbol{\lambda}} = -\boldsymbol{H}_{\boldsymbol{xx}}^*\delta\boldsymbol{x} - \boldsymbol{F}_{\boldsymbol{x}}^{*T}\delta\boldsymbol{\lambda} - \boldsymbol{H}_{\boldsymbol{xu}}^*\delta\boldsymbol{u} \qquad (16)$$

$$\boldsymbol{0} = \boldsymbol{H}_{\boldsymbol{ux}}^*\delta\boldsymbol{x} + \boldsymbol{F}_{\boldsymbol{u}}^{*T}\delta\boldsymbol{\lambda} + \boldsymbol{H}_{\boldsymbol{uu}}^*\delta\boldsymbol{u} \qquad (17)$$

$$\delta\boldsymbol{x}(0) = \delta\boldsymbol{x}_0 \qquad (18)$$

with additional conditions:

$$\delta\boldsymbol{\lambda}(t_f) = \left[\left(\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right)\delta\boldsymbol{x} + \bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\delta\bar{\boldsymbol{\nu}}\right]_{t=t_f} \tag{19}$$

$$\delta\bar{\boldsymbol{\psi}} = \left[\bar{\boldsymbol{\psi}}^*_{\boldsymbol{x}}\delta\boldsymbol{x}\right]_{t=t_f}. \tag{20}$$

A superscript * indicates that the corresponding quantity is evaluated along the extremal path $\boldsymbol{u}^*(t)$, $0 \leq t \leq t_f$, and corresponding states, adjoints and Lagrange multipliers.

Let us assume that the Hamiltonian function is regular, so that $\boldsymbol{H}^*_{\boldsymbol{uu}}$ is invertible along $0 \leq t \leq t_f$. The control variation $\delta\boldsymbol{u}(t)$ for these unconstrained arcs $\boldsymbol{\mu}^L = \boldsymbol{\mu}^U = \boldsymbol{0}$ is then given from (17):

$$\delta\boldsymbol{u}(t) = -(\boldsymbol{H}^*_{\boldsymbol{uu}})^{-1}\left[\boldsymbol{F}^{*T}_{\boldsymbol{u}}\delta\boldsymbol{\lambda}(t) + \boldsymbol{H}^*_{\boldsymbol{ux}}\delta\boldsymbol{x}(t)\right]. \tag{21}$$

Overall, along constrained arcs, the control variation is equal to zero $\delta\boldsymbol{u}(t) = 0$. Then, $\delta\boldsymbol{x}(t)$ and $\delta\boldsymbol{\lambda}(t)$ satisfy the following multi-point boundary value problem (MPBVP):

$$\begin{aligned}
&\begin{pmatrix}\delta\dot{\boldsymbol{x}}(t)\\ \delta\dot{\boldsymbol{\lambda}}(t)\end{pmatrix} = \boldsymbol{\Delta}(t)\begin{pmatrix}\delta\boldsymbol{x}(t)\\ \delta\boldsymbol{\lambda}(t)\end{pmatrix},\\
&\delta\boldsymbol{x}(0) = \delta\boldsymbol{x}_0, \quad \delta\bar{\boldsymbol{\psi}} = \left[\bar{\boldsymbol{\psi}}^*_{\boldsymbol{x}}\delta\boldsymbol{x}\right]_{t=t_f},\\
&\delta\boldsymbol{\lambda}(t_f) = \left[\left(\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right)\delta\boldsymbol{x} + \bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\delta\bar{\boldsymbol{\nu}}\right]_{t=t_f}
\end{aligned} \tag{22}$$

where:

$$\boldsymbol{\Delta}(t) = \begin{cases}\begin{pmatrix}\boldsymbol{\alpha}(t) & -\boldsymbol{\beta}(t)\\ -\gamma(t) & -\boldsymbol{\alpha}(t)^T\end{pmatrix} & \text{along unconstrained arcs}\\[2ex] \begin{pmatrix}\boldsymbol{F}^*_{\boldsymbol{x}} & \boldsymbol{0}\\ -\boldsymbol{H}^*_{\boldsymbol{xx}} & -\boldsymbol{F}^{*T}_{\boldsymbol{x}}\end{pmatrix} & \text{along constrained arcs}\end{cases} \tag{23}$$

and

$$\boldsymbol{\alpha}(t) := \boldsymbol{F}^*_{\boldsymbol{x}} - \boldsymbol{F}^*_{\boldsymbol{u}}(\boldsymbol{H}^*_{\boldsymbol{uu}})^{-1}\boldsymbol{H}^*_{\boldsymbol{ux}} \tag{24}$$

$$\boldsymbol{\beta}(t) := \boldsymbol{F}^*_{\boldsymbol{u}}(\boldsymbol{H}^*_{\boldsymbol{uu}})^{-1}\boldsymbol{F}^{*T}_{\boldsymbol{u}} \tag{25}$$

$$\gamma(t) := \boldsymbol{H}^*_{\boldsymbol{xx}} - \boldsymbol{H}^*_{\boldsymbol{xu}}(\boldsymbol{H}^*_{\boldsymbol{uu}})^{-1}\boldsymbol{H}^*_{\boldsymbol{ux}}. \tag{26}$$

Clearly, at each switching point between an unconstrained and a constrained arcs, a continuity of control, state and adjoint profiles must be preserved. For example, at a switching point between a lower bound and an interior arc, the value of control on lower bound matches the value of control in the interior arc $\boldsymbol{u}^H = \boldsymbol{u}^L$. Here, $\boldsymbol{u}^H$ represents the control obtained from solving the condition $H_{\boldsymbol{u}} = 0$. In addition, state and adjoint trajectories are continuous at this point, too:

$$\boldsymbol{x}^*(t_k^+) = \boldsymbol{x}^*(t_k^-), \quad \boldsymbol{\lambda}^*(t_k^+) = \boldsymbol{\lambda}^*(t_k^-) \tag{27}$$

Variations in switching times are difficult to determine and complicate the calculation of the NE control. To make this implementable, it is considered that the switching points are constant at their nominal times. The control values are then updated only between the fixed times. In practice, performance loss is negligible for small variations of switching times.

3.2 Numerical Computation of Neighbouring Feedback Control

The linear MPBVP (22) can be used to calculate the neighboring-extremal control correction $\delta\boldsymbol{u}(t)$, $0 \leq t \leq t_f$, in either one of two situations:

i. The initial state and (active) terminal constraint variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$ are available at discrete time instants, in which case the discrete feedback control can be obtained by directly re-solving the MPBVP. This can be done via a shooting method as described in Pesch (1989);
ii. The variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$ are available continuously in time, in which case the backward sweep method (Bryson and Ho, 1975) can be used to derive an explicit feedback control law. This approach is closely explained by Bryson and Ho (1975).

In this paper, we consider the first approach that is summarised in Subsection 3.3.

3.3 Shooting Method

The linear TPBVP (22) can be rewritten in the form

$$\begin{pmatrix}\delta\dot{\boldsymbol{x}}(t)\\ \delta\dot{\boldsymbol{\lambda}}(t)\end{pmatrix} = \underbrace{\begin{pmatrix}\boldsymbol{\alpha}(t) & \boldsymbol{\beta}(t)\\ -\gamma(t) & -\boldsymbol{\alpha}(t)\end{pmatrix}}_{=:\,\boldsymbol{\Delta}(t)}\begin{pmatrix}\delta\boldsymbol{x}(t)\\ \delta\boldsymbol{\lambda}(t)\end{pmatrix}, \tag{28}$$

with the boundary conditions

$$\begin{pmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\delta\boldsymbol{x}(0)\\ \delta\boldsymbol{\lambda}(0)\end{pmatrix} + \begin{pmatrix}\boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{B}_1 & \boldsymbol{I}\end{pmatrix}\begin{pmatrix}\delta\boldsymbol{x}(t_f)\\ \delta\boldsymbol{\lambda}(t_f)\end{pmatrix} = \begin{pmatrix}\delta\boldsymbol{x}_0\\ \boldsymbol{B}_2\end{pmatrix}, \tag{29}$$

where

$$\begin{aligned}
\boldsymbol{B}_1 &= -\left[\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t_f}\\
\boldsymbol{B}_2 &= \left[\bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\right]_{t_f}\delta\bar{\boldsymbol{\nu}}
\end{aligned} \tag{30}$$

The shooting approach proceeds by guessing the missing initial (or terminal) conditions in (29), and adjusting them in such a way that the corresponding terminal (or initial) conditions are satisfied (see, e.g., Pesch, 1989). Given the guess $\delta\boldsymbol{\lambda}(0) = \delta\boldsymbol{\lambda}_0$ for the adjoint variations at initial time $t = 0$, the (unique) solution to the linear ODE system (28) is of the form:

$$\begin{pmatrix}\delta\dot{\boldsymbol{x}}(t;\delta\boldsymbol{\lambda}_0)\\ \delta\dot{\boldsymbol{\lambda}}(t;\delta\boldsymbol{\lambda}_0)\end{pmatrix} = \underbrace{\begin{pmatrix}\boldsymbol{\Upsilon}_1(t;0) & \boldsymbol{\Upsilon}_2(t;0)\\ \boldsymbol{\Upsilon}_3(t;0) & \boldsymbol{\Upsilon}_4(t;0)\end{pmatrix}}_{=:\,\boldsymbol{\Upsilon}(t;0)}\begin{pmatrix}\delta\boldsymbol{x}_0\\ \delta\boldsymbol{\lambda}_0\end{pmatrix}, \tag{31}$$

where the transition matrix $\boldsymbol{\Upsilon}(t;0)$ is obtained as the solution to the initial value problem

$$\frac{\partial}{\partial t}\boldsymbol{\Upsilon}(t;0) = \boldsymbol{\Delta}(t)\boldsymbol{\Upsilon}(t;0), \ 0 \leq t \leq t_f; \quad \boldsymbol{\Upsilon}(0;0) = \boldsymbol{I}. \tag{32}$$

Substituting (31) into (29) and (20) leads to the following linear system in the variables $\delta\boldsymbol{\lambda}(0), \delta\bar{\boldsymbol{\nu}}$:

$$\begin{pmatrix}\boldsymbol{Z}_1 & \boldsymbol{Z}_2\\ \boldsymbol{Z}_3 & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\delta\boldsymbol{\lambda}(0)\\ \delta\bar{\boldsymbol{\nu}}\end{pmatrix} = \begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{I}\end{pmatrix}\delta\bar{\boldsymbol{\psi}} - \begin{pmatrix}\boldsymbol{Z}_4\\ \boldsymbol{Z}_5\end{pmatrix}\delta\boldsymbol{x}_0, \tag{33}$$

where

$$\begin{aligned}
\boldsymbol{Z}_1 &= \left[\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t_f}\boldsymbol{\Upsilon}_2(t_f;0) - \boldsymbol{\Upsilon}_4(t_f;0)\\
\boldsymbol{Z}_2 &= \left[\bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\right]_{t_f}\\
\boldsymbol{Z}_3 &= \left[\bar{\boldsymbol{\psi}}^*_{\boldsymbol{x}}\right]_{t_f}\boldsymbol{\Upsilon}_2(t_f;0)\\
\boldsymbol{Z}_4 &= \left[\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t_f}\boldsymbol{\Upsilon}_1(t_f;0) - \boldsymbol{\Upsilon}_3(t_f;0)\\
\boldsymbol{Z}_5 &= \left[\bar{\boldsymbol{\psi}}^*_{\boldsymbol{x}}\right]_{t_f}\boldsymbol{\Upsilon}_1(t_f;0)
\end{aligned} \tag{34}$$

For given initial state and active terminal constraint variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$, the solution to the linear system (33) provides the corresponding initial adjoint and Lagrange multiplier variations $\delta\boldsymbol{\lambda}(0)$ and $\delta\bar{\boldsymbol{\nu}}$. Finally, the NE control variation can be calculated from (21) as

$$\delta\boldsymbol{u}(t) = -(\boldsymbol{H}^*_{\boldsymbol{uu}})^{-1}\left(\boldsymbol{H}^*_{\boldsymbol{ux}}\ \boldsymbol{F}^{*T}_{\boldsymbol{u}}\right)\boldsymbol{\Upsilon}(t;0)\begin{pmatrix}\delta\boldsymbol{x}_0\\ \delta\boldsymbol{\lambda}(0)\end{pmatrix}. \quad (35)$$

4. RUN-TO-RUN CONSTRAINT ADAPTATION

The principle behind run-to-run optimization is similar to MPC. But instead of adapting the initial conditions and moving the control horizon as is done in MPC, the adaptation is performed on the optimization model (e.g., model parameters or constraint biases) before re-running the optimizer. In run-to-run constraint adaptation, more specifically, the terminal constraints (4) in the optimization model are adapted after each run as (Marchetti et al., 2007):

$$\boldsymbol{\psi}(\boldsymbol{x}(t_f), t_f) \leq \delta\boldsymbol{\psi}, \quad (36)$$

where $\delta\boldsymbol{\psi}$ stands for the terminal constraint bias. Such a bias can be directly updated as the difference between the available terminal constraint measurements, $\boldsymbol{\psi}^{\mathrm{meas}}$, at the end of each run and the predicted constraint values. This simple strategy may however lead to excessive correction when operating far away from the optimum, and it may also exacerbate the sensitivity of the adaptation scheme to measurement noise. A better strategy consists of filtering the bias, e.g., with a first-order exponential filter:

$$\delta\boldsymbol{\psi}_{k+1} = [\boldsymbol{I} - \boldsymbol{W}]\,\delta\boldsymbol{\psi}_k + \\ + \boldsymbol{W}\left[\boldsymbol{\psi}^{\mathrm{meas}}_k - \boldsymbol{\psi}(\boldsymbol{x}_k(t_f), t_f)\right], \quad (37)$$

with k the run index, and $\boldsymbol{W}$ a gain matrix—typically, a diagonal matrix.

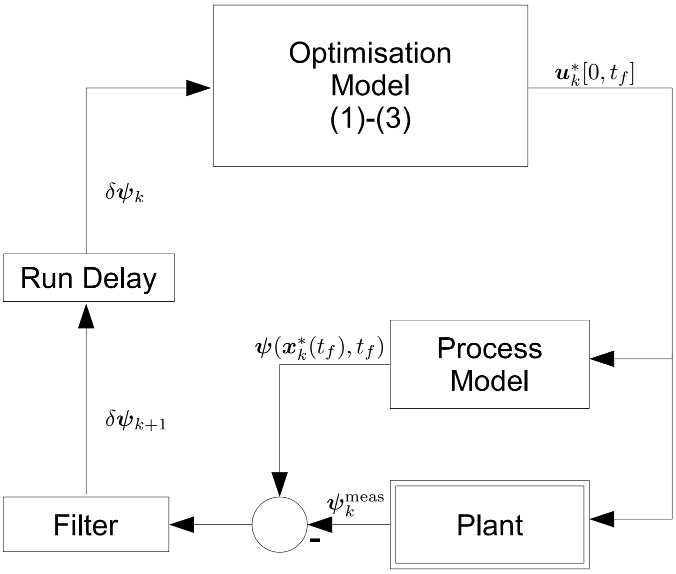

Optimisation Model (1)-(3)
$\boldsymbol{u}_k^*[0,t_f]$
$\delta\boldsymbol{\psi}_k$
Run Delay
$\boldsymbol{\psi}(\boldsymbol{x}_k^*(t_f),t_f)$
Process Model
$\delta\boldsymbol{\psi}_{k+1}$
$\boldsymbol{\psi}_k^{\mathrm{meas}}$
Filter
-
Plant

Fig. 1. Run-to-run constraint adaptation scheme.

The run-to-run constraint-adaptation scheme is shown in Figure 1. The constrained dynamic optimisation problem uses the available nominal process model. It is solved between each run, using any numerical procedure, such as the sequential or the simultaneous approach of dynamic optimisation. The optimal control trajectory $\boldsymbol{u}^*_k(t)$, $0 \leq t \leq t_f$, is computed and applied to the plant during the kth run. The predicted optimal response is denoted by $\boldsymbol{x}^*_k(t)$. The discrepancy between the measured terminal constraint values $\boldsymbol{\psi}^{\mathrm{meas}}_k$ and the optimizer predictions $\boldsymbol{\psi}(\boldsymbol{x}^*_k(t_f), t_f)$ is then used to adjust the constraint bias as described earlier, before re-running the optimizer for the next run.

Of course, optimal control trajectory calculated between runs is suboptimal as the real process is never known perfectly.

5. TWO-TIMES-SCALE OPTIMISATION SCHEME

Run-to-run constraint adaptation was shown to be a promising technology in Marchetti et al. (2007). This approach provides a natural framework for handling changes in active constraints in dynamic process systems and it is quite robust towards model mismatch and process disturbances. Moreover, its implementation is simple. Inherent limitations of this scheme, however, are that (i) it does not perform any control corrections during the runs, and (ii) it typically leads to suboptimal performance.

On the other hand, neighbouring-extremal control is able to correct small deviations around the nominal extremal path in order to deliver similar performance as with re-optimisation. Since no costly on-line re-optimisation is needed, this approach is especially suited for processes with fast dynamics. However, the performance of NE control typically decreases dramatically in the presence of large model mismatch and process disturbances, and it requires a full-state measurement. This leads to sub-optimality or, worse, infeasibility when constraints are present or limited measurements are available.

Our proposal is to combine the advantages of these two approaches: Run-to-run constraint adaptation is applied at a slow time scale (outer loop) to handle large model mismatch and changes in active constraints, based on run-end measurements only. Further, NE control is applied at a fast time scale (inner loop) and uses measurement information available within each run, in order to enhance convergence speed and mitigate sub-optimality. It need to be stated that full-state measurement is required even in case of integrated scheme. The proposed integrated two-time-scale optimization scheme is depicted in Figure 2.

The implementation procedure is as follows:

Initialisation:

(0) Initialise the constraint bias $\delta\boldsymbol{\psi} = \boldsymbol{0}$, select a gain matrix $\boldsymbol{W}$ and set the run index to $k = 1$

Outer Loop:

(1) Determine $\boldsymbol{u}^*_k$ by solving the optimal control problem (1)–(5), then obtain the corresponding states $\boldsymbol{x}^*_k$ and adjoints $\boldsymbol{\lambda}^*_k$, with the active terminal constraints $\bar{\boldsymbol{\psi}}$ and Lagrange multipliers $\bar{\boldsymbol{\nu}}^*_k$, and together with Lagrange multipliers for boundary constraints $\boldsymbol{\mu}^L$ and $\boldsymbol{\mu}^U$ that satisfy NCO (10)–(13).

(2) Design a NE controller around the extremal path $\boldsymbol{u}^*_k$, either by using the backward sweep approach (continuous measurements), or by applying the shooting method (discrete measurements).

(3) **Inner Loop:**
Implement the NE controller during the kth run in order to calculate the corrections $\delta\boldsymbol{u}_k(t)$ to $\boldsymbol{u}^*_k(t)$ based on the available (continuous or discrete) process measurements.

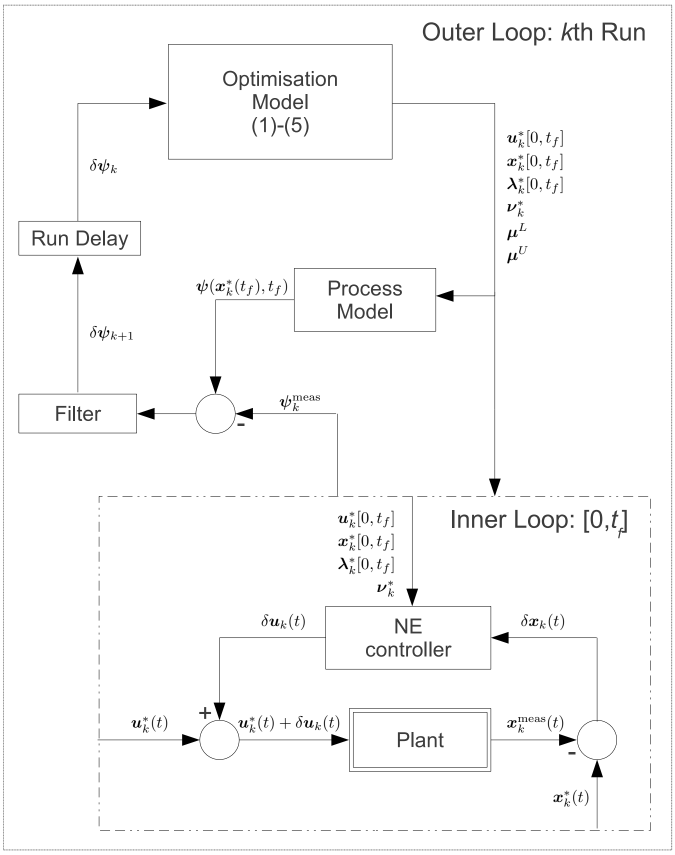

Outer Loop: kth Run
Optimisation Model (1)-(5)
$\delta\boldsymbol{\psi}_k$
Run Delay
$\boldsymbol{u}_k^*[0,t_f]$
$\boldsymbol{x}_k^*[0,t_f]$
$\boldsymbol{\lambda}_k^*[0,t_f]$
$\boldsymbol{\nu}_k^*$
$\boldsymbol{\mu}^L$
$\boldsymbol{\mu}^U$
$\boldsymbol{\psi}(\boldsymbol{x}_k^*(t_f),t_f)$
Process Model
$\delta\boldsymbol{\psi}_{k+1}$
Filter
$\boldsymbol{\psi}_k^{\text{meas}}$
-
Inner Loop: [0,t_f]
$\boldsymbol{u}_k^*[0,t_f]$
$\boldsymbol{x}_k^*[0,t_f]$
$\boldsymbol{\lambda}_k^*[0,t_f]$
$\boldsymbol{\nu}_k^*$
$\delta\boldsymbol{u}_k(t)$
NE controller
$\delta\boldsymbol{x}_k(t)$
$\boldsymbol{u}_k^*(t)$
+
$\boldsymbol{u}_k^*(t)+\delta\boldsymbol{u}_k(t)$
Plant
$\boldsymbol{x}_k^{\text{meas}}(t)$
-
$\boldsymbol{x}_k^*(t)$

Fig. 2. Two-times-scale optimisation scheme employing NE control in the inner loop and run-to-run constraint adaptation in the outer loop.

(4) Update the constraint bias $\delta\boldsymbol{\psi}_{k+1}$ as the filtered difference between the measured values of the terminal constraints and their predicted counterparts.
(5) Increment the run index $k \leftarrow k+1$, and return to Step 1.

6. TWO CONNECTED TANKS WITH LIQUID INTERACTION

The case study compares the performance of the two-time-scale integrated solution with a pure constraint adaptation control scheme and a pure neighbouring-extremal controller. At first, these control methods are tested in simulations and then they are verified in experimental conditions. The process and its model is introduced next.

6.1 Process Description

Level control of two connected tanks with liquid interaction is considered to illustrate integrated two-times-scale approach as can be seen in Figure 4. The experiment has been carried out on Amira DTS200 device (see Figure 3). There are 3 connected tube-shaped tanks connected through the bottoms and six valves to regulate the outflows. The levels are measured by pressure sensors situated at the bottom of tanks. Also, two inlet flows are available: the first pumps liquid to the first tank and the second pumps liquid to the third tank.

For our purposes, only the first two tanks have been used. The objective is to control level transition from an initial level to the terminal level given for the second tank. The manipulated variable is an inlet flow $u(t)$ pumped into the first tank at the top. The levels $h_1(t)$ and $h_2(t)$ are controlled and measured variables. The measurements are provided by pressure sensors. The outflow is situated at the bottom of the second tank and it is regulated by half-opened valve. Also, a liquid interaction take place, as the tanks are connected through the bottoms.

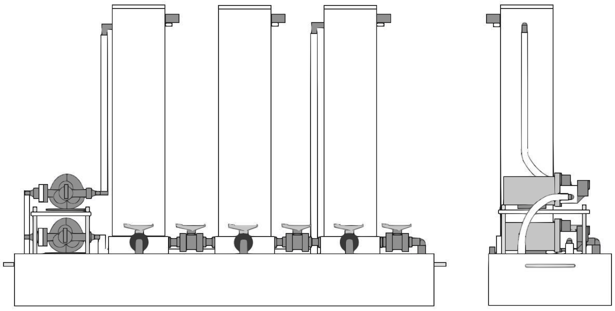


Fig. 3. **Amira DTS200** – Process for level control of tanks.

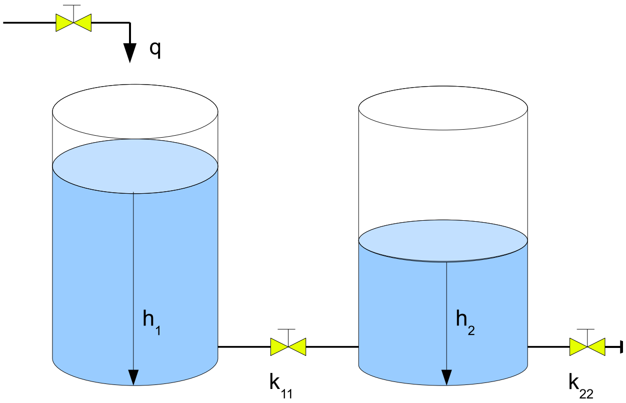


Fig. 4. Configuration of two tanks connected in series.

6.2 Process Model

The model is derived based on the process introduced in the previous subsection. In the model, it is assumed that a liquid density is constant and the walls of tank are vertical to the base. From material balance and from Bernoulli's equation in fluid dynamic, the resulting optimisation problem can be mathematically postulated as follows:

$$\min_u J = \int_{t_0}^{t_f} q(h_2 - h_{2,ref})^2 + ru^2 \mathrm{d}t \tag{38}$$

s.t.

$$\dot{h}_1(t) = \frac{u}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1 - h_2} \tag{39}$$

$$\dot{h}_2(t) = \frac{k_{11}}{F_2}\sqrt{h_1 - h_2} - \frac{k_{22}}{F_2}\sqrt{h_2} \tag{40}$$

$$\tag{41}$$

The state values $h_1(t)$ and $h_2(t)$ are levels [cm] in the first tank and in the second tank, respectively; constants F_1 and F_2 defines cross-sectional area of tank bases [cm^2]; k_{11} and

k_{22} are valve constants [$\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$]; variable u represents inlet flow [$\mathrm{ml\,s}^{-1}$].

The initial level values are determined at constant inlet flow $u = 25\,\mathrm{ml\,s}^{-1}$ as $h_1(t) = 16\,\mathrm{cm}$ and $h_2(t) = 8\,\mathrm{cm}$. The numerical values of constants are: cross-sectional areas $F_1 = 154\,\mathrm{cm}^2$ and $F_2 = 154\,\mathrm{cm}^2$; estimated values of valve constants $k_{11} = 10.68\,\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$ and $k_{22} = 7.5\,\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$. The final time is set to $t_f = 500\,\mathrm{s}$ and inlet flow u is bounded as:

$$0 \le u \le 100\,[\mathrm{ml\,s}^{-1}]. \tag{42}$$

Desired level in the second tank is $h_{2,ref} = 25\,\mathrm{cm}$, so the additional terminal conditions read:

$$h_2(t_f) = h_{2,ref} \tag{43}$$

$$\dot{h}_1(t_f) = 0 \tag{44}$$

$$\dot{h}_2(t_f) = 0. \tag{45}$$

Note that the integral term $\int_{t_0}^{t_f} ru^2 \mathrm{d}t$ augments the original objective function in order to make the control problem non-singular. Weighting variable r is set as low as possible in order to retain the original objective. This way $H_{\boldsymbol{u}}$ depends on the control variable and Hamiltonian H is regular.

6.3 Nominal Solution

Solving the optimisation problem (38)–(45) with the sequential method (Edgar and Himmelblau, 1988; Guntern et al., 1998), the piecewise constant control profile shows the presence of one interior arc and two boundary arcs. Further analysis of this solution indicates that optimal control consists of a upper bound, lower bound, and an interior constant arc. As the problem is regular, the control action along interior arc can be explicitly determined from the necessary conditions of optimality. Note that along boundary arcs, the control action is determined by an upper or a lower bound hence the control variations are simply $\delta\boldsymbol{u} = 0$. The switching times t_1 and t_2 between these arcs are not explicitly known and they need to be estimated, too. The switching times from piecewise constant control profile give good initial guess for these switching times. Overall, the optimal control solution is given as:

(1) $t \in (t_0, t_1)$, the control remains on its upper bound $u^*(t) = 100$
(2) $t \in (t_1, t_2)$, the control remains on its lower bound $u^*(t) = 0$
(3) $t \in (t_2, t_f)$, the control is constant

The optimal control profile is obtained by computing the switching times t_1 and t_2, and the constant control.

6.4 Simulation Results

In order to simulate real behaviour of the process, the valve constants are perturbed to following values: $k_{11} = 10.08\,\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$ and $k_{22} = 8.82\,\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$. The initial conditions remain unchanged. The measured outputs are states with addition of white noise. It is also considered that full-time measurements are available. While the NE controller is designed using the nominal mathematical model, the simulations are performed for measured outputs from the perturbed model. Difference between nominal and perturbed model causes variations which in turn result in a performance loss and terminal constraint violation, when applying the open-loop control profile (see Figure 5). Run-to-run constraint adaptation is initialised with a constraint bias of $\delta\psi = 0$ and considers a filter gain of $W = 0.5$. For proposed integrated control scheme a filter gain was set to $W = 0.4$. The filter parameters were chosen as a compromise between a controller aggressiveness and an ability to deal with measurement noise.

Figure 6 compares the evolution of the performance during the first 15 runs. The evolution of the terminal constraint is presented in the left plot. See that in the first run pure constraint adaptation starts far from the desired value compared to the neighbouring extremal approach. In the consequent runs, the constraints remains inactive with the pure neighbouring extremal approach. In contrast, with pure constraint adaptation and with integrated two-time-scale scheme, the terminal constraint is enhanced over the runs to meet the goal. The pure constraint adaptation approach needed 5 runs to converge. Then, in last 10 runs approach oscillates around the desired value. Note that this approach seems to be more sensitive to measurement noise then the other approaches because only end-point measurement is considered. The integrated scheme starts in close proximity of terminal constraint. In the following runs, this result is slowly enhanced to meet the terminal constraint. Due to the fact that control corrections are applied during each run as well, this approach is able to correct the control profile with lower sensitivity to measurement noise. These corrections affects also the end-point measurement which is less variant over the runs. Note that the sensitivity of NE controller to measurement noise is relative to the chosen number of NE corrections. Lower number expresses lower noise sensitivity but worse corrections and vice versa. In this case study, 120 NE corrections have been chosen.

The right plot of Figure 6 shows the evolution of the modelled terminal constraint hence the original terminal constraint plus constraint bias. This value varies a little for the integrated scheme because the NE controller in the inner loop is able to recover a large portion of optimality loss. In contrast, constraint adaptation requires heavier adaptation since no correction is made during the run.

The resulting control profile after adaptation within 15 runs is shown in left plot in Figure 7. The optimal control profile still consists of the tree arcs, but the switching times have changed compared to nominal solution displayed in Figure 5, as a result of the constrained adaptation. The corresponding measured levels are presented in the right plot in Figure 7. It can be seen that the measured level in second tank met the desired level with proposed integrated two-time-scale control approach.

6.5 Experimental Results

The nominal solution was obtained for certain positions of outflow valves (leakages). In order to test the performance of the control approaches, the outflow was increased. This change also invoked minor variation of initial conditions. Measurements of levels were available on-line, as required. The conversion between measured outputs (in volts) and

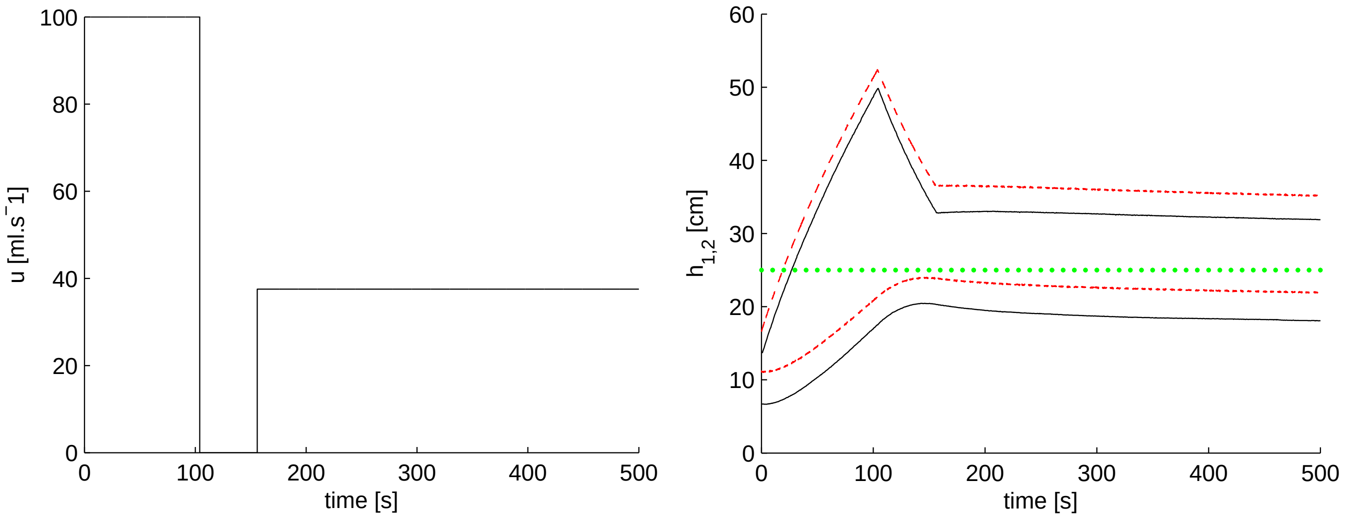


Fig. 5. **Left:** Nominal control trajectory; **Right:** Response for open-loop implementation of nominal control trajectory, **solid line:** nominal model, **dashed line:** perturbed model, and **bold dotted line:** desired level in the second tank.

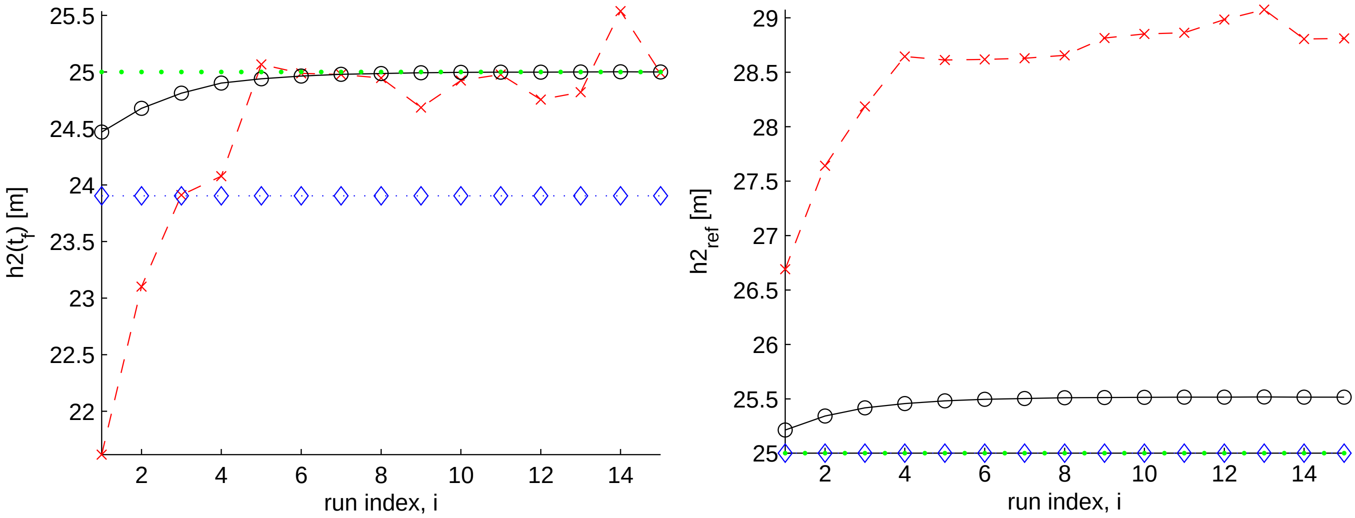


Fig. 6. Control approaches in simulations. **Dashed line with crosses**: pure constraint adaptation, **dotted line with diamonds**: pure neighbouring extremal control, **solid line with circles**: integrated two-time-scale scheme control, **bold dotted line:** desired level. **Left plot**: Evolution of the measured terminal constraint; **Right plot**: Evolution of the modelled terminal constraint.

states (in centimetres) was considered as another perturbation. As in simulations, the difference between model and process is causing performance loss and terminal constraint violation. Run-to-run constraint adaptation is initialised with a constraint bias of $\delta\psi = 0$ and considers a filter gain of $W = 0.6$. For proposed integrated control scheme a filter gain was set to $W = 0.5$. As previously, the filter parameters were chosen with respect to measurement noise and controller dynamics. Sampling time was set to 2 ms as the highest time instant needed to redesign NE controller. In order to reduce the sampling time, the design of NE controller starts an instant before second switching time as there are no control corrections along boundary arc.

Performance evolution over 15 runs of stand-alone approach and proposed two-time-scale approach is displayed in Figure 8. Left plot shows the development of terminal constraint. Similar results as in simulations can be observed. In the first run, NE and proposed two-time-scale approach start closely to the desired value. NE controller is able to recover some of performance loss but not completely. In consequent runs, two-time-scale slowly augments terminal constraint of model to satisfy the objective. In this case, first 3 runs were needed to almost reach the optimum. In last 9 runs, terminal constraint holds closely to desired level value and can be considered as active. The performance of NE does not change as control updates are only carried within the run and not between them as in case of two-time-scale and pure constraint adaptation. In contrast to proposed approach, pure run-to-run adaptation of constraints needed first 6 runs to come close to optimum but with higher value of filter gain. In the following runs, terminal constraint values oscillates around desired level value. This approach is more sensitive to

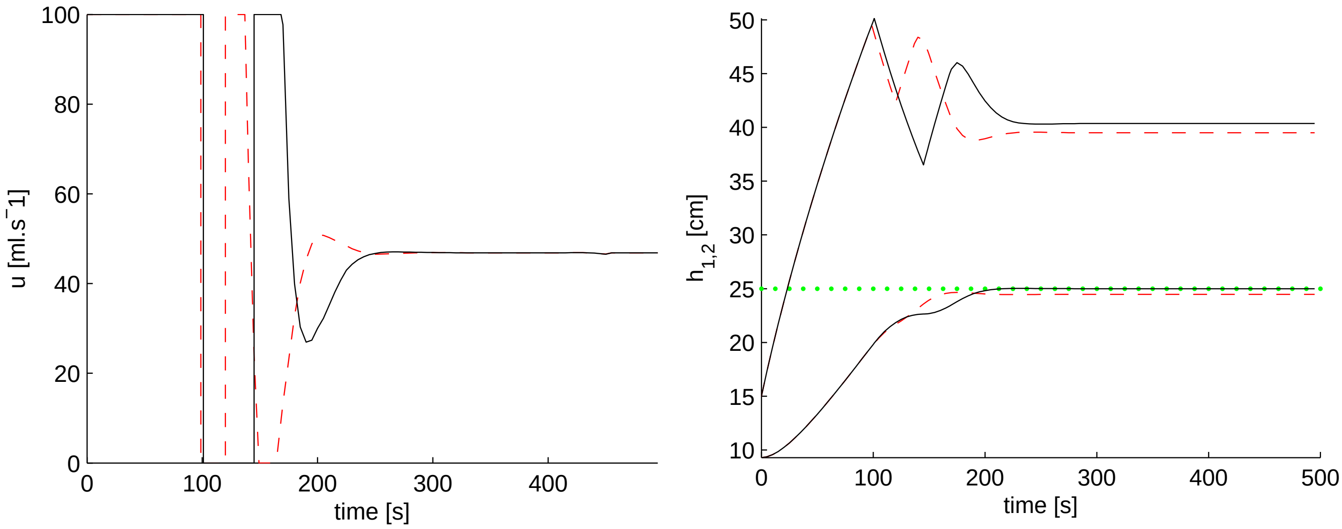


Fig. 7. Performance of proposed scheme in simulations with perturbed valve constants after 1st and 15th run of adaptation. **Left plot**: Control trajectory; **Right plot**: Measured outputs; **Dashed line**: First run; **Solid line**: Last run; **Bold dotted line**: Desired level.

measurement noise in comparison to the other approaches as only final measurements are taken. Proposed two-time-scale approach is able to correct the control profile with lower sensitivity to measurement noise, due to the fact that control corrections are applied during each run as well. This also results in more equal end-point measurements. In experiment, 750 NE corrections were performed. This higher number of NE updates caused high frequency noise in control actions. Thus, these inputs were filtered on-line in order to deliver smoother control actions.

Terminal constraint evolution plus constraint bias is depicted in the right plot of Figure 8. Observe small variations of this value for proposed scheme. In-run control corrections by NE controller provided more equal performance. Since no correction is made during the run, pure constraint adaptation approach clearly requires heavier adaptation.

Left plot in Figure 9 displays the resulting control profile after adaptation within 15 runs. The suboptimal control profile still begins with two boundary arcs but the switching times have changed compared to nominal solution displayed in Figure 5, as a result of the constrained adaptation. The unconstrained arc is no longer constant as is corrected by NE controller. Note that large state variations may cause unproportional control correction what results into clipped control. Right plot in Figure 9 presents corresponding measured levels. Obviously, the measured level in second tank meets the desired level with proposed integrated two-time-scale control approach.

7. CONCLUSIONS

In this paper, an integrated two-times-scale control scheme for level control of two connected tanks has been proposed. Simulation and experimental results show that this control approach improves the performance of dynamic real-time optimisation applied to the real process. The combination of two approaches, namely run-to-run adaptation and neighbouring extremal control, allows to complement the benefits of each other while mitigating some of their deficiencies. Standalone implementation of these approaches indicates lower performance compared to proposed approach. On one side, the NE approach can improve performance loss within the run, and on the other side, constraint adaptation handles terminal constraints. Advantages of the integrated scheme have been demonstrated on the case study for level control of two connected tanks with liquid interaction. As part of future work, an extension of the current scheme to singular control problems is currently under investigation, as well as the ability to handle problems with state path constraints.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of the Scientific Grant Agency of the Slovak Republic under the grant 1/0071/09. Also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

This contribution/publication is the partial result of the R&D Operational Programme for the project Support of Center of Excellence for Smart Technologies, Systems and Services, ITMS 26240120005, co-funded by the ERDF.

REFERENCES

Allgöwer, F. and Zheng, A. (2000). *Nonlinear Model Predictive Control*. Birkhäuser Verlag.

Bemporad, A., Morari, M., Dua, V., and Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1), 3–20.

Biegler, L. (2000). Efficient solution of dynamic optimization and nmpc problems. *Nonlinear Model Predictive Control*, 219–244.

Bonvin, D., Srinivasan, B., and Hunkeler, D. (2006). Control and optimization of batch processes: Improvement of process operation in the production of specialty chemicals. *IEEE Control Systems Magazine*, 26(6), 34–45.

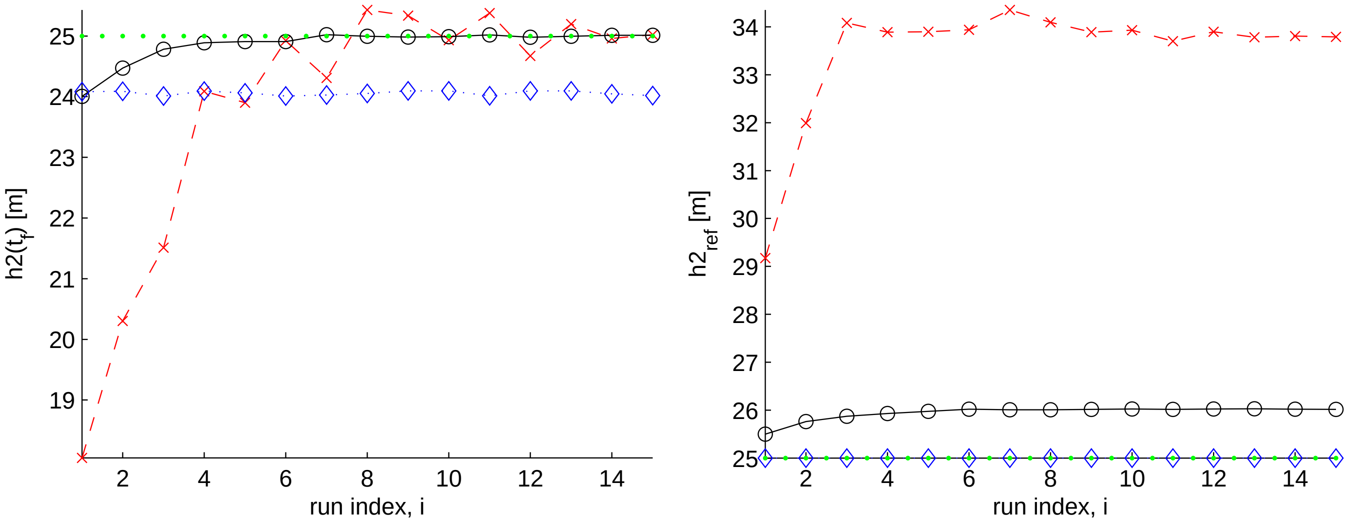


Fig. 8. Control approaches in real conditions. **Dashed line with crosses**: pure constraint adaptation, **dotted line with diamonds**: pure neighbouring extremal control, **solid line with circles**: integrated two-time-scale scheme control, **bold dotted line:** desired level. **Left plot**: Evolution of the measured terminal constraint; **Right plot**: Evolution of the modelled terminal constraint.

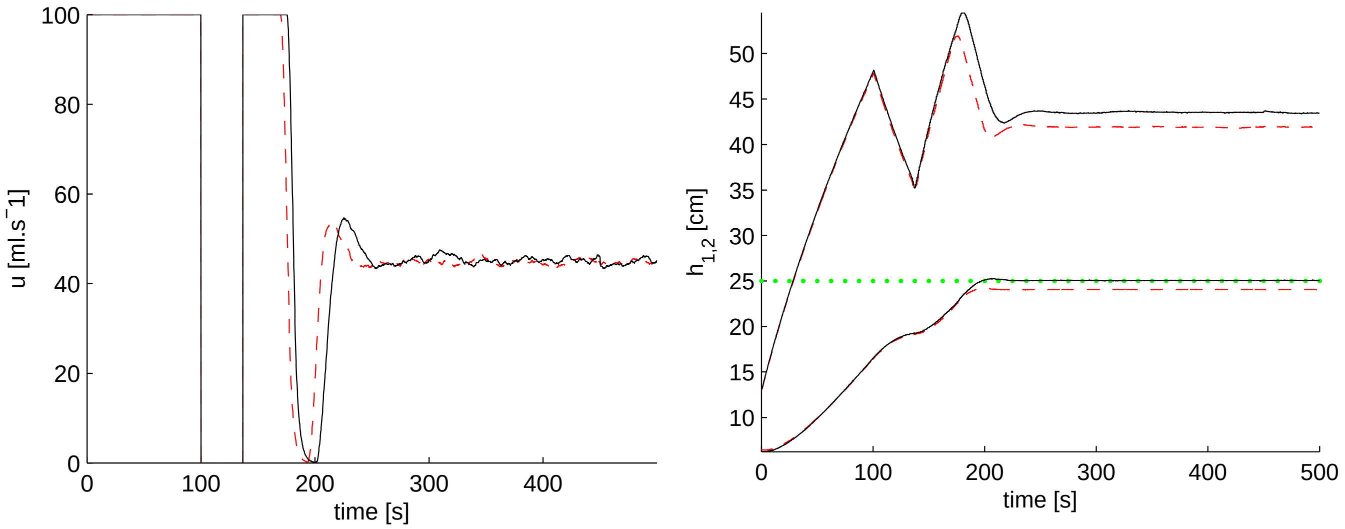


Fig. 9. Experimental performance of proposed scheme after 1st and 15th run of adaptation. **Left plot**: Control trajectory; **Right plot**: Measured outputs; **Dashed line**: First run; **Solid line**: Last run; **Bold dotted line**: Desired level.

Bryson, A.E. and Ho, Y.C. (1975). *Applied Optimal Control – Optimization, Estimation and Control*. Hemisphere publishing corporation.

Cannon, M., Kouvaritakis, B., Lee, Y.I., and brooms, A.C. (2001). Efficient non-linear model based predictive control. *International Journal of Control*.

Diehl, M., Findeisen, R., Nagy, Z., Bock, H.G., Schlöder, J.P., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential algebraic equations. *Jour. of Process Control*.

Edgar, T.F. and Himmelblau, D.M. (1988). *Optimization of Chemical Processes*. McGraw-Hill, New York.

Garcia, C.E., Prett, D.M., and Morari, M. (1989). Model Predictive Control: Theory and Practice – A Survey. *Automatica*, 25(3), 335–348.

Guntern, C., Keller, A., and Hungerbuhler, K. (1998). Economic Optimization of an Industrial Semi-batch Reactor Applying Dynamic Programming. *Industrial and Engineering Chemistry Research*, 37(10), 4017–4022.

Kadam, J.V. and Marquardt, W. (2007). Integration of economical optimization and control for intentionally transient process operation. *Lecture Notes in Control and Information Sciences*, 358, 419–434.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice-Hall, London.

Marchetti, A., Chachuat, B., and Bonvin, D. (2007). Batch process optimization via run-to-run constraints adaptation. In *European Control Conference*. Kos, Greece.

Pesch, H.J. (1989). Real-time computation of feedback controls for constrained optimal control problems. Part II: A correction method based on multiple shooting. *Optimal Control Applications & Methods*, 10, 147–171.

Optimal Control via Initial State of an Infinite Order Time Delay Hyperbolic System

A. Kowalewski *

* *Institute of Automatics*
AGH University of Scince and Technology
al. Mickiewicza 30, 30-059 Cracow, Poland
fax: +48 -12 -6341568, e-mail: ako@ia.agh.edu.pl

Abstract: In this paper, we consider an optimal control problem for a linear infinite order hyperbolic system. One from the initial conditions is given by control function. Sufficient conditions for the existence of a unique solution of such hyperbolic equations with the Dirichlet boundary conditions are presented. The performance functional has the quadratic form. The time horizon T is fixed. Finally, we impose some constraints on the control. Making use of the Lions scheme (Lions (1971)), necessary and sufficient conditions of optimality for the Dirichlet problem with the quadratic performance functional and constrained control are derived.

Keywords: optimal control, infinite order, hyperbolic system, time delay

1. INTRODUCTION

Various optimization problems associated with the optimal control of second order time delay distributed parameter systems have been studied in Wang (1975); Knowles (1978); Kowalewski (1988b 1993ab 1998 2000 2001) respectively.

In Knowles (1978), the time optimal control problems of linear parabolic systems with the Neumann boundary conditions involving constant time delays were considered.

These equations constitute in a linear approximation, a universal mathematical model for many diffusion processes in which time-delayed feedback signals are introduced at the boundary of a system's spatial domain. For example, in the area of plasma control (Wang (1975)), it is of interest to confine a plasma in a given bounded spatial domain Ω by introducing a finite electric potential barrier or a "magnetic mirror" sorrounding Ω. For a collision-dominated plasma, its particle density is describable by a parabolic equation. Due to particle inertia and finiteness of electric potential or the magnetic -mirror field strength, the particle reflection at the domain boundary is not instantaneous. Consequently, the particle flux at the boundary of Ω at any time depends on the flux of particles which escaped earlier and reflected back into Ω at a later time. This leads to the boundary conditions involving time delays.

Using the results of Wang (1975), the existence of a unique solution of such parabolic systems was discussed. A characterization of the optimal control in terms of the adjoint system was given. Consequently, this characterization was used to derive specific properties of the optimal control (bang-bangness, uniqueness, etc.). These results were also extended to certain cases of nonlinear control without convexity and to certain fixed-time problems.

Consequently, in Kowalewski (1988b 1993ab 1998 2000 2001) linear quadratic problems for second order hyperbolic systems with time delays given in the different form (constant time delays, time-varying delays, integral time delays, etc.) were presented.

Moreover, in Lions (1971) and Kowalewski (2004) optimal control problems via initial state for second order hyperbolic systems were investigated.

Such hyperbolic systems constitute in a linear approximation mathematical models of representative convection-reaction processes, e.g. fixed-bed reactors, pressure swing absorption processes, etc.

In particular, in Kowalewski (2010), the optimal control problems via initial condition for infinite order hyperbolic systems were considered. The presented optimal control problem can be generalized onto the case of time delay infinite hyperbolic systems.

For this reason, in the present paper we consider an optimal control problem for a linear time delay infinite order hyperbolic system with constant time delay appearing in the state equation.

We consider a different type of equations, namely, time delay infinite order partial differential equations of hyperbolic type with one from the initial conditions given by control function.

The paper is organized as follows. The existence and uniqueness of solutions for such hyperbolic equations were proved - Lemma 1 and Theorem 2. The optimal control is characterized by the adjoint problem - Lemma 3. The necessary and sufficient conditions of optimality with the quadratic performance functional and constrained control are derived for the Dirichlet problem - Theorem 4.

2. PRELIMINARIES

Let Ω be a bounded open set of R^n with smooth boundary Γ.

We define the infinite order Sobolev space $H^\infty\{a_\alpha,2\}(\Omega)$ of functions $\Phi(x)$ defined on Ω Dubinskij (1975) and Dubinskij (1976) as follows

$$H^\infty\{a_\alpha,2\}(\Omega) = \\ = \left\{ \Phi(x) \in C^\infty(\Omega) : \sum_{|\alpha|=0}^{\infty} a_\alpha \parallel \mathcal{D}^\alpha \Phi \parallel_2^2 < \infty \right\}, \tag{1}$$

where: $C^\infty(\Omega)$ is a space of infinite differentiable functions, $a_\alpha \geq 0$ is a numerical sequence and $\| \cdot \|_2$ is a norm in the space $L^2(\Omega)$, and

$$\mathcal{D}^\alpha = \frac{\partial^{|\alpha|}}{(\partial x_1)^{\alpha_1} \dots (\partial x_n)^{\alpha_n}} , \tag{2}$$

where: $\alpha = (\alpha_1, \dots, \alpha_n)$ is a multi-index for differentiation, $|\alpha| = \sum_{i=1}^{n} \alpha_i$.

The space $H^{-\infty}\{a_\alpha,2\}(\Omega)$ is defined as the formal conjugate space to the space $H^\infty\{a_\alpha,2\}(\Omega)$, namely:

$$H^{-\infty}\{a_\alpha,2\}(\Omega) = \\ = \left\{ \Psi(x) : \Psi(x) = \sum_{|\alpha|=0}^{\infty} (-1)^{|\alpha|} a_\alpha \mathcal{D}^\alpha \Psi_\alpha(x) \right\}, \tag{3}$$

where: $\Psi_\alpha \in L^2(\Omega)$ and $\sum_{|\alpha|=0}^{\infty} a_\alpha \parallel \Psi_\alpha \parallel_2^2 < \infty$.

The duality pairing of the spaces $H^\infty\{a_\alpha,2\}(\Omega)$ and $H^{-\infty}\{a_\alpha,2\}(\Omega)$ is postulated by the formula

$$\langle \Phi, \Psi \rangle = \sum_{|\alpha|=0}^{\infty} a_\alpha \int_\Omega \Psi_\alpha(x) \mathcal{D}^\alpha \Phi(x) \, dx, \tag{4}$$

where: $\Phi \in H^\infty\{a_\alpha,2\}(\Omega)$, $\Psi \in H^{-\infty}\{a_\alpha,2\}(\Omega)$.

From above, $H^\infty\{a_\alpha,2\}(\Omega)$ is everywhere dense in $L^2(\Omega)$ with topological inclusions and $H^{-\infty}\{a_\alpha,2\}(\Omega)$ denotes the topological dual space with respect to $L^2(\Omega)$ so we have the following chain:

$$H^\infty\{a_\alpha,2\}(\Omega) \subseteq L^2(\Omega) \subseteq H^{-\infty}\{a_\alpha,2\}(\Omega). \tag{5}$$

3. EXISTENCE AND UNIQUENESS OF SOLUTIONS

Consider now the distributed-parameter system described by the following infinite order hyperbolic equation

$$\frac{\partial^2 y}{\partial t^2} + Ay + y(x, t-h) = u \quad x \in \Omega, t \in (0,T), \tag{6}$$

$$y(x,t') = \Phi_0(x,t') \quad x \in \Omega, t' \in [-h, 0), \tag{7}$$

$$y(x,0) = 0 \qquad x \in \Omega, \tag{8}$$

$$y'(x,0) = v \qquad x \in \Omega, \tag{9}$$

$$y(x,t) = 0 \qquad x \in \Gamma, t \in (0,T), \tag{10}$$

where Ω has the same properties as in the Section 1.

$$y \equiv y(x,t;v), \qquad u \equiv u(x,t), \qquad v \equiv v(x)$$
$$Q = \Omega \times (0,T), \qquad \bar{Q} = \bar{\Omega} \times [0,T],$$
$$Q_0 = \Omega \times [-h, 0), \qquad \Sigma = \Gamma \times (0,T),$$

h is a specified positive number representing a time delay, Φ_0 is an initial function defined on Q_0.

The operator $\frac{\partial^2}{\partial t^2} + A$ in (6) is an infinite order hyperbolic operator and A (Dubinskij (1986)) is given by

$$Ay = \Big(\sum_{|\alpha|=0}^{\infty} (-1)^{|\alpha|} a_\alpha \mathcal{D}^{2\alpha} + 1 \Big) y \tag{11}$$

and $\sum_{|\alpha|=0}^{\infty} (-1)^{|\alpha|} a_\alpha \mathcal{D}^{2\alpha}$ is an infinite order elliptic partial differential operator.

The operator A is a mapping of $H^\infty\{a_\alpha,2\}$ onto $H^{-\infty}\{a_\alpha,2\}$. For this operator the bilinear form $\Pi(t;y,\varphi) = (Ay,\varphi)_{L^2(\Omega)}$ is coercive on $H^\infty\{a_\alpha,2\}$ i.e. there exists $\lambda > 0, \lambda \in I\!R$ such that $\Pi(t;y,\varphi) \geq \lambda \|y\|^2_{H^\infty\{a_\alpha,2\}}$. We assume that $\forall\ y, \varphi \in H^\infty\{a_\alpha,2\}$ the function $t \rightarrow \Pi(t;y,\varphi)$ is continuously differentiable in $[0,T]$ and $\Pi(t;y,\varphi) = \Pi(t;\varphi,y)$.

The equations (6) - (10) constitute a Dirichlet problem.

First we shall prove sufficient conditions for the existence of a unique solution of the mixed initial-boundary value problem (6) - (10) for the case where $v \in L^2(\Omega)$.

The existence of a unique solution for the mixwd initial-boundary value problem (6) - (10) on the cylinder Q can be proved using a constructive method, i.e. first solving (6) - (10) on the subcylinder Q_1 and in turn on Q_2, etc., until the procedure covers the whole cylinder Q. In this way the solution in the previous step determines the next one.

For simplicity, we introduce the notations

$$E_j \overset{\wedge}{=} ((j-1)h, jh), \quad Q_j = \Omega \times E_j,\ j = 1, ..., K$$

Using the results of Section 6 of (Lions (1971), p. 314) we can prove the following lemma.

Lemma 1. Let

$$u \in L^2(Q), \tag{12}$$

$$f_j \in L^2(Q_j), \tag{13}$$

where

$$f_j(x,t) = u(x,t) - y_{j-1}(x, t-h),$$

$$y_{j-1}(\cdot, (j-1)h) \in H^\infty\{a_\alpha,2\}(\Omega), \tag{14}$$

$$y'_{j-1}(\cdot, (j-1)h) \in L^2(\Omega). \tag{15}$$

Then, there exists a unique solution $y_j \in L^2(E_j; H^\infty\{a_\alpha,2\}(\Omega))$ with $\frac{dy_j}{dt} \in L^2(E_j; L^2(\Omega))$ for the mixed initial-boundary value problem (6), (14) and (15).

Proof. We observe that for $j = 1$ we have $y_{j-1}\big|_{Q_0}(x, t-h) = \Phi_0(x, t-h)$. Then the assumptions

(13), (14) and (15) are fullfilled if we assume that $\Phi_0 \in L^2(-h, 0; H^\infty\{a_\alpha, 2\}(\Omega))$, $\frac{d\Phi_0}{dt} \in L^2(-h, 0; L^2(\Omega))$, $y_0(x, 0) \in H^\infty\{a_\alpha, 2\}(\Omega)$ and $y_0'(x, 0) \in L^2(\Omega)$. These assumptions are sufficient to ensure the existence of a unique solution $y_1 \in L^2(E_1; H^\infty\{a_\alpha, 2\}(\Omega))$ with $\frac{dy_1}{dt} \in L^2(E_1; L^2(\Omega))$.

In order to extend the results to Q_2, we have to to prove that $y_1(\cdot, h) \in H^\infty\{a_\alpha, 2\}(\Omega)$, $y_1'(\cdot, h) \in L^2(\Omega)$ and $f_2 \in L^2(Q_2)$. From the Theorem 3.1 of Lions and Magenes (1972) (Vol.1, p.19) $y_1 \in L^2(E_1; H^\infty\{a_\alpha, 2\}(\Omega))$ jointly with $\frac{dy_1}{dt} \in L^2(E_1; L^2(\Omega))$ imply that the mappings $t \to y_1(\cdot, t)$ and $t \to y_1'(\cdot, t)$ are continuous from $[0, h] \to H^\infty\{a_\alpha, 2\}(\Omega)$ and $[0, h] \to L^2(\Omega)$ respectively. Thus, $y_1(\cdot, h) \in H^\infty\{a_\alpha, 2\}(\Omega)$ and $y_1'(\cdot, h) \in L^2(\Omega)$.

Also it is easy to notice that the assumption (13) follows from the fact that $y_1 \in L^2(E_1; H^\infty\{a_\alpha, 2\}(\Omega))$ and $u \in L^2(Q)$. Thus, there exists a unique solution $y_2 \in L^2(E_2; H^\infty\{a_\alpha, 2\}(\Omega))$ with $\frac{dy_2}{dt} \in L^2(E_2; L^2(\Omega))$.□

The foregoing result is now summarized for $j = 1, ..., K$.

Theorem 2. Let $y(x, 0)$, $y'(x, 0), \Phi_0$ and u be given with $y(\cdot, 0) \in H^\infty\{a_\alpha, 2\}(\Omega)$, $y'(\cdot, 0) \in L^2(\Omega)$, $\Phi_0 \in L^2(-h, 0; H^\infty\{a_\alpha, 2\})$, $\frac{d\Phi_0}{dt} \in L^2(-h, 0; L^2(\Omega))$ and $u \in L^2(Q)$. Then, there exists a unique solution $y \in L^2(0, T; H^\infty\{a_\alpha, 2\}(\Omega))$ with $\frac{dy}{dt} \in L^2(0, T; L^2(\Omega))$ for the mixed initial-boundary value problem (6) - (10).

4. PROBLEM FORMULATION. OPTIMIZATION THEOREM.

We shall now formulate the optimal control problem for the Dirichlet problem. Let us denote by $U = L^2(\Omega)$ the space of controls. The time horizon T is fixed in our problem.

The performance functional is given by

$$I(v) = \lambda_1 \int\limits_\Omega | y(x, T; v) - z_d |^2 \, dx + \\ + \lambda_2 \int\limits_\Omega (Nv)v \, dx, \qquad (16)$$

where: $\lambda_i \geq 0$ and $\lambda_1 + \lambda_2 > 0; z_d$ is a given element in $L^2(\Omega)$; N is a positive linear operator on $L^2(\Omega)$ into $L^2(\Omega)$.

Finally, we assume the following constraint on controls $v \in U_{ad}$, where

$$U_{ad} \text{ is a closed, convex subset of } U. \qquad (17)$$

Let $y(x, t; v)$ denote the solution of the mixed initial-boundary value problem (6) - (10) at (x, t) corresponding to a given control $v \in U_{ad}$. We note from the Theorem 2 that for any $v \in U_{ad}$ the performance functional (16) is well-defined since $y(x, T; v) \in H^\infty\{a_\alpha, 2\}(\Omega) \in L^2(\Omega)$. The solving of the formulated optimal control problem is equivalent to seeking a $v_0 \in U_{ad}$ such that $I(v_0) \leq I(v)$ $\forall v \in U_{ad}$.

Then from the Theorem 1.3 (Lions (1971), p. 10) it follows that for $\lambda_2 > 0$ a unique optimal control v_0 exists; moreover, v_0 is characterized by the following condition

$$I'(v_0) \cdot (v - v_0) \geq 0 \qquad \forall v \in U_{ad}. \qquad (18)$$

Using the form of the performance functional (16) we can express (18) in the following form

$$\lambda_1 \int\limits_\Omega (y(x, T; v_0) - z_d)(y(x, T; v) - y(x, T; v_0))dx + \\ + \lambda_2 \int\limits_\Omega (Nv_0)(v - v_0)dx \geq 0 \quad \forall v \in U_{ad}. \qquad (19)$$

To simplify (19), we introduce the adjoint equation and for every $v \in U_{ad}$, we define the adjoint variable $p = p(v) = p(x, t; v)$ as the solution of the equation

$$\frac{\partial^2 p(v)}{\partial t^2} + Ap(v) + p(x, t + h; v) = 0 \\ x \in \Omega, \ t \in (0, T - h), \qquad (20)$$

$$\frac{\partial^2 p(v)}{\partial t^2} + Ap(v) = 0 \quad x \in \Omega, t \in (T - h, T), \qquad (21)$$

$$p(x, T; v) = 0 \qquad x \in \Omega, \qquad (22)$$

$$p'(x, T; v) = -\lambda_1(y(x, T; v) - z_d) \quad x \in \Omega, \qquad (23)$$

$$p(x, t) = 0 \quad x \in \Gamma, \ t \in (0, T). \qquad (24)$$

The existence of a unique solution for the problem (20)-(24) on the cylinder Q can be proved using a constructive method. It is easy to notice that for given z_d and v, problem (20)-(24) can be solved backwards in time starting from $t = T$, i.e., first, solving (20)-(24) on the subcylinder Q_K and in turn on Q_{K-1}, etc. until the procedure covers the whole cylinder Q. For this purpose, we may apply Theorem 2 (with an obvious change of variables) to problem (20)-(24).

Lemma 3. Let the hypothesis of Theorem 2 be satisfied. Then, for given $z_d \in L^2(\Omega)$ and any $v \in L^2(\Omega)$, there exists a unique solution such that $p(v) \in L^2(0, T; H^\infty\{a_\alpha, 2\}(\Omega))$ and $\frac{\partial p(v)}{\partial t} \in L^2(0, T; L^2(\Omega))$ for the problem (20)-(24).

We simplify (19) using the adjoint equation (20)-(24). For this purpose setting $v = v_0$ in (20)-(24), multiplying both sides of (20)-(21) by $(y(v) - y(v_0))$ and then integrating over $\Omega \times (0, T - h)$ and $\Omega \times (T - h, T)$ respectively and then adding both sides of (20), (21) we get

$$\int\limits_Q \left(\frac{\partial^2 p(v_0)}{\partial t^2} + Ap(v_0) \right) (y(v) - y(v_0)) \, dxdt \\ + \int\limits_0^{T-h} \int\limits_\Omega p(x, t + h; v_0)(y(x, t; v) - y(x, t; v_0))dxdt \\ = \int\limits_\Omega p'(x, T, v_0)[y(x, T; v) - y(x, T; v_0)]dx$$

$$\begin{aligned}
&- \int\limits_\Omega p'(x,0,v_0)[y(x,0;v) - y(x,0;v_0)]dx \\
&- \int\limits_\Omega p(x,T,v_0)[y'(x,T;v) - y'(x,T;v_0)]dx \\
&+ \int\limits_\Omega p(x,0,v_0)[y'(x,0;v) - y'(x,0;v_0)]dx \\
&+ \int\limits_Q p(v_0)\frac{\partial^2}{\partial t^2}(y(v) - y(v_0))\ dxdt \\
&+ \int\limits_Q Ap(v_0)(y(v) - y(v_0))\ dxdt \\
&+ \int\limits_0^{T-h}\int\limits_\Omega p(x,t+h;v_0)\cdot \\
&\qquad \cdot(y(x,t;v) - y(x,t;v_0))dxdt = 0 \qquad (25)
\end{aligned}$$

and so, by (22) and (23),

$$\begin{aligned}
&\lambda_1 \int\limits_\Omega (y(x,T,v) - z_d)[y(x,T;v) - y(x,T;v_0)]dx \\
&= \int\limits_\Omega p(x,0,v_0)[y'(x,0;v) - y'(x,0;v_0)]dx \\
&+ \int\limits_Q p(v_0)\frac{\partial^2}{\partial t^2}(y(v) - y(v_0))\ dxdt \\
&+ \int\limits_Q Ap(v_0)(y(v) - y(v_0))dxdt \\
&+ \int\limits_0^{T-h}\int\limits_\Omega p(x,t+h;v_0)\cdot \\
&\qquad \cdot(y(x,t;v) - y(x,t;v_0))dxdt \qquad (26)
\end{aligned}$$

Using the equation (6), the second integral on the right-hand side of (26) can be rewritten as

$$\begin{aligned}
&\int\limits_Q p(v_0)\frac{\partial^2}{\partial t^2}(y(v) - y(v_0))dxdt \\
&= - \int\limits_Q p(v_0)A(y(v) - y(v_0))dxdt \\
&- \int\limits_0^{T}\int\limits_\Omega p(x,t;v_0)\cdot \\
&\qquad \cdot(y(x,t-h;v) - y(x,t-h;v_0))dxdt \\
&= - \int\limits_Q p(v_0)A(y(v) - y(v_0))dxdt \\
&- \int\limits_{-h}^{T-h}\int\limits_\Omega p(x,t'+h;v_0)\cdot \\
&\qquad \cdot(y(x,t';v) - y(x,t';v_0))dxdt' \qquad (27)
\end{aligned}$$

Substituting (27) into (26) we obtain

$$\begin{aligned}
&\lambda_1 \int\limits_\Omega (y(x,T;v_0) - z_d)(y(x,T;v) - y(x,T;v_0))dx \\
&= \int\limits_\Omega p(x,0,v_0)[y'(x,0;v) - y'(x,0;v_0)]dx \\
&- \int\limits_Q p(v_0)A(y(v) - y(v_0))dxdt \\
&- \int\limits_{-h}^{0}\int\limits_\Omega p(x,t+h;v_0)(y(x,t;v) - y(x,t;v_0))dxdt \\
&- \int\limits_0^{T-h}\int\limits_\Omega p(x,t+h;v_0)(y(x,t;v) - y(x,t;v_0))dxdt \\
&+ \int\limits_Q p(v_0)A(y(v) - y(v_0))dxdt \\
&+ \int\limits_0^{T-h}\int\limits_\Omega p(x,t+h;v_0)\cdot \\
&\qquad \cdot(y(x,t;v) - y(x,t;v_0))dxdt \qquad (28)
\end{aligned}$$

Afterwards using the formulae $y'(x,0;v) = v$ and $y'(x,0;v_0) = v_0$ in (28) we get

$$\begin{aligned}
&\lambda_1 \int\limits_\Omega (y(x,T;v_0) - z_d)(y(x,T;v) - y(x,T;v_0))dx \\
&\qquad = \int\limits_\Omega p(x,0;v_0)(v - v_0)dx \qquad (29)
\end{aligned}$$

Substituting (29) into (19) we obtain

$$\int\limits_\Omega (p(x,0;v_0) + \lambda_2 N v_0)(v - v_0)dx \geq 0 \qquad \forall v \in U_{ad} \qquad (30)$$

Theorem 4. For the problem (6)-(10) with the performance functional (16) with $z_d \in L^2(\Omega)$ and $\lambda_2 > 0$ and with constraints on controls (17), there exists a unique optimal control v_0 which satisfies the maximum condition (30).

We must notice that the conditions of optimality derived above (Theorem 4) allow us to obtain an analytical formula for the optimal control in particular cases only (e.g. there are no constraints on controls). This results from the following: the determining of the function $p(v_0)$ in the maximum condition from the adjoint equation is possible if and only if we know y_0 which corresponds to the control v_0. These mutual connections make the practical use of the derived optimization formulas difficult. Therefore we resign from the exact determining of the optimal control and we use approximation methods.

In the case of the performance functional (16) with $\lambda_1 > 0$ and $\lambda_2 = 0$, the optimal control problem reduces to the minimizing of the functional on a closed and convex subset in a Hilbert space. Then, the optimization problem is equivalent to a quadratic programming one (Kowalewski (1988a)) which can be solved by the use of the well-known algorithms, e.g. Gilbert's in Kowalewski (1988a).

5. CONCLUSIONS

The results presented in the paper can be treated as a generalization of the results obtained in Kowalewski (2010) onto the case of infinite order time delay hyperbolic systems with one of the initial conditions given by control function.

In this paper we have considered optimal control problem for such hyperbolic systems with the Dirichlet boundary conditions.

We can also consider similar optimal control problems for time delay infinite order hyperbolic systems with Neumann boundary conditions.

Finally we can consider optimal control problem for infinite order hyperbolic systems with two initial conditions given by control functions.

The ideas mentioned above will be developed in forthcoming papers.

ACKNOWLEDGEMENTS

The research presented here was carried out within the research programme of the AGH University of Science and Technology, No. 11.11.120.768.

REFERENCES

Dubinskij, J.A. (1975). Sobolev spaces of infinite order and behavior of solution of some boundary value problems with unbounded increase of the order of the equation. *Matiematiczeskii Sbornik*, 98, 163–184.

Dubinskij, J.A. (1976). Non-trivality of Sobolev spaces of infinite order for a full Euclidean space and a Tour's. *Matiematiczeskii Sbornik*, 100, 436–446.

Dubinskij, J.A. (1986). *Sobolev Spaces of Infinite Order and Differential Equations. Teubner-Texte zur Mathematik - volume 87.* Teubner-Verlag, Leipzig.

Knowles, G. (1978). Time optimal control of parabolic systems with boundary conditions involving time delays. *Journal of Optimization Theory and Applications*, 25(4), 563–574.

Kowalewski, A. (1988a). Boundary control of distributed parabolic system with boundary condition involving a time-varying lag. *International Journal of Control*, 48, 2233–2248.

Kowalewski, A. (1988b). Boundary control of hyperbolic system with boundary condition involving a time-delay. *Lecture Notes in Control and Information Sciences*, 111, 507–518.

Kowalewski, A. (1993a). Boundary control of a hyperbolic system with time lags. *IMA Journal of Mathematical Control and Information*, 10, 261–272.

Kowalewski, A. (1993b). Optimal control of hyperbolic system with time lags. *Journal of Applied Mathematics and Computer Science*, 3, 687–697.

Kowalewski, A. (1998). Optimal control of distributed hyperbolic systems with multiple time-varying lags. *International Journal of Control*, 71, 419–435.

Kowalewski, A. (2000). Optimal control of distributed hyperbolic systems with deviating arguments. *International Journal of Control*, 73, 1026–1041.

Kowalewski, A. (2001). *Optimal Control of Infinite Dimensional Distributed Parameter Systems with Delays.* AGH University of Science and Technology Press, Cracow.

Kowalewski, A. (2004). Optimal control via initial state of a time delay hyperbolic system. In *Proceedings of the 10th IEEE International Conference on Methods and Models in Automation and Robotics, 30 August - 2 September 2004, Międzyzdroje, Poland*, 565–568.

Kowalewski, A. (2010). Optimal control via initial state of an infinite order hyperbolic system. In *Proceedings of the 15th IEEE International Conference on Methods and Models in Automation and Robotics, 12-26 August 2010, Międzyzdroje, Poland*, 275–277.

Lions, J. (1971). *Optimal Control of Systems Governed by Partial Differential Equations.* Springer-Verlag, Berlin-Heidelberg.

Lions, J. and Magenes, E. (1972). *Non-Homogeneous Boundary Value Problems and Applications - Vols. 1 and 2.* Springer-Verlag, Berlin-Heidelberg.

Wang, P.K.C. (1975). Optimal control of parabolic systems with boundary conditions involving time delays. *SIAM Journal of Control*, 13(2), 274–293.

Digital Self-tuning Smith Predictor Based on Pole Assignment Approach

V. Bobál* P. Chalupa* P. Dostál* M. Brázdil*

*Tomas Bata University in Zlín, Faculty of Applied Informatics, Department of Process Control, Nam. T. G. Masaryka 5555, 760 05 Zlín 5, Czech Republic (Tel.: +420 603 57 4197; e-mail: bobal@fai.utb.cz)

Abstract: Time-delays (dead times) are found in many processes in industry. Time-delays are mainly caused by the time required to transport mass, energy or information, but they can also be caused by processing time or accumulation. The contribution is focused on a design of algorithms for self-tuning digital control for processes with time-delay. The algorithms are based on the some modifications of the Smith Predictor (SP). One modification of the SP based on the digital PID controller was applied and it was compared with new designed modification based on polynomial (pole assignment) approach. The program system MATLAB/SIMULINK was used for simulation verification of these algorithms.

1. INTRODUCTION

Time-delays appear in many processes in industry and other fields, including economical and biological systems (see Normey-Rico and Camacho 2007). They are caused by some of the following phenomena:

- the time needed to transport mass, energy or information,
- the accumulation of time lags in a great numbers of low order systems connected in series,
- the required processing time for sensors, such as analyzers; controllers that need some time to implement a complicated control algorithms or process.

Consider a continuous time dynamical linear SISO (single input $u(t)$ – single output $y(t)$) system with time-delay T_d.

The transfer function of a pure transportation lag is $e^{-T_d s}$, where s is complex variable. Overall transfer function with time-delay is in the form

$$G_d(s) = G(s)e^{-T_d s} \tag{1}$$

where $G(s)$ is the transfer function without time-delay.

Processes with significant time-delay are difficult to control using standard feedback controllers mainly because of the following:

- the effect of the disturbances is not felt until a considerable time has elapsed;
- the effect of the control action requires some time to elapse;
- the control action that is applied based on the actual error tries to correct a situation that originated some time before.

The problem of controlling time-delay processes can be solved by some control methods using

- PID controllers;
- time-delay compensators;
- model predictive control techniques.

It is clear that many processes in industry are controlled by the PID controllers. When the process contains a time-delay, the tuning of the PID controller is difficult. The most popular tuning rules for processes with small time-delay were proposed by Ziegler and Nichols 1942. Several methods for new tuning rules were proposed for stable and unstable processes with time-delay. A presentation and review of some these methods are introduced in (Åström and Hägglund 1995).

When a high performance of the control process is desired or the relative time-delay is very large, a predictive control strategy must be used. The predictive control strategy includes a model of the process in the structure of the controller. The first time-delay compensation algorithm was proposed by (Smith 1957). This control algorithm known as the Smith Predictor (SP) contained a dynamic model of the time-delay process and it can be considered as the first model predictive algorithm.

Historically first modifications of time-delay algorithms were proposed for continuous-time (analogue) controllers. On the score of implementation problems, only the discrete versions are used in practice in this time.

The majority of processes met in the industrial practice have stochastic characteristics and eventually they embody nonlinear behaviour. Traditional controllers with fixed parameters are often unsuitable for such processes because their parameters change. One possible alternative for improving the quality of control for such processes is the use of adaptive control systems. Different approaches were proposed and utilized. One successful approach is represented by self-tuning controller (STC). The main idea of an STC is based on the combination of a recursive identification procedure and a selected controller synthesis. Some STC modifications of the digital Smith Predictors are designed and verified by simulation in this paper.

The paper is organized in the following way. The problem of a control of the time-delay systems is described in Section 1.

The general principle of the Smith Predictor is described in Section 2. The discretization of analogue version and principle of digital Smith Predictor is introduced in Section 3. Two modifications of digital controllers that are used for self-tuning versions SPs are proposed in Section 4. Section 5. contains brief description of the recursive identification procedure. Simulation configuration is presented in Section 6. Results of simulation experiments are summed in Section 7.

2. PRINCIPLE OF SMITH PREDICTOR

The principle of the SP is shown in Fig. 1. It can be divided into two parts – the primary $G_c(s)$ controller and predictor part. This algorithm was primarily designed for continuous time PID controller. The predictor is composed of a model of the process without time delay and $G_m(s)$ (so called as the fast model) and a model of the time delay $e^{-T_d s}$. Then the complete process model is

$$G_p(s) = G_m(s) e^{-T_d s} \tag{2}$$

The fast model $G_m(s)$ is used to compute an open-loop prediction. The difference between the output of the process $y(t)$ and the model including time delay $\hat{y}(t)$ is the predicted error $\hat{e}_p(t)$ as shown is in Fig. 2, where $u(t), w(t)$, $e(t)$ and $e_s(t)$ are the control signal, reference signal, the error and the noise. If there are no modeling errors or disturbances, the error between the current process output and the model output will be null and the predictor output signal $\hat{y}_p(t)$ will be the time-delay-free output of the process. Under these conditions, the controller $G_c(s)$ can be tuned, at least in the nominal case, if the process had no time delay.

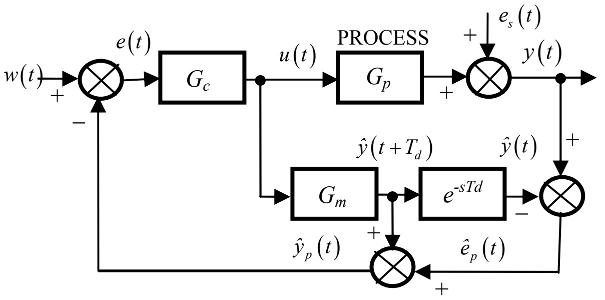


Fig. 1 Block diagram of an analogue Smith Predictor

The Smith Predictor structure for the nominal case (without modelling errors) has three fundamental properties: time-delay compensation, prediction and dynamic compensation.

3. DIGITAL SMITH PREDICTORS

Although time-delay compensators appeared in the mid 1950s, their implementation with analogue technique was very difficult and these were not used in industry. Since 1980s digital time-delay compensators can be implemented. In spite of the fact that all these algorithms are implemented on digital platforms, most works analyze only the continuous case. The digital time-delay compensators are presented e.g. in (Vogel and Edgar 1980, Palmor and Halevi 1990, Normey-Rico and Camacho 1998).

The discrete versions of the SP and its modifications are suitable for time-delay compensation in industrial practice. Most of authors designed the digital SP using discrete PID controllers with fixed parameters. However, the SP is more sensitive to process parameter variations and therefore requires an auto-tuning or adaptive approach in many practical applications (Hang et al. 1986, Hang et al. 1993). In (Torrico and Normey-Rico 2005), the structure of the discrete disturbance observer time-delay compensator is analyzed.

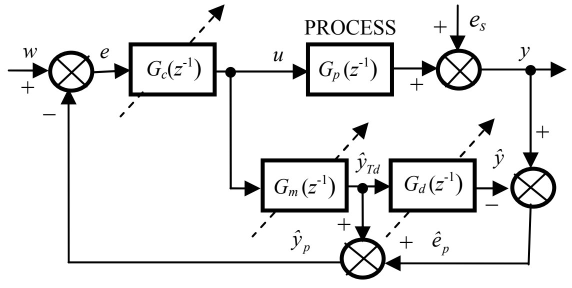


Fig. 2 Block diagram of a digital Smith Predictor with tuning

3.1 Structure of Digital SP

The block diagram of a digital SP (see Vogel and Edgar 1980, Hang et al. 1986, Hang et al. 1989, Hang et al. 1993) is shown in Fig. 2. The function of the digital version is similar to the classical analogue version. The block $G_m(z^{-1})$ represents process dynamics without the time-delay and is used to compute an open-loop prediction. The difference between the output of the process y and the model including time delay $\hat{y}$ is the predicted error $\hat{e}_p$ as shown is in Fig. 2, where u, w, e and e_s are the control signal, the reference signal, the error and the noise. If there are no modelling errors or disturbances, the error between the current process output and the model output will be null and the predictor output signal $\hat{y}_p$ will be the time-delay-free output of the process. Under these conditions, the controller $G_c(s)$ can be tuned, at least in the nominal case, as if the process had no time-delay. The primary (main) controller $G_c(z^{-1})$ can be designed by the different approaches (for example digital PID control or methods based on algebraic approach). The outward feedback-loop through the block $G_d(z^{-1})$ in Fig. 2 is used to compensate for load disturbances and modelling errors. The dash arrows indicate the self-tuned parts of the Smith Predictor.

Most industrial processes can be approximated by a reduced order model with some pure time-delay. Consider the following second order linear model with a time-delay

$$G\left(z^{-1}\right)=\frac{B\left(z^{-1}\right)}{A\left(z^{-1}\right)}z^{-d}=\frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}}z^{-d} \tag{3}$$

for demonstration of some approaches to the design of the adaptive Smith Predictor. The term z^{-d} represents the pure discrete time-delay. The time-delay is equal to dT_0 where T_0 is the sampling period.

If the time-delay is not an exact multiple of the sampling period T_0, then dT_0 represents the largest integer multiple of the sampling period with remaining fractional deal absorbed into $B\left(z^{-1}\right)$ using the modified z-transformation (Åström and Wittenmark 1984).

3.2 Identification of Time-delay

In this paper, the time-delay is assumed approximately known or possible to be obtained separately from an off-line identification using the least squares method (Ljung 1987).

$$\hat{\boldsymbol{\Theta}}=\left(\boldsymbol{F}^T\boldsymbol{F}\right)^{-1}\boldsymbol{F}^T\boldsymbol{y} \tag{4}$$

where the matrix $\boldsymbol{F}$ has dimension (N-n-d, $2n$), the vector $\boldsymbol{y}$ (N-n-d) and the vector of parameter model estimates $\hat{\boldsymbol{\Theta}}$ ($2n$). N is the number of samples of measured input and output data, n is the model order.

Equation (4) serves for a one-off calculation of the vector of parameter estimates $\hat{\boldsymbol{\Theta}}$ using N samples of measured data. The individual vectors and matrix in equation (4) have the form

$$\boldsymbol{y}^T=\left[y(n+d+1)\quad y(n+d+2)\quad \cdots \quad y(N)\right] \tag{5}$$

$$\hat{\boldsymbol{\Theta}}^T=\left[\hat{a}_1\quad \hat{a}_2\quad \cdots\quad \hat{a}_n\quad \hat{b}_1\quad \hat{b}_2\quad \cdots\quad \hat{b}_n\right] \tag{6}$$

$$\boldsymbol{F}=\left[\begin{matrix} -y(n+d) & -y(n+d-1) & \cdots & -y(d+1) \\ -y(n+d+1) & -y(n+d) & \cdots & -y(d+2) \\ \vdots & \vdots & \cdots & \vdots \\ -y(N-1) & -y(N-2) & \cdots & -y(N-n) \end{matrix}\right.$$

$$\left.\begin{matrix} u(n) & u(n-1) & \cdots & u(1) \\ u(n+1) & u(n) & \cdots & u(2) \\ \vdots & \vdots & \cdots & \vdots \\ u(N-d-1) & u(N-d-2) & \cdots & u(N-d-n) \end{matrix}\right] \tag{7}$$

For the second order model (3) are these expressions e.g. for $N=10$

$$\boldsymbol{y}^T=\left[y(5)\quad y(6)\quad \cdots\quad y(10)\right]$$

$$\hat{\boldsymbol{\Theta}}^T=\left[\hat{a}_1\quad \hat{a}_2\quad \hat{b}_1\quad \hat{b}_2\right]$$

$$\boldsymbol{F}=\begin{bmatrix} -y(4) & -y(3) & u(2) & u(1) \\ -y(5) & -y(4) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ -y(9) & -y(8) & u(7) & u(6) \end{bmatrix}$$

Consider that model (3) is the deterministic part of the stochastic process described by the ARX (regression) model

$$y(k)=\frac{B\left(z^{-1}\right)z^{-d}}{A\left(z^{-1}\right)}u(k)+\frac{1}{A\left(z^{-1}\right)}e_s(k) \tag{8}$$

where $e_s(k)$ is the random nonmeasurable component.

The ARX model (8) can be expressed as a stochastic difference equation

$$\begin{aligned} y(k)=&-a_1y(k-1)-a_2y(k-2)+ \\ &+b_1u(k-d-1)+b_2u(k-d-2)+e_s(k) \end{aligned} \tag{9}$$

The vector of parameter model estimates is computed by solving equation (4)

$$\hat{\boldsymbol{\Theta}}^T(k)=\left[\hat{a}_1\quad \hat{a}_2\quad \hat{b}_1\quad \hat{b}_2\right] \tag{10}$$

and is used for computation of the prediction output

$$\begin{aligned} \hat{y}(k)=&-\hat{a}_1y(k-1)-\hat{a}_2y(k-2)+ \\ &\hat{b}_1u(k-d-1)+\hat{b}_2u(k-d-2) \end{aligned} \tag{11}$$

The quality of ARX model can be judged by the prediction error, i.e. the deviation

$$\hat{e}(k)=y(k)-\hat{y}(k) \tag{12}$$

The prediction error plays a key role in identification of regressions model parameters derived from measured data. It is important for selecting the structure (order) of the regression model and a suitable sampling period. The quality of the model is also judged by the purpose for which it is used. In this paper, the prediction error was used for suitable choice of the time-delay dT_0. The LSM algorithm (4) – (7) is computed for several time-delays dT_0 and the suitable time-delay is chosen on the based of quality identification by using prediction error (12).

4. ALGORITHMS OF DIGITAL SMITH PREDICTORS

4.1 DigitalPID Smith Predictor (PIDSP)

Hang et al. 1989, 1993 used to design of the main controller $G_c\left(z^{-1}\right)$ the Dahlin PID algorithm (Dahlin 1968). This algorithm is based on the desired close-loop transfer function in the form

$$G_e\left(z^{-1}\right)=\frac{1-e^{-\alpha}}{1-z^{-1}} \tag{13}$$

where $\alpha={T_0}/{T_m}$ and T_m is desired time constant of the first order closed-loop response. It is not practical to set T_m to be small since it will demand a large control signal $u(k)$ which may easily exceed the saturation limit of the actuator. Then the individual parts of the controller are described by the transfer functions

$$G_c\left(z^{-1}\right)=\frac{\left(1-e^{-\alpha}\right)}{\left(1-z^{-1}\right)}\frac{\hat{A}\left(z^{-1}\right)}{\hat{B}(1)};\quad G_m\left(z^{-1}\right)=\frac{z^{-1}\hat{B}(1)}{\hat{A}\left(z^{-1}\right)}$$

$$G_d\left(z^{-1}\right)=\frac{z^{-d}\hat{B}\left(z^{-1}\right)}{z^{-1}\hat{B}(1)} \qquad (14)$$

where $\hat{B}(1)=\hat{B}\left(z^{-1}\right)\Big|_{z=1}=\hat{b}_1+\hat{b}_2$.

Since $G_m\left(z^{-1}\right)$ is the second order transfer function, the main controller $G_c\left(z^{-1}\right)$ becomes a digital PID controller having the following form:

$$G_c\left(z^{-1}\right)=\frac{U(z)}{E(z)}=\frac{q_0+q_1z^{-1}+q_2z^{-2}}{1-z^{-1}} \qquad (15)$$

where $q_0=\gamma, q_1=\hat{a}_1\gamma, q_2=\hat{a}_2\gamma$ using by the substitution $\gamma=\left(1-e^{-\alpha}\right)/\hat{B}(1)$. The PID controller output is given by

$$u(k)=q_0e(k)+q_1e(k-1)+q_2e(k-2)+u(k-1) \qquad (16)$$

4.2 Digital Pole Assignment Smith Predictor (PASP)

The second controller applied in this paper was designed using a polynomial approach. Polynomial control theory is based on the apparatus and methods of a linear algebra (see e.g. Kučera 1991, Vidyasagar 1985). The polynomials are the basic tool for a description of the transfer functions. They are expressed as the finite sequence of figures – the coefficients of a polynomial. Thus, the signals are expressed as infinite sequence of figures. The controller synthesis consists in the solving of linear polynomial (Diophantine) equations (Kučera 1993).

The design of the controller algorithm is based on the general block scheme of a closed-loop with two degrees of freedom (2DOF) according to Fig. 3.

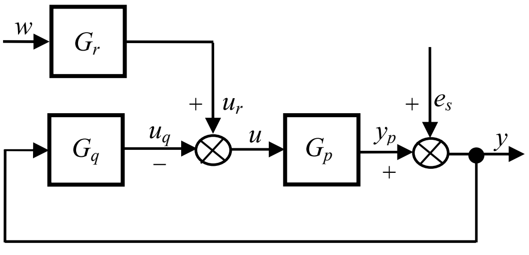


Fig. 3 Block diagram of a closed loop 2DOF control system

The controlled process is given by the transfer function in the form

$$G_p(z^{-1})=\frac{Y_p(z)}{U(z)}=\frac{B(z^{-1})}{A(z^{-1})} \qquad (17)$$

where A and B are the second order polynomials. The controller contains the feedback part G_q and the feedforward part G_r. Then the digital controllers can be expressed in the form of a discrete transfer functions

$$G_r\left(z^{-1}\right)=\frac{R\left(z^{-1}\right)}{P\left(z^{-1}\right)}=\frac{r_0}{1+p_1z^{-1}} \qquad (18)$$

$$G_q\left(z^{-1}\right)=\frac{Q\left(z^{-1}\right)}{P\left(z^{-1}\right)}=\frac{q_0+q_1z^{-1}+q_2z^{-2}}{\left(1+p_1z^{-1}\right)\left(1-z^{-1}\right)} \qquad (19)$$

According to the scheme presented in Fig. 3 (for e_s = 0), the output y can be expressed as

$$Y\left(z^{-1}\right)=\frac{G_p(z)G_r(z)}{1+G_p(z)G_q(z)}W\left(z^{-1}\right) \qquad (20)$$

Upon substituting from Equation (17) - (19) into Equation (20) it yields

$$Y(z^{-1})=\frac{B(z^{-1})R(z^{-1})}{A(z^{-1})P(z^{-1})+B(z^{-1})Q(z^{-1})}W(z^{-1}) \qquad (21)$$

where

$$A(z^{-1})P(z^{-1})+B(z^{-1})Q(z^{-1})=D(z^{-1}) \qquad (22)$$

is the characteristic polynomial.

The procedure leading to determination of polynomials Q, R and P in (18) and (19) can be briefly described as follows (see Bobál et al. 2005). A feedback part of the controller is given by a solution of the polynomial Diophantine equation (22). An asymptotic tracking is provided by a feedforward part of the controller given by a solution of the polynomial Diophantine equation

$$S(z^{-1})D_w\left(z^{-1}\right)+B(z^{-1})R(z^{-1})=D(z^{-1}) \qquad (23)$$

For a step-changing reference signal value, polynomial $D_w\left(z^{-1}\right)=1-z^{-1}$ and S is an auxiliary polynomial which does not enter into controller design.

A feedback controller to control a second-order system without time-delay will be derived from Equation (22). A system of linear equations can be obtained using the uncertain coefficients method

$$\begin{bmatrix}\hat{b}_1 & 0 & 0 & 1\\ \hat{b}_2 & \hat{b}_1 & 0 & \hat{a}_1-1\\ 0 & \hat{b}_2 & \hat{b}_1 & \hat{a}_2-\hat{a}_1\\ 0 & 0 & b_2 & -\hat{a}_2\end{bmatrix}\begin{bmatrix}q_0\\ q_1\\ q_2\\ p_1\end{bmatrix}=\begin{bmatrix}d_1+1-\hat{a}_1\\ d_2+\hat{a}_1-\hat{a}_2\\ d_3+\hat{a}_2\\ d_4\end{bmatrix} \qquad (24)$$

where the characteristic polynomial is chosen as

$$D\left(z^{-1}\right)=1+d_1z^{-1}+d_2z^{-2}+d_3z^{-3}+d_4z^{-4} \qquad (25)$$

For a step-changing reference signal value it is possible to solve Equation (23) by substituting z = 1

$$R=r_0=\frac{D(1)}{B(1)}=\frac{1+d_1+d_2+d_3+d_4}{b_1+b_2} \qquad (26)$$

The 2DOF controller output is given by

$$u(k)=r_0w(k)-q_0y(k)-q_1y(k-1)-q_2y(k-2)+ \\ +(1+p_1)u(k-1)+p_1u(k-2) \tag{27}$$

5. RECURSIVE IDENTIFICATION PROCEDURE

The regression (ARX) model of the following form

$$y(k)=\boldsymbol{\Theta}^T(k)\boldsymbol{\Phi}(k)+e_s(k) \tag{28}$$

is used in the identification part of the designed controller algorithms, where

$$\boldsymbol{\Theta}^T(k)=\begin{bmatrix}a_1 & a_2 & b_1 & b_2\end{bmatrix} \tag{29}$$

is the vector of model parameters and

$$\boldsymbol{\Phi}^T(k-1)=\begin{bmatrix}-y(k-1) & -y(k-2) & u(k-d-1) & u(k-d-2)\end{bmatrix} \tag{30}$$

is the regression vector. The non-measurable random component $e_s(k)$ is assumed to have zero mean value $E[e_s(k)] = 0$ and constant covariance (dispersion) $R = E[e_s^2(k)]$.

All digital adaptive SP controllers use the algorithm of identification based on the Recursive Least Squares Method (RLSM) extended to include the technique of directional (adaptive) forgetting. Numerical stability is improved by means of the LD decomposition (Kulhavý 1987, Bobál et al. 2005). This method is based on the idea of changing the influence of input-output data pairs to the current estimates. The weights are assigned according to amount of information carried by the data.

6. SIMULATION VERIFICATION ADAPTIVE DIGITAL SP CONTROLLER ALGORITHMS.

Simulation is a useful tool for the synthesis of control systems, allowing one not only to create mathematical models of a process but also to design virtual controllers in a computer. The mathematical models provided are sufficiently close to a real object that simulation can be used to verify the dynamic characteristics of control loops when the structure or parameters of the controller change. The models of the processes may also be excited by various random noise generators which can simulate the stochastic characteristics of the processes noise signals with similar properties to disturbance signals measured in the machinery.

The above mentioned SP controllers are not suitable for the control of unstable processes. Therefore, three types of processes were chosen for simulation verification of digital adaptive SP controller algorithms.

Consider the following continuous-time transfer functions:

1) Stable non-oscillatory $G_1(s)=\dfrac{2}{(s+1)(4s+1)}e^{-4s}$

2) Stable oscillatory $G_2(s)=\dfrac{2}{4s^2+2s+1}e^{-4s}$

3) With non-minimum phase $G_3(s)=\dfrac{-5s+1}{(s+1)(4s+1)}e^{-4s}$

Let us now discretize them a sampling period $T_0 = 2\,\mathrm{s}$. The discrete forms of these transfer functions are (see Equation (3))

$$G_1(z^{-1})=\frac{0.4728z^{-1}+0.2076z^{-2}}{1-0.7419z^{-1}+0.0821z^{-2}}z^{-2}$$

$$G_2(z^{-1})=\frac{0.6806z^{-1}+0.4834z^{-2}}{1-0.7859z^{-1}+0.3679z^{-2}}z^{-2}$$

$$G_3(z^{-1})=\frac{-0.5489z^{-1}+0.8897z^{-2}}{1-0.7419z^{-1}+0.0821z^{-2}}z^{-2}$$

A simulation verification of proposed design was performed in MATLAB/SIMULINK environment. A typical control scheme used is depicted in Fig. 4.

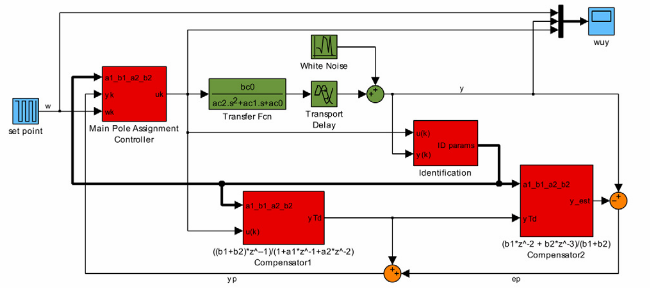


Fig. 4 Simulink control scheme

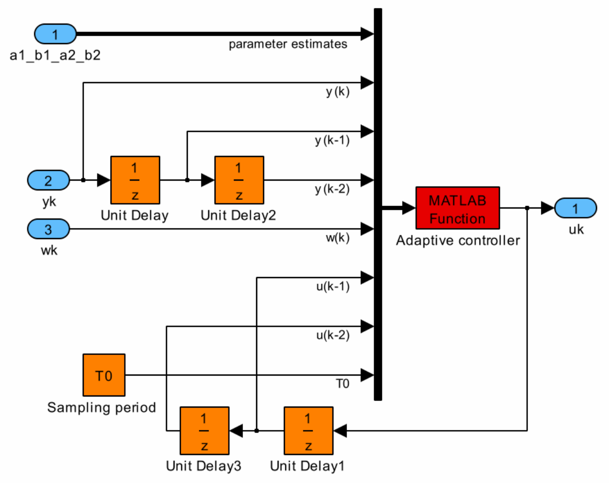
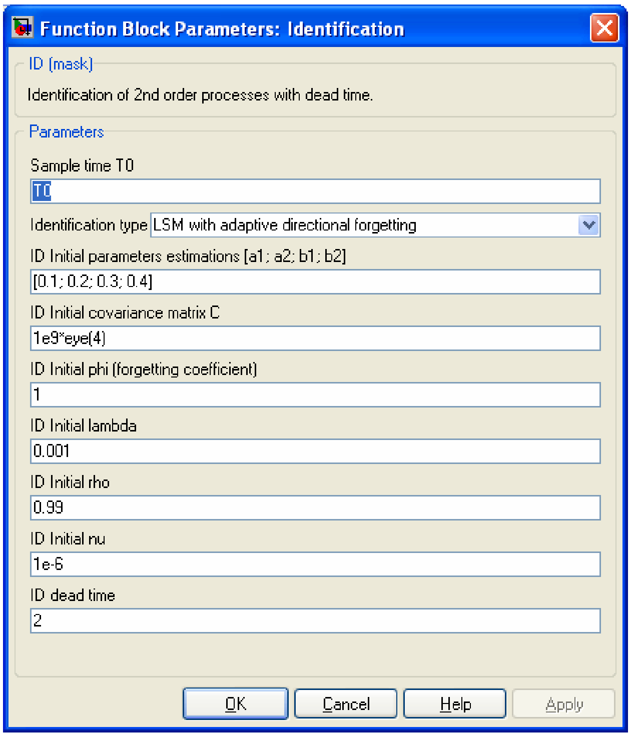


Fig. 5 Internal structure of the controller

This scheme is used for systems with time-delay of two sample steps. Individual blocks of the Simulink scheme correspond to blocks of the general control scheme presented in Fig. 2. Blocks Compensator 1 and Compensator 2 are parts of the Smith Predictor and they correspond to $G_m(z^{-1})$ and $G_d(z^{-1})$ blocks of Fig. 2 respectively. The control algorithm is encapsulated in Main Pole Assignment Controller which corresponds to $G_c(z^{-1})$ Fig. 2 block. The Identification block performs the on-line identification of controlled system and outputs the estimates of 2nd order ARX model (a1, b1, a2, b2) parameter.

The internal structure of the Main Pole Assignment Controller block is shown in Fig. 5. Block MATLAB Fcn is the heart of the controller. The inputs to this function are current ARX estimates, current and previous values of process without time-delay, reference signal as well as previous control values and sample time. The MATLAB Fcn is a standard m-function which carries out desired control algorithm as described in Section 4.

The on-line identification part of the scheme, which is represented by block Identification block in Fig. 3, uses several parameters that are entered via standard Simulink dialog. This dialog is presented in Fig. 6.



Function Block Parameters: Identification

ID (mask)

Identification of 2nd order processes with dead time.

Parameters

Sample time T0: T0

Identification type: LSM with adaptive directional forgetting

ID Initial parameters estimations [a1; a2; b1; b2]: [0.1; 0.2; 0.3; 0.4]

ID Initial covariance matrix C: 1e9*eye(4)

ID Initial phi (forgetting coefficient): 1

ID Initial lambda: 0.001

ID Initial rho: 0.99

ID Initial nu: 1e-6

ID dead time: 2

Fig. 6. Dialog for setting identification parameters

The most important parameters form the point of view of the problem this papers is coping with are sample time, initial parameters estimations and dead time. The dead time is not entered in time units but in sample times. The other parameters affect the method used to compute ARX model and their detailed description can be found in (Kulhavý 1987, Bobál et al. 2005).

7. SIMULATION RESULTS

The configuration for simulation verification of the designed algorithms was chosen as follows:

- All three control loops were verified in the non-adaptive versions without a random noise.
- A suitable time constant T_m was chosen for the control using the PIDSP controller and the pole assignment of the closed-loop was calculated. These poles were used for the design of the PASP controller.
- Both control loops were verified in the adaptive versions with a random noise. Firstly, without a priori information (the initial values of the model parameter estimates were chosen randomly). Secondly, using a priori information (the initial estimates were chosen on the basis of the previous experiments).
- The outputs of the process models were influenced by White Noise Generator with mean value $E = 0$ and covariance $R = 10^{-4}$.

7.1 Simulation verification of Adaptive Digital PIDSP

Figs. 7 - 10 illustrate the simulation control performance using PIDSP controller (14) - (16). From Figs. 7 and 8 (the control of the stable model $G_1\left(z^{-1}\right)$) is obvious that the control process is dependent on knowledge of a priori information. The process output y has a large overshoot, when the initial model parameter estimates are chosen randomly.

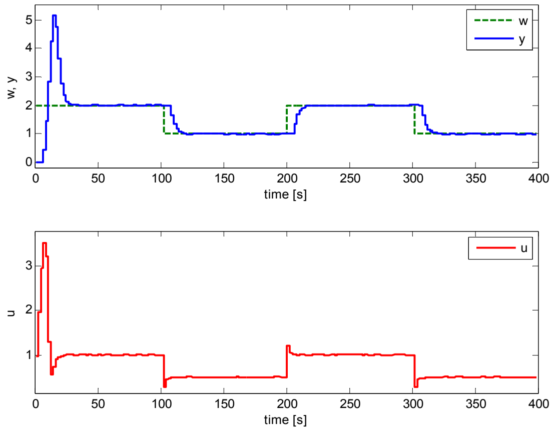


Fig. 7 Simulation results: control of the model $G_1\left(z^{-1}\right)$, controller PIDSP (without a priori information)

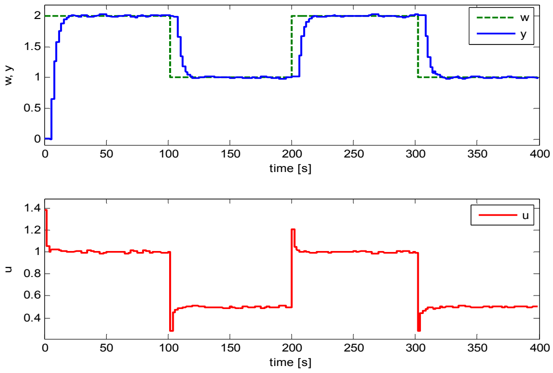


Fig. 8 Simulation results: control of the model $G_1\left(z^{-1}\right)$, controller PIDSP (with a priori information)

Using a priori information (the initial estimates were chosen on the basis of the previous experiments) leads to very good control quality (without overshoot of y and with short settling time).

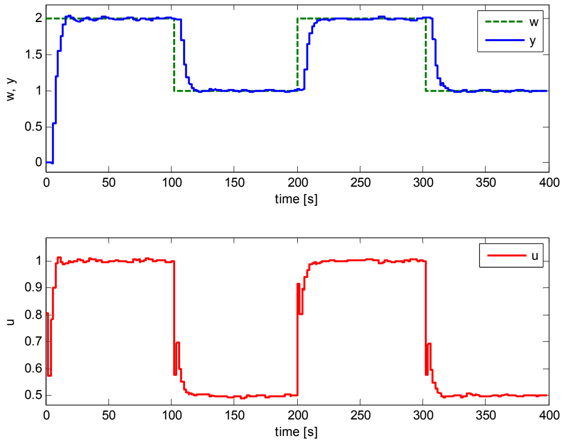


Fig. 9 Simulation results: control of the model $G_2\left(z^{-1}\right)$, controller PIDSP (with a priori information)

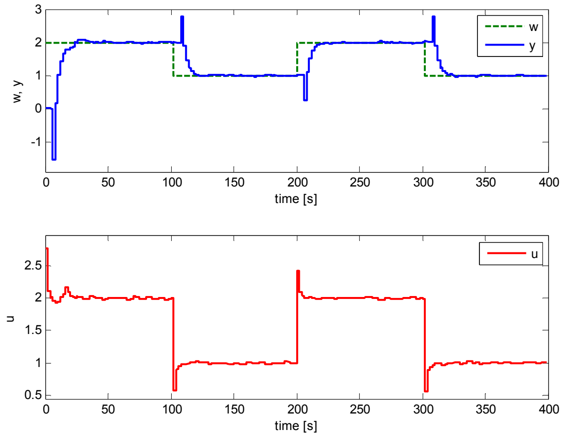


Fig. 10 Simulation results: control of the model $G_3\left(z^{-1}\right)$, controller PIDSP (with a priori information)

Simulation results for the models $G_2\left(z^{-1}\right)$ (the stable oscillatory model) and $G_3\left(z^{-1}\right)$ (the non-minimum phase model) are shown in Figs. 9 and 10. The control quality (with a priori information) is very good.

7.2 Simulation verification of Adaptive Digital PASP

Figs. 11 - 13 illustrate the simulation control performance using PASP controller (14), (27). From Figs. 11 and 12 (the control of the stable model $G_1\left(z^{-1}\right)$), it is obvious that the control process is not dependent on knowledge of a priori information (the control courses in both cases are practically identical). In the case of choosing of the suitable closed-loop poles, the adaptive PASP controller is more robust than the adaptive PIDSP controller.

Fig. 13 illustrates the simulation control performance of the stable oscillatory model $G_2\left(z^{-1}\right)$. The control process is

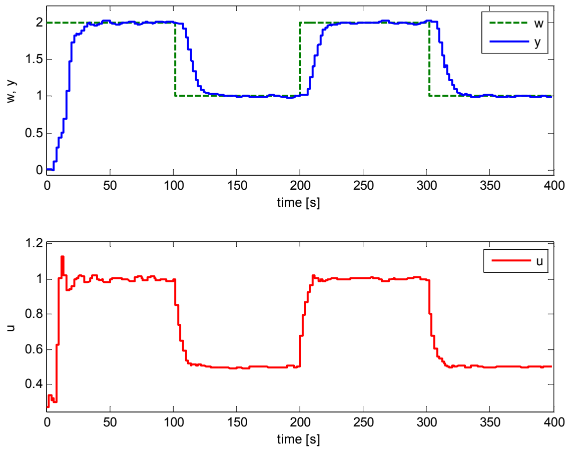


Fig. 11 Simulation results: control of model $G_1\left(z^{-1}\right)$, controller PASP (without a priori information)

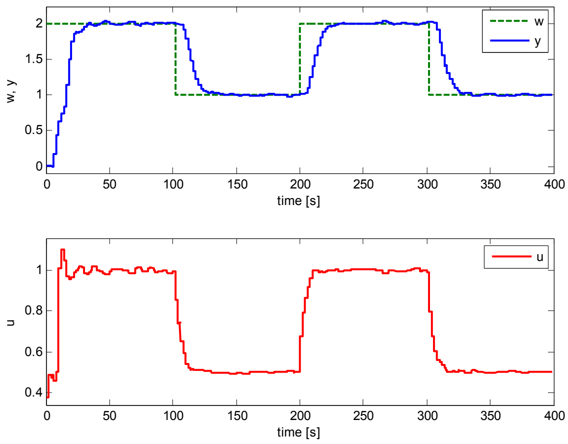


Fig. 12 Simulation results: control of the model $G_1\left(z^{-1}\right)$, controller PASP (with a priori information)

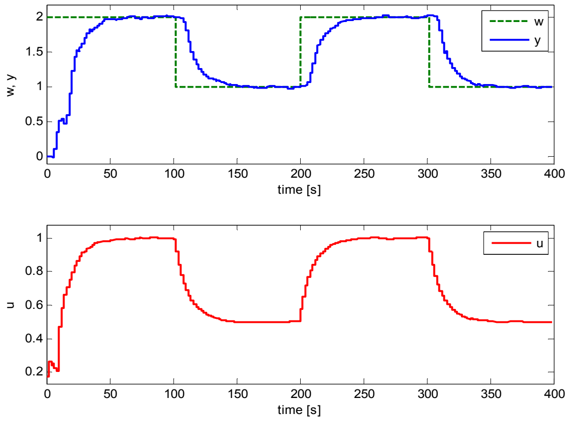


Fig. 13 Simulation results: control of the model $G_2\left(z^{-1}\right)$, controller PASP (with a priori information)

relatively slow without overshot of y and u (it is the cautious adaptive controller).

Fig. 14 illustrates the simulation control performance of the non-minimum phase model $G_3\left(z^{-1}\right)$. The control process is good after initial part.

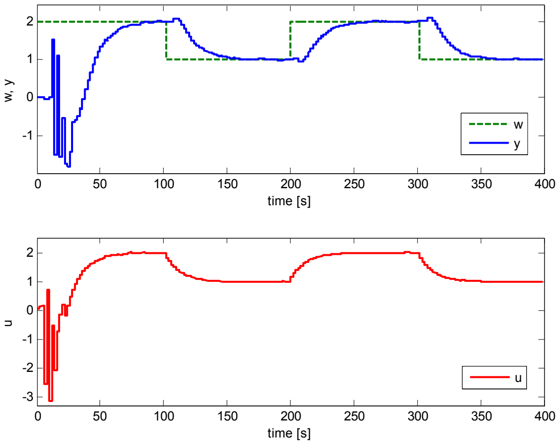


Fig. 14 Simulation results: control of the model $G_3\left(z^{-1}\right)$, controller PASP (with a priori information)

7. CONCLUSION

Adaptive Smith Predictor algorithms for control of processes with time-delay based on polynomial design (pole assignment) was proposed. The polynomial controllers were derived purposely by analytical way (without utilization of numerical methods) to obtain algorithms with easy implementability in industrial practice. The pole assignment control algorithm was compared by simulation with adaptive digital Smith PID Predictor. Three models of control processes were used for simulation verification (the stable non-oscillatory, the stable oscillatory and the non-minimum phase). Results of simulation verification demonstrated advantages and disadvantages of individual approaches (see Figs. 7 – 14) for control of above mentioned processes with time-delay. The control quality depended on suitable start-up conditions of the recursive identification – important part plays a priori information. For the pole assignment method is very important suitable choice of the closed – loop poles. The designed adaptive SP algorithms will be verified in real time conditions for a control of the laboratory heat exchanger.

ACKNOWLEDGMENTS

The authors wish to thank to the Ministry of Education of the Czech Republic (MSM7088352101) for financial support. This article was created with support of Operational Programme Research and Development for Innovations co-funded by the European Regional Development Fund (ERDF), national budget of the Czech Republic within the framework of the Centre of Polymer Systems project (reg. number: CZ.1.05/2.1.00/03.0111) and grant 1M0567.

REFERENCES

Åström, K.J. and B. Wittenmark (1984). *Computer Controlled Systems-Theory and Design.* Prentice-Hall, Englewood Cliffs, NJ.

Åström, K.J. and T. Hägglund (1995). *PID Controllers: Theory, Design and Tuning.* Instrument Society of America.

Bobál, V. Böhm, J., Fessl, J. and J. Macháček (2005). *Digital Self-tuning Controllers: Algrithms, Implementation and Applications.* Springer-Verlag, London.

Dahlin, D.B. (1968). Designing and tuning digital controllers. *Inst. Control Systems* **42**, 77-73.

Guo, S.N., Wang, W. and L.S. Shieh (2000). Discretisation of two degree-of-freedom controller and system with state, and output delays. *IEE Proceedings. Control Theory and Applications* **147**, 87-96.

Hang C.C, Lim, K.W. and T.T. Tay (1986). On practical applications of adaptive control, In: *Proceedings Adaptive and Learning Systems-Theory and Applications*, London/New York, 105-108.

Hang, C.C., Lim, K. W. and B.W. Chong (1989). A dual-rate adaptive digital Smith predictor. *Automatica* **20**, 1-16.

Hang, C. C., Tong, H.L. and K.H. Weng (1993). *Adaptive Control.* Instrument Society of America.

Kučera, V. (1991). *Analysis and Design of Discrete Linear Control Systems.* Prentice-Hall, Englewood Cliffs, NJ.

Kučera, V. (1993). Diophantine equations in control – a survey. *Automatica* **29**, 1361-1375.

Kulhavý, R. (1987). Restricted exponential forgetting in real time identification. *Automatica* **23**, 586-600.

Ljung, L. (1987). *System Identification: Theory for the User.* Prentice-Hall, Englewood Cliffs, NJ.

Normey-Rico, J.E. and E.F. Camacho (1998). Dead-time compensators: A unified approach. In: *Proceedings of IFAC Workshop on Linear Time Delay Systems (LDTS'98)*, Grenoble, France, 141-146.

Normey-Rico, J. E. and E. F. Camacho (2007). *Control of Dead-time Processes.* Springer-Verlag, London.

Palmor, Z.J. and Y. Halevi (1990). Robustness properties of sampled-data systems with dead time compensators. *Automatica* **26**, 637-640.

Smith, O.J. (1957). Closed control of loops, *Chem. Eng. Progress* **53**, 217-219.

Torrico, B.C. and J.E. Normey-Rico (2005). 2DOF discrete dead-time compensators for stable and integrative processes with dead-time. *Journal of Process Control* **15**, 341-352.

Vidyasagar, M. (1985). *Control System Synthesis: A Factorization Approach.* MIT Press, Cambridge M.A.

Vogel, E.F. and T.F. Edgar (1980). A new dead time compensator for digital control. In: *Proceedings ISA Annual Conference*, Houston, USA, 29-46.

Ziegler, J.G. and N.B. Nichols (1942). Optimum settings for automatic controllers. *Trans. ASME* **64**, 1942, 759-768.

Robust PSD controller design

V. Veselý, D. Rosinová

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics
Bratislava, Slovakia
(e-mail:danica.rosinova@stuba.sk, vojtech.vesely@stuba.sk)

Abstract: A state space approach to a design of PSD robust controllers is studied for linear uncertain system with affine (polytopic) uncertainty. The discrete time PSD controller design is based on stability condition derived using parameter dependent Lyapunov-Krasovskii function in the form for time-delay system. The resulting design employs solution of BMI, the results are illustrated on the example.

1. INTRODUCTION

Robustness belongs to important issues in control design for real plants. In practice, uncertainties are always present in modelling and control of real systems (modelling errors due to linearization and approximation, disturbances etc.), which must be taken into consideration. The appropriate control has to cope with uncertainties and guarantee closed loop stability and required performance qualities overall the uncertainty domain. Various approaches have been developed in robust control both in time and frequency domains. A frequently used paradigm developed in past decades formulates the problem of robust stability and robust control as an optimization problem. Efficient computational techniques have been developed recently for solving Linear Matrix Inequality (LMI), which enables to solve a large set of convex problems in polynomial time (e.g. Boyd et al., 1994). Significant effort has been made in this field to formulate control problems within algebraic framework (Skelton et al., 1998), and transform them into LMI. Though many control problems for uncertain linear systems can be formulated as convex one, there are still many important control problems even for linear systems, that have been proven as NP hard (Blondel and Tsitsiklis, 1997). Robust static output feedback (SOF) control belongs also to this class, generally it can be formulated as bilinear matrix inequality (BMI). In this case either solution through BMI solver (as PENBMI) can be used, or, convex approximation or linearization can be applied (deOliveira et al., 2000; Han and Skelton, 2003, Veselý, 2003; Rosinová and Veselý, 2003). Characterization of basic LMI and BMI features in control problems can be found in (Van Antwerp and Braatz 2000).

Proportional-integral-derivative (PID) controllers belong to the most popular and frequent ones in the industrial applications. For a discrete time case often PSD abbreviation is used instead of PID, where “S” stands for a summation term (instead of integration). Results on LMI approach to design PID controller can be found e.g. in (Ge et al. 2002, Zheng Feng et al. 2000). Robust PSD controller design can be treated as dynamic controller, which further can be formulated as SOF problem for augmented system including controller dynamics, (Rosinová and Veselý, 2007). As indicated above, SOF problem, which is generally non-convex, can be either solved by solving BMI or by linearization (convex approximation) and then through LMI.

In this paper, another approach is proposed for robust PSD controller design, where the difference term is considered in the framework of time-delay systems and the respective Lyapunov-Krasovskii function comprising two parts. In Section 2, robust PSD controller problem is formulated for linear uncertain polytopic system with quadratic performance index. Section 3 presents the main result- robust stability condition with guaranteed cost formulated for robust PSD controller. This condition is developed using parameter dependent Lyapunov function in the form for time- delay systems. The proposed approach is illustrated on the example in Section 4.

2. PRELIMINARIES AND PROBLEM FORMULATION

Consider the class of linear uncertain discrete-time systems described as:

$$\begin{aligned} x(t+1) &= A(\alpha)x(t) + B(\alpha)u(t) \\ y(t) &= Cx(t) \end{aligned} \qquad t = 0,1,... \qquad (1)$$

where $x(t) \in R^n, u(t) \in R^m, y(t) \in R^l$ are state, control and output vectors respectively; uncertain model matrices $A(\alpha), B(\alpha)$ are from convex polytopic uncertainty domain given by polytope vertices $A_j \in R^{nxn}, B_j \in R^{nxm}$, $j = 1,...,N$:

$$(A(\alpha), B(\alpha)) \in S,$$

$$S = \left\{ (A(\alpha), B(\alpha)) \colon A(\alpha) = \sum_{j=1}^{N} \alpha_j A_j, B(\alpha) = \sum_{j=1}^{N} \alpha_j B_j, \sum_{j=1}^{N} \alpha_j = 1, \alpha_j \geq 0 \right\} \qquad (2)$$

2.1 PSD controller

Assuming that the input reference variable $w(t)$ is constant (its changes are relatively slow in comparison with system dynamics), PSD control algorithm for uncertain system (1), (2) can be considered as

$$u(t) = k_p y(t) + k_i \sum_{i=0}^{t} y_i(k) + k_d [y_d(t) - y_d(t-1)] \qquad (3)$$

where $y_i(t) = C_i x(t), y_d(t) = C_d x(t)$ denote the respective outputs for summation (discrete approximation of integration) and difference term of control algorithm respectively, which in general can differ from output $y(t)$; k_p, k_i, k_d are constant matrices of corresponding dimensions.

In parallel with standard approach for continuous-time case, to include summation term we introduce augmented state vector defined by

$$z(t) = \begin{bmatrix} x(t) \\ r(t) \end{bmatrix}$$

where $r(t+1) = r(t) + y_i(t)$, $t = 0,1,...$ (4)

Using (4), PSD control algorithm can be rewritten as

$$\begin{aligned} u(t) &= k_p Cx(t) + k_i r(t) + k_d C_d x(t) - k_d C_d x(t-1) = \\ &= (k_p C + k_d C_d) x(t) + k_i r(t) - k_d C_d x(t-1) \qquad (5) \\ &= \begin{bmatrix} k_p C + k_d C_d & k_i \end{bmatrix} z(t) - k_d C_d x(t-1) \end{aligned}$$

From (1), (4) and (5) we obtain the uncertain closed-loop polytopic system described in a compact form as

$$\begin{aligned} z(t+1) &= A_C(\alpha) z(t) + B_C(\alpha) K_C C_C z(t) - B_C(\alpha) k_d C_{d1} z(t-1) = \\ &= (A_C(\alpha) + B_C(\alpha) K_C C_C) z(t) - B_C(\alpha) k_d C_{d1} z(t-1) \end{aligned} \qquad (6)$$

where

$$A_C(\alpha) = \begin{bmatrix} A(\alpha) & 0 \\ C_i & I \end{bmatrix}, \; B_C(\alpha) = \begin{bmatrix} B(\alpha) \\ 0 \end{bmatrix}, \; K_C = \begin{bmatrix} k_p & k_d & k_i \end{bmatrix}$$

$$C_C = \begin{bmatrix} C & 0 \\ C_d & 0 \\ 0 & I \end{bmatrix}, \; C_{d1} = [C_d \;\; 0] \qquad (7)$$

2.2 Performance index

A performance for closed loop system (6) is assessed using quadratic cost function

$$J = \sum_{t=0}^{\infty} J(t)$$

$$J(t) = z(t)^T Q z(t) + u(t)^T R u(t) + z(t-1)^T S\, z(t-1) \qquad (8)$$

where $Q, S \in R^{nxn}, R \in R^{m \times m}$ are symmetric positive definite matrices.

Definition 1

Control law (5) is called guaranteed cost control when there exist PID controller parameter matrices k_p, k_i, k_d and a constant J_0 such that

$$J \le J_0$$

holds for closed loop system (6); J_0 is the guaranteed cost.

2.3 Robust stability with guaranteed cost

Let $V(t)$ is Lyapunov function for uncertain closed-loop system (6). From LQ theory, see e.g. (Rosinova et al. 2003), the following lemma for robust stability of system (6) with guaranteed cost holds.

Lemma 1

Control algorithm (5) is the guaranteed cost control law for the closed loop system (6) if and only if there exist $V(t) > 0$ and constant matrices k_p, k_i, k_d such that the following inequality holds for $t = 0,1,...$

$$B(t) = \Delta V(t) + J(t) < 0 \qquad (9)$$

Moreover, summarizing (9) from initial time t_0 to $t \to \infty$, the following inequality is obtained

$$-V(t_0) + J < 0 \qquad (10)$$

Definition 1 with inequality (10) provides guaranteed cost

$$J_0 = V(t_0)$$

for closed loop system (6) with control law (5).

3. MAIN RESULT: ROBUST PSD CONTROLLER DESIGN

In this section a robust stability condition including guaranteed cost is developed based on Lyapunov-Krasovskii function. Due to the presence of $z(t-1)$ in control algorithm, we consider parameter dependent Lyapunov-Krasovskii function consisting of the respective two parts for $z(t)$ and $z(t-1)$

$$V(t) = V_1(t) + V_2(t) \qquad (11)$$

where $V_1(t) = z(t)^T P_1(\alpha) z(t)$ (11a)

$V_2(t) = z(t-1)^T P_2(\alpha) z(t-1)$ (11b)

and $P_1(\alpha) \in R^{n \times n}$, $P_2(\alpha) \in R^{n \times n}$ are parameter dependent symmetric positive definite matrices of the corresponding dimensions.

In the following developments we employ the backward difference given by

$$\Delta z(t) = z(t) - z(t-1) \qquad (12)$$

"to interconnect" actual and past values of *z*, which can be interpreted as discrete counterpart to Leibnitz-Newton formula used for continuous time-delay systems.

In the development of a robust stability condition for uncertain closed-loop system we use augmented state vector

$$v(t) = \begin{bmatrix} z(t+1) \\ z(t) \\ \Delta z(t) \end{bmatrix}. \tag{13}$$

Firstly, we express particular components which will be used later in terms of denotation (7), (12) and (13).

Control law (5) can be rewritten as

$$\begin{aligned} u(t) &= K_C C_C\, z(t) - k_d C_d x(t-1) \\ &= K_C C_C\, z(t) - k_d C_{d1}(z(t) - \Delta z(t)) \\ &= \begin{bmatrix} 0 & K_C C_C - k_d C_{d1} & k_d C_{d1} \end{bmatrix} v(t) \end{aligned} \tag{14}$$

Closed loop system (6) can be analogically rewritten as

$$z(t+1) = \widetilde{A}_C(\alpha) z(t) + B_C(\alpha) k_d C_{d1} \Delta z(t) \tag{15}$$

with

$$\widetilde{A}_C(\alpha) = A_C(\alpha) + B_C(\alpha) K_C C_C - B_C(\alpha) k_d C_{d1}$$

from which we have

$$\begin{bmatrix} I & -\widetilde{A}_C(\alpha) & -B_C(\alpha) k_d C_{d1} \end{bmatrix} \begin{bmatrix} z(t+1) \\ z(t) \\ \Delta z(t) \end{bmatrix} = 0 \tag{16}$$

The first difference of Lyapunov-Krasovskii function (11) is

$$\Delta V(t) = \Delta V_1(t) + \Delta V_2(t) \tag{17a}$$

$$\Delta V_1(t) = z(t+1)^T P_1(\alpha) z(t+1) - z(t)^T P_1(\alpha) z(t) \tag{17b}$$

$$\begin{aligned} &\Delta V_2(t) = z(t)^T P_2(\alpha) z(t) - z(t-1)^T P_2(\alpha) z(t-1) = \\ &= -\Delta z(t)^T P_2(\alpha) \Delta z(t) + z(t)^T P_2(\alpha) \Delta z(t) + \Delta z(t)^T P_2(\alpha) z(t) \end{aligned} \tag{17c}$$

Substituting (17b) and (17c) into (17a) and rearranging, we obtain $\Delta V(t)$ in a compact form as

$$\Delta V(t) = v(t)^T \begin{bmatrix} P_1(\alpha) & 0 & 0 \\ 0 & -P_1(\alpha) & P_2(\alpha) \\ 0 & P_2(\alpha) & -P_2(\alpha) \end{bmatrix} v(t) \tag{18}$$

The main result on robust stability condition is given in the next theorem.

Theorem 1

Consider the uncertain discrete-time system (1) with PID controller (3) and parameter dependent Lyapunov-Krasovskii function (11),(11a),(11b). Control algorithm (3), or, alternatively, (14) is guaranteed cost robust control law for performance index (8) if and only if there exist positive definite matrices $P_1(\alpha) \in R^{n \times n}$, $P_2(\alpha) \in R^{n \times n}$ and constant matrices N_1, N_2, N_3 of appropriate dimensions such that

$$v(t)^T W(\alpha) v(t) < 0 \tag{19}$$

where $v(t)$ is given in (13) and $W(\alpha) = \{w_{ij}(\alpha)\}$:

$$\begin{aligned} &w_{11}(\alpha) = N_1 + N_1^T + P_1(\alpha) \\ &w_{12}(\alpha) = -N_1^T \widetilde{A}_C(\alpha) + N_2 \\ &w_{13}(\alpha) = -N_1^T B_C(\alpha) k_d C_{d1} + N_3 \\ &w_{22}(\alpha) = -N_2^T \widetilde{A}_C(\alpha) - \widetilde{A}_C^T(\alpha) N_2 - P_1(\alpha) + \\ &+ \left(K_C C_C - k_d C_{d1}\right)^T R (K_C C_C - k_d C_{d1}) + S + Q \\ &w_{23}(\alpha) = -N_2^T B_C(\alpha) k_d C_{d1} - \widetilde{A}_C^T(\alpha) N_3 + P_2(\alpha) + \\ &+ \left(K_C C_C - k_d C_{d1}\right)^T R\, k_d C_{d1} - S \\ &w_{33}(\alpha) = -N_3^T B_C(\alpha) k_d C_{d1} - C_{d1}^T k_d^T B_C^T(\alpha) N_3 - P_2(\alpha) + S + \\ &+ C_{d1}^T k_d{}^T R\, k_d C_{d1} \end{aligned}$$

Proof

To derive robust stability condition we use (9) together with (16). Due to (16), the following equality holds

$$\begin{aligned} &v(t)^T \begin{bmatrix} N_1^T \\ N_2^T \\ N_3^T \end{bmatrix} \begin{bmatrix} I & -\widetilde{A}_C(\alpha) & -B_C(\alpha) k_d C_{d1} \end{bmatrix} v(t) + \\ &+ v(t)^T \begin{bmatrix} I \\ -\widetilde{A}_C^T \\ -C_{d1}^T k_d^T B_C^T(\alpha) \end{bmatrix} \begin{bmatrix} N_1 & N_2 & N_3 \end{bmatrix} v(t) = 0 \end{aligned} \tag{20}$$

By substituting from (17), (14) and (8) to (9) for $J(t)$ and $\Delta V(t)$ respectively, and adding (20) to the left hand side of (9), the resulting inequality (19) is obtained. □

Robust stability condition (19) can be advantageously applied for polytopic systems with uncertainties respective to (2). Parameter dependent Lyapunov-Krasovskii function is considered in the form

$$\begin{aligned} P_1(\alpha) &= \sum_{i=1}^{N} \alpha_i P_{1i} \quad \text{where } P_{1i} = P_{1i}^T > 0 \\ P_2(\alpha) &= \sum_{i=1}^{N} \alpha_i P_{2i} \quad \text{where } P_{2i} = P_{2i}^T > 0 \end{aligned} \tag{21}$$

Robust stability condition (19) is in this case linear with respect to α (there are no products of matrices depending on α), therefore it is equivalent to

$$v(t)^T W(\alpha_i) v(t) < 0\,,\ i = 1,\ldots,N \tag{22}$$

where $\sum_{i=1}^{N} \alpha_i = 1,\ \alpha_i \geq 0$.

Note, that robust stability condition (22) is in LMI form for stability analysis – for unknown matrices $N_1, N_2, N_3, P_{1i}, P_{2i}$. For robust PID controller design, where unknown controller parameter matrices k_p, k_i, k_d are to be found, inequality (22) turns to bilinear matrix inequality (BMI), which can be solved either directly using some BMI solver or through linearization of nonlinear terms, see e.g. (deOliveira et al.

2000). We have applied the former approach and solved BMI (22) via PENBMI solver with YALMIP interface.

4. EXAMPLE

In this section the proposed approach to design a robust PSD controller is illustrated on the example.

Consider uncertain system (1), (2) with 3 states, 2 inputs and 2 outputs with nominal model

$$A_0=\begin{bmatrix}0.6 & 0.0097 & 0.0143\\ 0.012 & 0.945 & 0.0049\\ -0.0047 & 0.01 & 0.46\end{bmatrix},\quad B_0=\begin{bmatrix}0.0425 & 0.0053\\ 0.0052 & 0.01\\ 0.0024 & 0.0474\end{bmatrix}$$

$$C=\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}$$

and uncertainty matrices

$$A_u=\begin{bmatrix}0.004 & 0 & 0\\ 0 & 0.055 & 0\\ 0 & 0.0005 & 0.0002\end{bmatrix},\ B_u=10^{-4}\begin{bmatrix}0.2 & 0\\ 0 & 1\\ 0 & 1\end{bmatrix}$$

The respective uncertain polytopic model vertices for (2) are:

vertex 1: $A_1=A_0+A_u, B_1=B_0+B_u$,

vertex 2: $A_2=A_0-A_u, B_2=B_0-B_u$

eigenvalues of vertex system matrices A_1, A_2 are:

$$eig(A_1):\begin{matrix}1.0004\\ 0.6032\\ 0.4606\end{matrix}\qquad eig(A_2):\begin{matrix}0.8905\\ 0.5951\\ 0.4602\end{matrix}$$

Vertex 1 corresponds to unstable system.

Performance index (8) is considered, with weighting matrices

$R=I_{2x2}, Q=0.1*I_{5x5}, S=0.1*I_{5x5}$.

The aim is to find PSD controller parameter matrices k_p, k_i, k_d (of dimensions 2x2) so that the closed loop system (6) is robustly stable with guaranteed cost. The outputs for summation and difference part of control algorithm are $C_i=C,\ C_d=C$.

PSD controller has been designed by solving (19) as BMI. Two alternative PSD controllers have been computed:

using parameter dependent Lyapunov function (11) denoted as PQS

using simple quadratic Lyapunov function, with the same Lyapunov function matrices P_1, P_2 in (11) for the whole uncertainty domain; this case is computed for comparison and denoted as QS

The obtained results are summarized in Tab.1

	PQS	QS
k_p	$\begin{bmatrix}-5.6450 & -0.2486\\ -0.6602 & -5.257\end{bmatrix}$	$\begin{bmatrix}-3.7366 & -0.2483\\ -0.5043 & -3.2184\end{bmatrix}$
k_i	$\begin{bmatrix}-2.1584 & -0.1348\\ -0.4630 & -2.4361\end{bmatrix}$	$\begin{bmatrix}-1.6671 & -0.1716\\ -0.3897 & -1.8401\end{bmatrix}$
k_d	$\begin{bmatrix}1.2917 & -0.1006\\ 0.0473 & 1.2494\end{bmatrix}$	$\begin{bmatrix}0.1044 & -0.0085\\ 0.0174 & 0.0731\end{bmatrix}$
Closed-loop system eigenvalues		
Vertex1	0.4054 0.5990 0.7277 0.8415 0.9968	0.4721 0.6962 0.7126 0.8715 0.9968
Vertex2	0.4052 0.5788 0.7412 0.8433 **0.8846**	0.4713 0.6537 0.7500 0.8729 0.8837

Tab.1 PSD controller design results for parameter dependent Lyapunov function (PQS) and quadratic Lyapunov function (QS).

Step responses obtained from simulation in Simulink for vertices A_1 and A_2 are shown in Fig. 1 and 2.

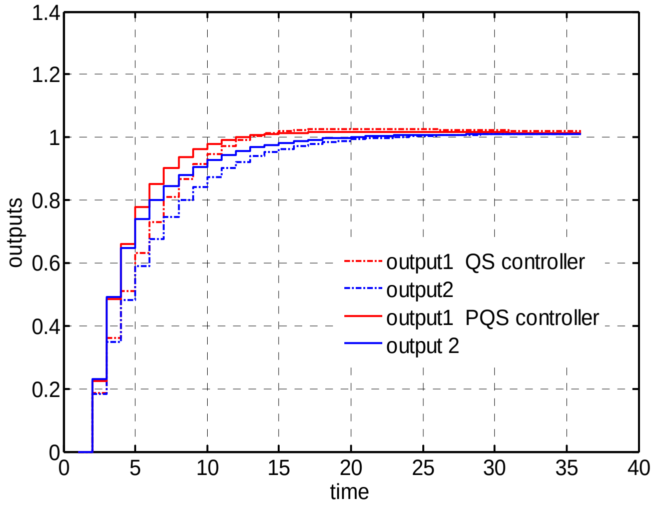


Fig.1 Comparison of closed-loop step responses for PSD controllers QS and PQS in vertex A_1, B_1 .

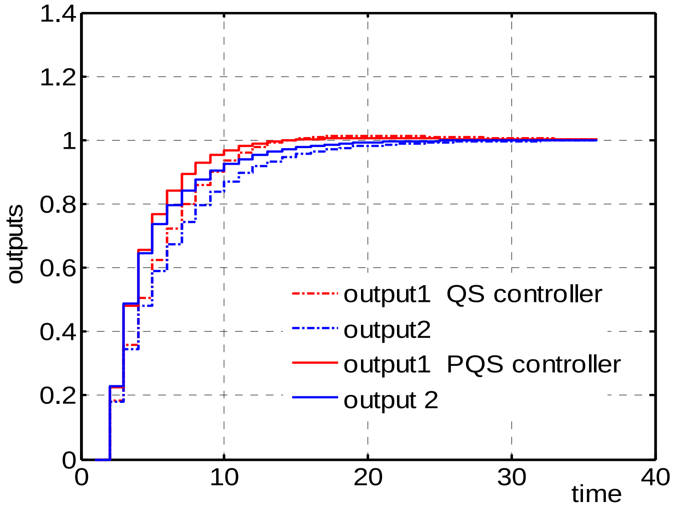


Fig.2 Comparison of closed-loop step responses for PSD controllers QS and PQS in vertex A_2, B_2.

From Fig.1 and 2 it can be seen that the closed-loop dynamics favours parameter dependent Lyapunov function based design over the quadratic one.

The respective control inputs are shown in Fig.3.

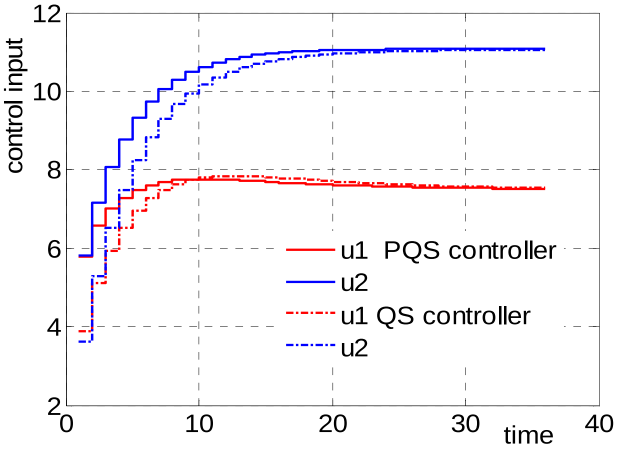


Fig.3 Comparison of control inputs in vertex A_2, B_2

5. CONCLUSION

In the paper the novel PSD controller design procedure is presented, which is based on Lyapunov function with special term corresponding to time-delay part of control algorithm. The resulting robust stability condition is in BMI form, in the illustrating example the PSD controller design has been performed using BMI solver.

ACKNOWLEDGMENTS

The work has been supported by the Slovak Scientific Grant Agency, Grant No. 1/0544/09 and 1/0592/10. This support is very gratefully acknowledged.

REFERENCES

Boyd, S., L. El Ghaoui, E. Feron and V. Balakrishnan (1994). Linear matrix inequalities in system and control theory. *SIAM Studies in Applied mathematics*, Philadelphia.

Dettori, M. and C.W. Scherer (2000). New robust stability and performance conditions based on parameter dependent multipliers. In: *Proc. 39nd IEEE CDC, Sydney, Australia*.

Han, J. and R.E. Skelton (2003). An LMI optimization approach for structured linear controllers. In: *Proc. 42nd IEEE CDC, Hawaii, USA*, pp. 5143-5148.

Henrion, D., D. Arzelier and D. Peaucelle (2002). Positive polynomial matrices and improved LMI robustness conditions. In: *15th IFAC World Congress*, Barcelona, Spain.

Henrion, D, J. Löfberg, M. Kocvara, M. Stingl (2005) Solving polynomial static output feedback problems with PENBMI. LAAS-CNRS Research Report No. 05165, March 2005. In: *Proceedings of the joint IEEE Conference on Decision and Control and European Control Conference*, Sevilla, Spain, December 2005

deOliveira, M.C., J. Bernussou and J.C. Geromel (1999). A new discrete-time robust stability condition. *Systems and Control Letters*, **37**, pp. 261-265.

deOliveira, M.C., J.F. Camino and R.E. Skelton (2000). A convexifying algorithm for the design of structured linear controllers. In: *Proc. 39nd IEEE CDC, Sydney, Australia*, pp. 2781-2786.

Ge, Ming, Min-Sen Chiu, Qing-Guo Wang (2002). Robust PID controller design via LMI approach. *Journal of Process Control*, **12**, pp.3-13.

Gyurkovics, E., Takacs, T.(2000). Stabilisation of discrete-time interconnected systems under control constraints. *IEE Proceedings - Control Theory and Applications*, **147**, No. 2, 137-144.

Löfberg J. (2004). YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In:Proceedings of the CACSD Conference Taipei, Taiwan. http://control.ee.ethz.ch/ joloef/yalmip.php

Peaucelle, D., D. Arzelier, O. Bachelier and J. Bernussou (2000). A new robust *D*-stability condition for real convex polytopic uncertainty. *Systems and Control Letters*, **40**, pp. 21-30.

Rosinová, D. and V. Veselý (2003). Robust output feedback design of discrete-time systems – linear matrix inequality methods. In: *2th IFAC Conf. CSD'03* (CD-ROM), Bratislava, Slovakia.

Rosinová, D. and V. Veselý (2007). Decentralized PID controller design for uncertain linear system, Proceedings of 11th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Complex Systems Theory and Applications, Gdansk, Poland, July 2007.

Skelton, R.E., T. Iwasaki and K. Grigoriadis (1998). *A Unified Algebraic Approach to Linear Control Design*, Taylor and Francis.

Van Antwerp, J. G. and Braatz, R. D. (2000). A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, Vol. 10 pp. 363–385.

Veselý, V. (2003). Robust output feedback synthesis: LMI Approach. In: *2th IFAC Conference CSD'03* (CD-ROM), Bratislava, Slovakia.

Zheng Feng, Qing-Guo Wang, Tong Heng Lee (2002). On the design of multivariable PID controllers via LMI approach. *Automatica*, **38**, pp.517-526

http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.HomePage

The robust motion control of a robot manipulator

J. Kardoš *

** Faculty of Electrical Engineering and Information Technology, Slovak Univ. of Technology,
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
(Tel: +421-2-6029-1776; e-mail: jan.kardos@stuba.sk)*

Abstract: The contribution presents a verification of the fast and chattering-free robust variable structure fixed-target position control of the 2-DOF robot manipulator considering both the mutual interaction between the links and the gravitational forces influence as the signal disturbances. Numerical simulations in the joint space show the feasibility and effectiveness of the provided algorithm in the control of a complex mechatronic system.

1. INTRODUCTION

Any feasible control of the multi-DOF mechatronic system has to face the problem of an extreme variability of the plant parameters as well as a strong influence of the variable external forces (gravitational, Coriolis and centrifugal). The classical control methods are not able to manage satisfactory such a complex and difficult task at all. One of the promising control approaches is the contemporary variable structure control (VSC) theory with its specific and unique attribute – the sliding mode (sliding mode control – SMC) (Utkin et al. 1999), (Utkin 2002), (Kardoš 2009). The basic feature of the sliding mode is the high frequency oscillation of the actuating variable because of the switching principle in the control algorithm. In sliding mode, the system's phase trajectory is robust and independent of the parametric and external disturbances due to reserve in power.

Based on the VSC, the equivalent time sub-optimal control (ETSC) algorithm has been formulated for a single-DOF motion control system (Kardoš 2005). The main benefits of this control are a simple control structure, the fastest possible and overshoot-free response and the insensitivity to any (parametric and/or signal) type of disturbances. One of the problems of SMC, the chattering elimination, has been solved via the reaching law approach (Hung et al. 1993). The aim of this contribution is the implementation of the prospective ETSC algorithm in the control structure of a multi-DOF robot manipulator. Using the Euler-Lagrange formalism, a dynamic model of the robot manipulator has been derived (Kardoš 2010). For the purposes of this paper, the reduced model of a 2-DOF manipulator has been considered without loss of generality.

2. THE OUTLINE OF THE CONTROL ALGORITHM

The goal of the original time sub-optimal control (TSC) algorithm (Kardoš 2007) is the fast and overshoot-free positioning of the motion control system despite its variable dynamics. Let for the basic model of the generalized position q generator given in the phase space $(e, \dot{e})$ by the system of the differential equations

$$\begin{aligned} \frac{de}{dt} &= \dot{e} \\ \frac{d\dot{e}}{dt} &= -\frac{1}{T}\left(Ku + \dot{e}\right) \end{aligned} \tag{1}$$

the time sub-optimal control be described by the group of expressions

$$\begin{aligned} u &= M\,\mathrm{sgn}(F(\mathbf{e})) \\ F(\mathbf{e}) &= \dot{e} + \alpha e \\ \alpha &= \frac{1}{T_{\max}(1-\ln 2)} \end{aligned} \tag{2}$$

In (1) and (2), $e = q_d - q$ stands for the control error (q_d – desired position value), $\mathbf{e} = (e, \dot{e})^T$ corresponds to the system's error vector (the phase vector), $F(\mathbf{e})$ represents the linear switching function (a switching line with the slope α), u is the system's input (the control), $M > 0$ refers to the value of a natural limitation of the control u, K stands for the control channel gain and the time constant T represents the dominant variable parameter

$$T \in \langle T_{\min}, T_{\max} \rangle \tag{3}$$

Note that control (2) belongs to the switching type (discontinuous) VSC algorithms and that the majority of VSC prefers the linear switching function with its simple computation and realization. To avoid the chattering problem in a motion control system (the chattering denotes a low frequency oscillation of system variables in sliding mode due to presence of parasitic dynamics and non-linearities in real mechatronic systems), the reaching law modification of the original control algorithm (2) has been performed (Kardoš 2005). The main idea of the reaching law is to force the system's state (the representative point in the phase portrait) to reach the switching function using the prescription given by the differential equation

$$\frac{dF(\mathbf{e})}{dt} = -kF(\mathbf{e}) \tag{4}$$

where $1/k$ represents the time constant of the switching function exponential evolution (decrease). Equation (4) meets the sliding mode existence condition (Utkin et al. 1999) and assures the chattering elimination replacing the discontinuous control by its smooth equivalent (close to mean value) in the vicinity of the switching function. Assuming the non-oscillating behaviour of the controlled variable q, using (1) and (2), we obtain a linear continuous equivalent control u_{EQU}

$$u_{EQU} = \frac{1}{K}\left[kT_{\max}\dot{e} + \frac{1}{1-\ln 2}\left(\dot{e}\ln 2 + ke\right)\right] \tag{5}$$

Combining equation (5) with the control u limitation ($\mathrm{abs}(u) \le M$) in (2) yields the resultant equivalent time sub-optimal control (ETSC) (Kardoš 2005) in the form

$$u_{ETSC} = \begin{cases} u_{EQU} & \textit{for}\ \mathrm{abs}(u_{EQU}) < M \\ M\,\mathrm{sgn}(u_{EQU}) & \textit{for}\ \mathrm{abs}(u_{EQU}) \ge M \end{cases} \tag{6}$$

The recommended value of parameter k

$$\begin{aligned} k &>> 1 \\ k &>> \alpha \end{aligned} \tag{7}$$

corresponds with the requirement of the fast and accurate control.

Algorithm (6) guarantees the robust and near-to-time optimal control of a single-DOF mechatronic system (SISO) (1) with the parametric uncertainty (3) in time constant (Kardoš 2005), (Kardoš 2007) as well as the chattering elimination in control structure (Kardoš 2005), which implies the low energy consumption. Furthermore, it is evident that signals necessary for the control algorithm completion, i.e. the position signal and the velocity one (cf. (5)), are in mechatronic systems directly accessible.

3. CONTROL OF THE ROBOT MANIPULATOR

The implementation of the proposed control strategy in the robot manipulator positioning control requires the thorough analysis of both the mutual interaction among the manipulator's degrees of freedom and the interaction between any DOF and the environment (e.g. the friction, the influence of the gravitation etc). Due to robustness of the presented control algorithm, such an interaction can be viewed as the signal disturbance. Consequently, the robot manipulator representing a complex MIMO system can be decoupled to a set of SISO systems, one for each DOF. Thus, any DOF can be controlled by its individual control algorithm (6).

Based on the Euler-Lagrange formalism, let the robot manipulator's dynamics be described by the matrix differential equation (Kardoš 2010)

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} = \tau - \mathbf{B}\dot{\mathbf{q}} - \mathbf{c}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) \tag{8}$$

where **q** and $\boldsymbol{\tau}$ stand for the vectors of generalized coordinates and generalized driving forces or torques respectively, **J**(.) represents the inertia matrix, **B** denotes the diagonal matrix of viscous friction coefficients, **c**(.) corresponds to the vector of Coriolis and centrifugal forces and **g**(.) stands for the vector of gravitational forces.

To keep the controllability of the manipulator DOF's, for the i^{th} DOF, the limitation M_i of the control u_i (in this particular case a limitation of the driving torque τ_i as an element of the driving forces vector $\boldsymbol{\tau}$) in the corresponding ETSC algorithm (6) should satisfy the condition

$$M_i \ge \max\left\{B_i\dot{q}_i + c_i(\mathbf{q},\dot{\mathbf{q}}) + g_i(\mathbf{q}) + \sum_j j_{ij}(\mathbf{q})\ddot{q}_j\right\} \tag{9}$$

where B_i, c_i, g_i, j_{ij}, $\dot{q}_i$ and $\ddot{q}_j$ are the elements of the related matrices or vectors in (8). Note, that the values of elements in matrix **J** as well as the ones in vectors **c** and **g** are extremely variable for the period of the robot positioning, which makes the utilization of conventional control theory methods almost impossible. To get the values of the control boundary M_i, the maximal values of both the joint angular acceleration and the joint angular velocity should be known. These values are given by the particular industrial technology where the controlled robot manipulator is supposed to be utilized. Taking into consideration the requirement of the fastest possible motion as well as the boundaries of acceleration and velocity, the command (desired value q_{di}) in high-quality controllable robot positioning should have the form of S-curve.

For the control purposes, after the decoupling of the original controlled plant (8), the values of the i^{th} DOF parameters are given by the pair of expressions

$$K_i = \frac{1}{B_i}$$
$$T_{i\max} = \frac{j_{ii\max}}{B_i} \tag{10}$$

4. EXPERIMENTAL RESULTS

The provided control algorithm has been applied in a numerical model of the robot manipulator control. A corresponding kinematic structure of the two-link manipulator is depicted in Figure 1, with the positive orientation of joint variables q_i (i = 1, 2) indicated by arrows. Both the controlled manipulator and the control algorithm parameters are given in Table 1. The resultant control structure in Matlab-Simulink can be seen in Figure 2.

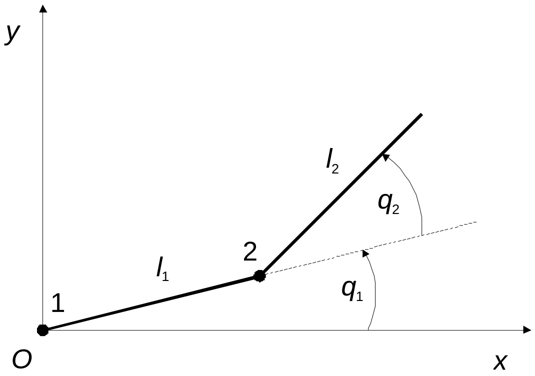


Fig. 1. Kinematic structure of the two-link manipulator

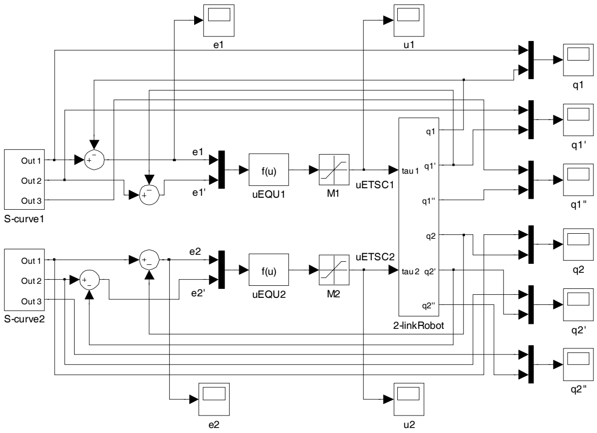


Fig. 2. Diagram of the Matlab control structure

The dynamic parameters of the robot endpoint trajectory (in the task space) equivalent to the angular values in the joint space are as follows: the maximal value of the acceleration $a_{\max}$ = 1.767ms^{-2}, the maximal value of the velocity $v_{\max}$ = 1.767ms^{-1}. Such parameters represent a sufficiently dynamic motion in robotized technologies, particularly in mechatronic systems with a strong coupling between the DOF's and with extremely variable time constants (proportional to the square of the varying distance between the revolute joint and the gravity centre of any rotating mass).

Parameter	Joint number i	
	1	2
Length l_i of a link [m]	0.75	0.75
Total mass m_i of a link [kg]	30	32
Coefficient of the viscous friction B_i [kgms^{-1}]	2	2
Control channel gain K_i	0.5	0.5
Max. value of the moment of inertia $j_{ii\max}$ [kgm^2]	88.875	18
Max. value of the system's time constant $T_{i\max}$ [s]	44.4375	9
Joint angular velocity boundary $\dot{q}_{i\max}$ [rads^{-1}]	0.7854	0.7854
Joint angular acceleration boundary $\ddot{q}_{i\max}$ [rads^{-2}]	0.7854	0.7854
Driving torque (control) boundary M_i [Nm]	840	300
Control parameter α_i (2)	0.0733	0.3621
Control parameter k_i (5)	10000	10000

Tab. 1. Parameters of the 2-DOF robot control

Figures 3 to 8 show the control system behaviour in the time domain for a period of positioning between the starting position $q_1 = q_2 = 0$ and the target one $q_1 = q_2 = \pi/2$ (in radians) in both joints. The left column of figures corresponds to the first link of the manipulator, the right column to the second one. The perfect accuracy of positioning is evident in Figures 3 and 6, where the desired (S-curve) angular position q_{di} (i = 1, 2) and the controlled position q_i are depicted. The perfect tracking performance is assured by the high value of parameter k in (5). The related trapezoidal plots of the desired angular velocity $\dot{q}_{di}$ and the output velocity $\dot{q}_i$ in Figures 5 and 8 show the accuracy and robustness of the ETSC algorithm. In Figures 4 and 7 there is a time history of the driving torques τ_i in manipulator joints. The discontinuities in plots correspond with the intentional discontinuities in the motion acceleration. Neither of the driving torques exceeds the prescribed limitation M_i defined by (9) (cf. Table 1). A non-zero value of the driving torque in steady state represents in both joints the reaction of the driving force to the gravitational one (the second link is in a horizontal position).

To prove the robustness of the proposed control despite the significant coupling between the DOF's, a simulation with the constant mutual position of the robot links has been performed. Figures 9 to 14 show the system variables versus time plots in the case of the maximally stretched manipulator arm ($q_2 = 0$) during the whole period of motion (the initial angular position of the first link $q_1 = 0$, the target position $q_1 = \pi/2$). Again, the perfect accuracy and robustness of the proposed control is illustrated in Figures 9 and 12 with the angular position in the time domain. The stretched arm of the manipulator matches with the zero value of the angular position q_2. A minimal difference between the desired and the actual angular position of the second link can be seen in Figure 14 with the 900-times enlarged scale of the vertical axis.

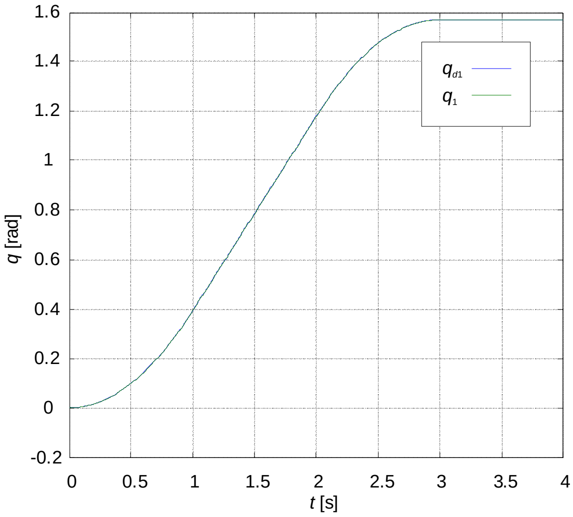


Fig. 3. First link: desired position q_{d1} and link position q_1

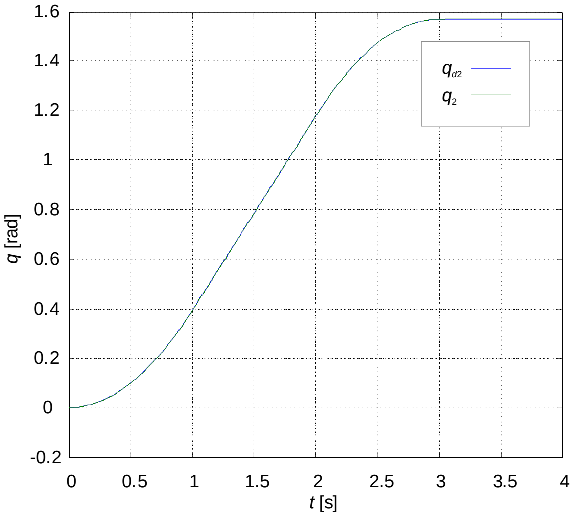


Fig. 6. Second link: desired position q_{d2} and link position q_2

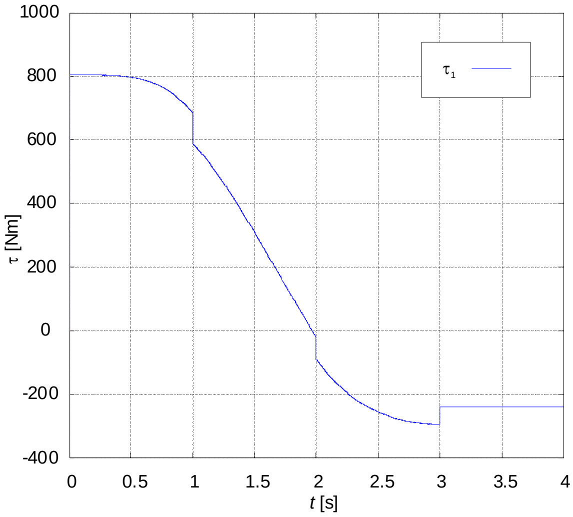


Fig. 4. First link: driving torque τ_1

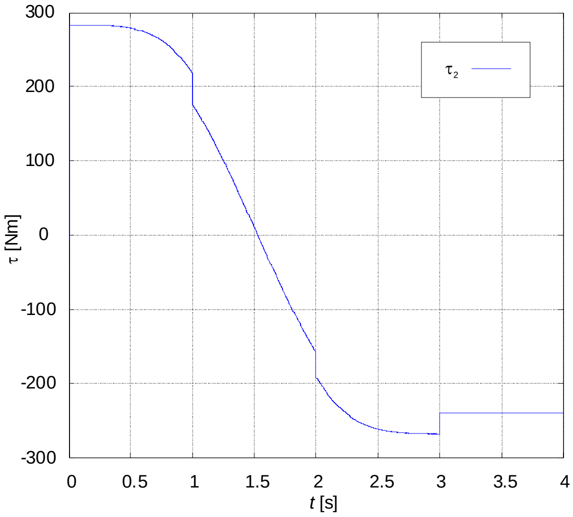


Fig. 7. Second link: driving torque τ_2

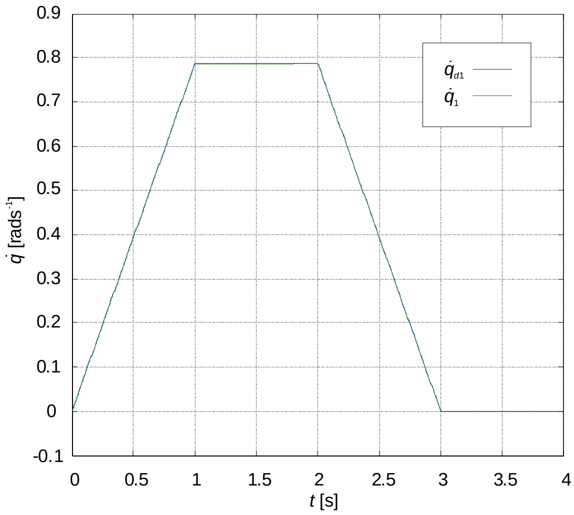


Fig. 5. First link: desired angular velocity $\dot{q}_{d1}$ and link angular velocity $\dot{q}_1$

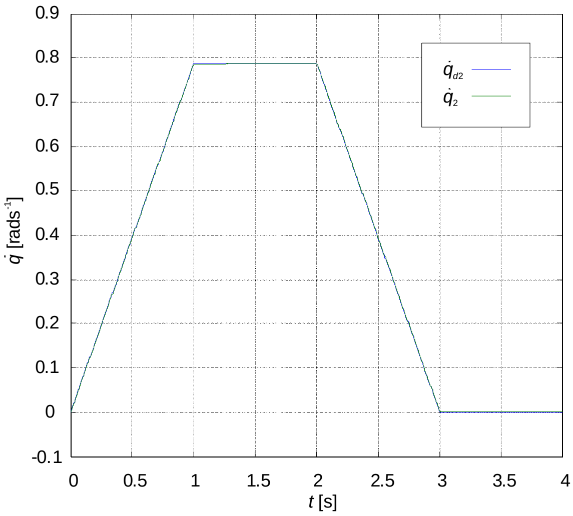


Fig. 8. Second link: desired angular velocity $\dot{q}_{d2}$ and link angular velocity $\dot{q}_2$

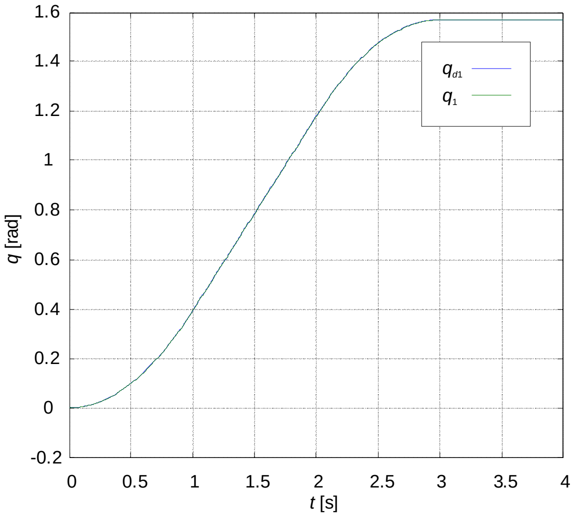


Fig. 9. First link, stretched arm: desired position q_{d1} and link position q_1

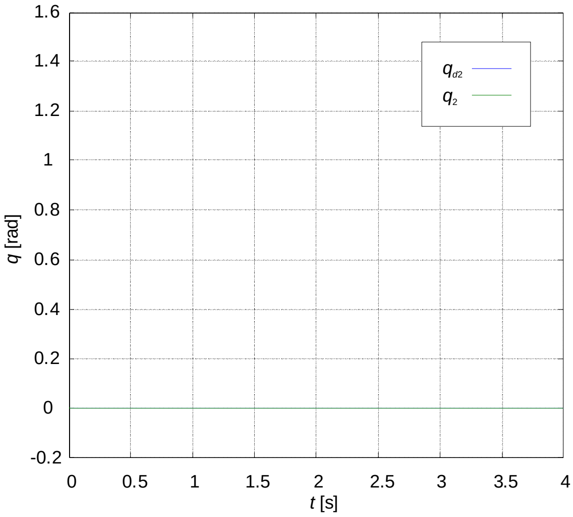


Fig. 12. Second link, stretched arm: desired position q_{d2} and link position q_2

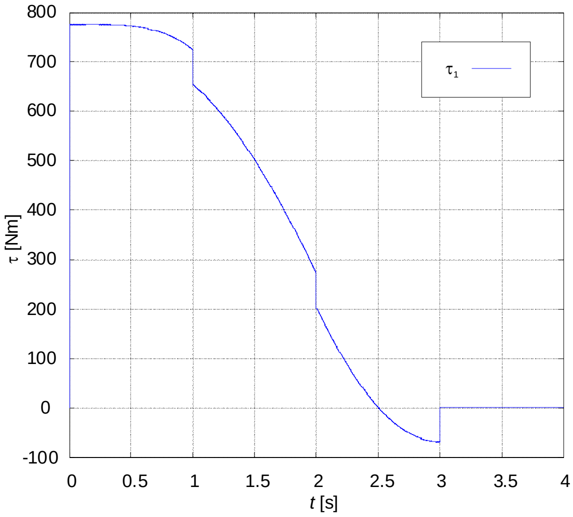


Fig. 10. First link, stretched arm: driving torque τ_1

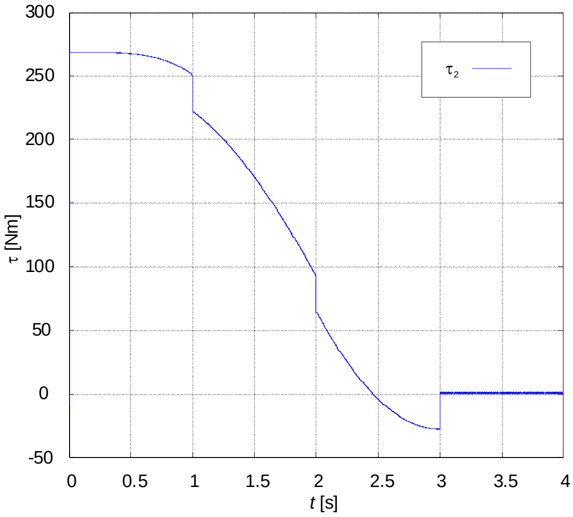


Fig. 13. Second link, stretched arm: driving torque τ_2

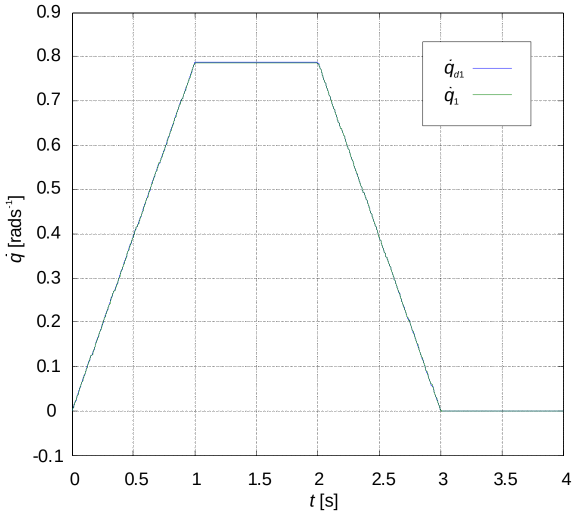


Fig. 11. First link, stretched arm: desired angular velocity $\dot{q}_{d1}$ and link angular velocity $\dot{q}_1$

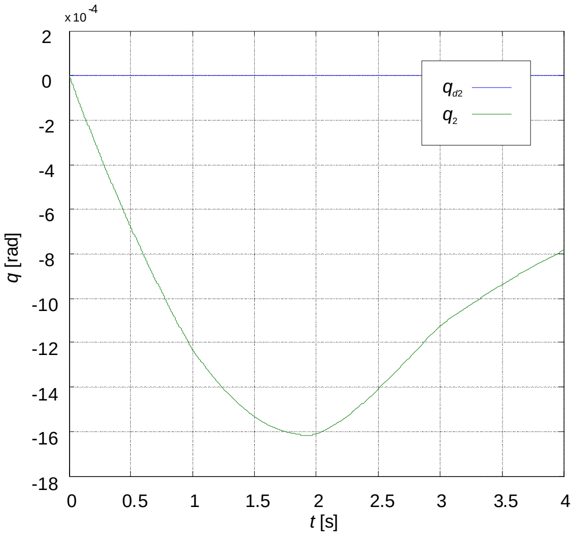


Fig. 14. Second link, stretched arm, detailed view: desired position q_{d2} and link position q_2

Notice the zero value of driving torques in steady state in Figures 10 and 13. This is the consequence of the vertical final position of the stretched link pair and therefore the zero influence of the gravitational forces. The wide interval of the driving torque variation in the second DOF, mirroring the torque in the first joint in spite of the constant mutual position between the links, shows the influence of the first link motion to the second link and proves the robustness of the control algorithm.

5. CONCLUSIONS

In this paper, the robustness and accuracy of a motion control algorithm based on the VSC theory – the equivalent time sub-optimal control – is verified and illustrated by the numerical simulation of a multi-DOF control system. Both the dynamic (tracking) and the steady-state accuracy have been achieved despite the enormous influence of the mechanical coupling among the DOF's of the robot manipulator. It has been demonstrated, that the proposed method assures the robustness against the signal as well as the parametric disturbances. The key to this method is the sliding mode control combined with the reaching law approach. The simple implementation of the control algorithm, given by the linear combination of the mechatronic system's straightforwardly accessible phase variables, represents an additional benefit of the presented method.

ACKNOWLEDGMENTS

The work presented in this contribution has been supported by the Grant Agency of Ministry of Education and Academy of Science of Slovak Republic VEGA under Grant No. 1/0690/09.

REFERENCES

Hung, J.Y., W. Gao and J.C. Hung (1993). Variable Structure Control: A Survey. *IEEE Transactions on Industrial Electronics* **40**, 2-22.

Kardoš, J. (2005). Reaching law modification of the time sub-optimal variable structure control. *International Journal of Mechanics and Control* **6**, 39-49.

Kardoš, J. (2007). *Theory of Variable Structure Systems and the Time Sub-Optimal Control* (in Slovak). HMH Press. Bratislava.

Kardoš, J. (2009). Variable structure motion control systems (in Slovak). *Automatizace* **52**, 688-692.

Kardoš, J. (2010). The Simplified Dynamic Model of a Robot Manipulator. *Proceedings of the 18th International Conference – Technical Computing Bratislava 2010*. Bratislava.

Utkin, V.I. (2002). First Stage of VSS: People and Events. In: *Variable Structure Systems: Towards the 21st Century* (X. Yu and J.X. Xu, Ed.). pp. 1-32. Springer Verlag. Berlin.

Utkin, V.I., J. Guldner and J. Dhi (1999). *Sliding Mode Control in Electromechanical Systems.* Taylor & Francis. London.

Robust elimination lemma: sufficient condition for robust output feedback controller design

V. Veselý, D. Rosinová

Slovak University of Technology
Faculty of Electrical Engineering and Information Technology
Institute of Control and Industrial Informatics
Bratislava, Slovakia
(e-mail: danica.rosinova@stuba.sk, vojtech.vesely@stuba.sk)

Abstract: A linear algebra result known as Elimination lemma is frequently used in lot of filtering and control problems to transform products of unknown matrices to LMI form, however, the robust counterpart to elimination lemma is not known. In this paper, sufficient robust stability condition inspired by elimination lemma is developed and the respective robust static output feedback controller design procedure based on LMI formulation and solution is proposed. The proposed robust controller design procedure is computationally not demanding and is illustrated on examples.

1. INTRODUCTION

Linear algebraic result known as Elimination lemma plays an important role in the study of robust stability conditions for linear systems with polytopic uncertainties (Boyd et al. 1994, Skelton et al. 1998). The following matrix inequality often appears in robust control problem formulation

$$G_i + U_i X V_i + V_i^T X^T U_i^T < 0, \qquad i = 1,2,...,N \tag{1}$$

The matrices G_i, U_i, V_i, X may all depend on the control system parameters to be designed; X may represent control gain which is the same for the whole uncertainty domain, when robust controller is concerned. Elimination lemma enables to eliminate unknown matrix X from (1) when N = 1, thus simplifying the resulting design inequality, which then often turns to LMI. Unfortunately, for $N > 1$, which is the case of uncertain polytopic linear system, the elimination lemma cannot be directly extended, (deOliveira 2005; Vesely, et al 2009). Moreover, it is not clear if such counterpart in the form of necessary and sufficient condition can be found since the class of structured linear control problems such as decentralized control and simultaneous static output feedback (SOF) belongs to NP hard problems as have been proven in (Blondel and Tsitsiklis, 1997). Nevertheless, various techniques have been developed to reformulate the problem as LMI one using certain convex approximation as linearizing or convexifying functions (deOliveira et al., 2000; Han and Skelton, 2003, Veselý, 2003; Rosinová and Veselý, 2003). The problem remains in linearizing the off-diagonal terms, since in this case, the upper bound based linearization formulas are not quite suitable to receive workable results.

In this paper, sufficient condition for (1) is developed, which can be advantageously used for robust static output feedback controller design. The respective control design procedure, which is computationally not demanding, is presented. Section 2 brings problem formulation and preliminaries. In Section 3, the sufficient condition for (1) is developed, in which the unknown matrix X is eliminated from off-line terms of the respective matrix determining robust stability condition. The corresponding robust control design procedure is proposed and in Section 4 it is illustrated on two examples.

2. PRELIMINARIES AND PROBLEM FORMULATION

The robust static output feedback control design problem is formulated in this section and the respective sufficient robust stability condition in the form (1) is presented. Consider the class of linear uncertain continuous or, alternatively, discrete-time systems described by convex polytopic model:

$$\begin{aligned} \delta x(t) &= A(\alpha)x(t) + B(\alpha)u(t) \\ y(t) &= Cx(t) \end{aligned} \tag{2}$$

where

$\delta x(t) = \dot{x}(t)$ for continuous - time system
$\delta x(t) = x(t+1)$ for discrete - time system

$x(t) \in R^n, u(t) \in R^m, y(t) \in R^l$ are state, control and output vectors respectively; uncertain model matrices $A(\alpha), B(\alpha)$ are from convex polytopic uncertainty domain given by polytope vertices $A_i \in R^{nxn}, B_i \in R^{nxm}$, $i = 1,...,N$:

$(A(\alpha), B(\alpha)) \in S,$

$$S = \left\{ (A(\alpha), B(\alpha)) : \; A(\alpha) = \sum_{i=1}^{N} \alpha_i A_i, B(\alpha) = \sum_{i=1}^{N} \alpha_i B_i, \sum_{i=1}^{N} \alpha_i = 1, \; \alpha_i \geq 0 \right\} \tag{3}$$

Consider a static output feedback control law

$$u(t) = Fy(t) = FCx(t) \tag{4}$$

and the respective closed loop uncertain system

$$\delta x(t) = A_C(\alpha)x(t) \tag{5}$$

where

$$A_C(\alpha) \in \left\{ \sum_{i=1}^{N} \alpha_i A_{Ci}, \quad \sum_{i=1}^{N} \alpha_i = 1, \quad \alpha_i \geq 0 \right\}. \tag{6}$$

$$A_{Ci} = A_i + B_i FC$$

To study the stability of uncertain linear system (2), the parameter-dependent quadratic Lyapunov function is used

$$V(t) = x(t)^T P(\alpha) x(t) \tag{7}$$

and the respective robust stability condition is considered in compliance with (Oliveira et al. 1999).

Definition 1

System (5) is *robustly stable* in the convex uncertainty domain (6) with parameter-dependent Lyapunov function (7) if and only if there exists a matrix $P(\alpha) = P(\alpha)^T > 0$ such that

$$\begin{aligned} r_{12}P(\alpha)A_C(\alpha) + r_{12}^* A_C^T(\alpha)P(\alpha) + r_{11}P(\alpha) + \\ + r_{22}A_C^T(\alpha)P(\alpha)A_C(\alpha) < 0 \end{aligned} \tag{8}$$

for all α such that $A_C(\alpha)$ is given by (6) and r_{11}= 0, r_{12}= 1, r_{22}= 0 for a continuous-time system; r_{11}= -1, r_{12}= 0, r_{22}= 1 for a discrete-time system.

In the following we consider Lyapunov matrix $P(\alpha)$ in the form

$$P(\alpha) = \sum_{i=1}^{N} \alpha_i P_i \text{ where } P_i = P_i^T > 0\,,\ j = 1,\ldots,N \tag{9}$$

Robust static output feedback control design aims at finding an output feedback gain matrix *F* for control law (4) so that the uncertain closed loop system (5) is robustly stable.

Recall a sufficient robust stability condition proposed in (Peaucelle et al., 2000), which has been favoured in comparison of several available results (Grman et al. 2005)

Lemma 1

If there exist matrices $E \in R^{nxn}, H \in R^{nxn}$ and *N* symmetric positive definite matrices $P_i \in R^{nxn}$ such that for all *i*=1,…,*N*:

$$\begin{bmatrix} r_{11}P_i + A_{Ci}^T E^T + EA_{Ci} & r_{12}P_i - E + A_{Ci}^T H \\ r_{12}^* P_i - E^T + H^T A_{Ci} & r_{22}P_i - (H + H^T) \end{bmatrix} < 0 \tag{10}$$

$$A_{Ci} = A_i + B_i FC$$

then system (5) is robustly stable, where r_{11}= 0, r_{12}= 1, r_{22}= 0 for a continuous-time system; r_{11}= -1, r_{12}= 0, r_{22}= 1 for a discrete-time system.

Matrix inequality (10) is in the form of LMI for robust stability analysis with for unknown matrices E, H, P_i. On the contrary, for robust control design (10) is no more LMI since in this case, *F* is unknown matrix as well as E, H, P_i, and these unknown matrices appear in bilinear terms. One possibility to cope with nonlinear (bilinear) terms is to use bilinear matrix inequality (BMI) solvers; this approach, however, has its limitations (e.g dimension of problem, case sensitivity). To improve numerical tractability, there is an effort to transform (10) to LMI, the frequent approach is to employ linearization (deOliveira et al. 2000). In this paper the upper bound on the left hand side of (10) is used, based on the following well known matrix inequality.

Lemma 2

For any $\varepsilon_i > 0$ following inequalities hold for any matrices U_i, V_i, X

$$\begin{aligned} U_i X V_i + V_i^T X^T U_i^T \leq \varepsilon_i^{-1} U_i U_i^T + \varepsilon_i V_i^T X^T X V_i, \\ i = 1,2,\ldots,N \end{aligned} \tag{11}$$

Lemma 2 immediately follows from inequality

$$\left(\frac{U_i^T}{\sqrt{\varepsilon_i}} - \sqrt{\varepsilon_i} X V_i \right)^T \left(\frac{U_i^T}{\sqrt{\varepsilon_i}} - \sqrt{\varepsilon_i} X V_i \right) \geq 0, \qquad i = 1,2,\ldots,N \tag{12}$$

which holds for any $\varepsilon_i > 0$ and any matrices U_i, V_i, X .

A closed loop performance is assessed considering the *guaranteed cost* notion. The quadratic cost function is used.

$$J_c = \int_0^\infty [x(t)^T Q x(t) + u(t)^T R u(t)] dt$$

for a continuous-time and

$$J_d = \sum_{k=0}^{\infty} [x(t)^T Q x(t) + u(t)^T R u(t)]$$

for a discrete-time system, (13)

where $Q \in R^{nxn}, R \in R^{mxm}$ are symmetric positive definite matrices.

Control law (4) is called guaranteed cost control when there exist a feedback gain matrix *F* and a constant J_0 such that

$$J \leq J_0 \tag{14}$$

holds for the closed loop system (5), (6); J_0 is the guaranteed cost.

Extending robust stability condition (10) by guaranteed cost requirement as known from LQ theory, the robust stability condition with guaranteed cost is obtained in the form

$$\begin{bmatrix} r_{11}P_i + A_{Ci}^T E^T + EA_{Ci} + Q + C^T F^T RFC & r_{12}P_i - E + A_{Ci}^T H \\ r_{12}^* P_i - E^T + H^T A_{Ci} & r_{22}P_i - (H + H^T) \end{bmatrix} < 0 \tag{15}$$

Inequality (15) is LMI for stability analysis, i.e. for unknown matrices P_i, E, H . In the case of controller design, where *F* is also unknown, the bilinear terms appears in (15) both in diagonal and off-diagonal terms. Nonlinear diagonal terms in (15) can be treated by existing convexifying approaches as in (deOliveira et al., 2000; Han and Skelton, 2003, Veselý et al, 2009).

The respective potential convexifying function for terms as X^{-1} and XWX has been proposed in the linearizing form (Han and Skelton, 2003):

- The linearization of $X^{-1} \in R^{nxn}$ about the value $X_k > 0$ is

$$\Phi(X^{-1}, X_k) = X_k^{-1} - X_k^{-1}(X - X_k)X_k^{-1} \tag{16}$$

- The linearization of $XWX \in R^{nxn}$ about X_k is

$$\Psi(XWX, X_k) = -X_k W X_k + XWX_k + X_k WX \ . \tag{17}$$

Both functions defined in (16) and (17) meet one of the basic requirements on convexifying function: to be equal to the original nonconvex term if and only if $X_k = X$. However, the question how to choose the appropriate “nice” convexifying function remains still open.

In fact, linearization (16) and (17) is based on using upper bounds on bilinear terms, which is suitable for treating diagonal terms in (15). As soon as the bilinear terms in the off-diagonal part of testing matrix are to be linearized, the upper bounds based approaches are no more appropriate and a different way to linearization must be found. One possible way to transform robust stability condition to LMI is proposed in the next section.

In the sequel, $X > 0$ denotes positive definite matrix; * in matrices denotes the respective transposed term to make the matrix symmetric; *I* denotes identity matrix and 0 denotes zero matrix of the respective dimensions.

3. ROBUST SOF CONTROLLER DESIGN PROCEDURE

In this section, the novel static output feedback design procedure is proposed based on sufficient robust stability condition, which yields LMI formulation for controller design.

The following corollary of Lemma 2 provides the bound (sufficient condition) which will be used below.

Corollary 1

If there exists $\varepsilon_i > 0$ such that

$$G_i + \varepsilon_i^{-1} U_i U_i^T + \varepsilon_i V_i^T X^T X V_i < 0, \quad i = 1,2,...,N \tag{18}$$

then

$$G_i + U_i X V_i + V_i^T X^T U_i^T < 0, \qquad i = 1,2,...,N \tag{19}$$

Sufficient robust stability condition is then formulated in Theorem 1.

Theorem 1

If there exist matrices $E \in R^{nxn}, H_1 \in R^{nxn}, H_2 \in R^{nxn}$, N symmetric positive definite matrices $P_i \in R^{nxn}$ and $\varepsilon_i > 0$ such that for all i = 1,…,*N*:

$$\widetilde{G}_i \le 0 \tag{20}$$

where $\widetilde{G}_i = \begin{bmatrix} g_{i11} & g_{i12} \\ g_{i21} & g_{i22} \end{bmatrix}$

$$g_{i11} = r_{11}P_i + A_{Ci}^T E^T + EA_{Ci} + Q + C^T F^T RFC + \varepsilon_i^{-1} A_{Ci}^T A_{Ci}$$
$$g_{i12} = r_{12}P_i - E + A_{Ci}^T H_1 - \varepsilon_i^{-1} A_{Ci}^T$$
$$g_{i21} = r_{12}^* P_i - E^T + H_1^T A_{Ci} - \varepsilon_i^{-1} A_{Ci}$$
$$g_{i22} = r_{22}P_i - (H_1 + H_1^T) + \left(\rho_X^2 \varepsilon_i + \varepsilon_i^{-1}\right)I \quad (13)$$

and ρ_X is a chosen positive constant,

then closed loop uncertain system (5), (6) is stable with guaranteed cost.

Proof

Robust stability condition (10) can be for $H = H_1 + H_2$ rewritten in the form (19) where

$$G_i = \begin{bmatrix} r_{11}P_i + A_{Ci}^T E^T + EA_{Ci} + Q + C^T F^T RFC & r_{12}P_i - E + A_{Ci}^T H_1 \\ r_{12}^* P_i - E^T + H_1^T A_{Ci} & r_{22}P_i - (H_1 + H_1^T) \end{bmatrix}$$

$$U_i^T = \begin{bmatrix} A_{Ci} & -I \end{bmatrix}, \ X = H_2, \quad V_1 = \begin{bmatrix} 0 & I \end{bmatrix}. \tag{21}$$

Let us apply now Corollary 1 to the matrices defined in (21).

$$U_i U_i^T = \begin{bmatrix} A_{Ci}^T \\ -I \end{bmatrix} \begin{bmatrix} A_{Ci} & -I \end{bmatrix} = \begin{bmatrix} A_{Ci}^T A_{Ci} & -A_{Ci}^T \\ -A_{Ci} & I \end{bmatrix}$$

$$V_i^T H_2^T H_2 V_i = \begin{bmatrix} 0 \\ I \end{bmatrix} H_2^T H_2 \begin{bmatrix} 0 & I \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & H_2^T H_2 \end{bmatrix} \tag{22}$$

H_2 is any matrix, let us consider it as diagonal: $H_2 = \rho_X I$, where ρ_X is a chosen constant.

Substituting (21) and (22) into (18), the sufficient robust stability condition (20) is obtained. □

Note, that the nonlinear (bilinear) terms appear only in the diagonal part of matrix $\widetilde{G}_i$ in robust stability condition (20).

The robust SOF controller can be now designed using (20) as described above, by the following proposed procedure.

Procedure for robust SOF controller design:

1. Choose an upper bound constant r_0 for

$$0 \le P_i \le r_0 I$$

2. Choose $H_1 \approx 0.8 r_0$ and $\rho_X \approx 0.04 r_0$.

3. Choose starting value of $\varepsilon_i > 0$

4. Apply linearization of the diagonal terms using (17) and the respective iterative procedure and solve LMI (20) for unknown matrices P_i, E, H, F .

If (20) is infeasible, change the constant $\varepsilon_i > 0$ and repeat steps 3. and 4.

The outlined procedure requires iterative computation in steps 3. and 4., in fact it is one dimensional search for appropriate value of $\varepsilon_i > 0$ so that the outlined procedure provides feasible solution of (20).

4. EXAMPLES

In this section the proposed robust controller design procedure is illustrated on two examples. The previous result for robust SOF design with guaranteed cost (Veselý et al, 2009) is recalled and used for comparison.

Sufficient robust stability condition for uncertain system (5), (6) with guaranteed cost can be formulated in the following form (Veselý et al, 2009):

If there exist matrices $E \in R^{nxn}, H \in R^{nxn}$, N symmetric positive definite matrices $P_i \in R^{nxn}$ and $\gamma > 0$ such that for all $i = 1,\ldots,N$:

$$\begin{bmatrix} w_{11} & r_{12}P_i - H & A_{Ci}^T \\ r_{12}P_i - H^T & r_{22}P_i - 2\gamma I + I & 0 \\ A_{Ci} & 0 & -\gamma^2 I \end{bmatrix} \le 0 \qquad (23)$$

where

$$w_{11} = r_{11}P_i + A_{Ci}^T E^T + EA_{Ci} + Q + C^T F^T RFC$$

then closed loop uncertain system (5), (6) is stable with guaranteed cost.

In inequality (23) scalar parameter $\gamma > 0$ is to be appropriately chosen.

Example 1

Consider uncertain system with 10 states, 2 inputs and 4 outputs with nominal model described by matrices A_0, B_0, C and uncertainty matrices $A_{u1}, A_{u2}, B_{u1}, B_{u2}$.

$$A_0 = \begin{bmatrix} 0 & -0.2148 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1.0142 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2605 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -0.9107 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.1639 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -0.8137 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.2279 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.8251 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A_{u1} = \begin{bmatrix} 0 & -0.025 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.1395 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0938 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2911 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0188 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0208 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1173 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_{u2} = \begin{bmatrix} 0 & 0.0125 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0594 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0938 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2911 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0188 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0208 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.0333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.1173 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0.3148 & 0 \\ 0.0478 & 0 \\ 0 & -0.1028 \\ 0 & -0.0091 \\ -0.0841 & 0 \\ -0.0287 & 0 \\ 0 & 0.3676 \\ 0 & 0.2448 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$B_{u1} = \begin{bmatrix} 0.0625 & 0 \\ -0.0798 & 0 \\ 0 & -0.0462 \\ 0 & 0.0449 \\ 0.0016 & 0 \\ 0.0072 & 0 \\ 0 & 0.0770 \\ 0 & -0.0050 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, B_{u2} = \begin{bmatrix} -0.0094 & 0 \\ 0.0151 & 0 \\ 0 & 0.0019 \\ 0 & -0.003 \\ -0.0121 & 0 \\ -0.03 & 0 \\ 0 & -0.0640 \\ 0 & 0.0189 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The respective uncertain polytopic model (2), (3) for the above matrices has 4 vertices given by:

$$A_i = A_0 + q_1 A_{u1} + q_2 A_{u2}, B_i = B_0 + q_1 B_{u1} + q_2 B_{u2}, \qquad (24)$$

where $q_1, q_2 \in \{-1, 1\}$.

Output matrix is

$$C = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Cost function matrices are: $Q = 0.1I, R = I$.

The robust controller design procedure proposed in the end of Section 3 has been realized for the above described uncertain system. Upper bound on Lyapunov matrix P_i has been chosen as $r_0 = 1000$, $H_1 = 0.8r_0$ and $\rho_X = 0.04r_0$. Results obtained using new design procedure and results corresponding to a solution of (23) are summarized as follows.

New procedure based on (20):

values of ε providing a feasible solution to (20): $\varepsilon \in \langle 0.006, 0.016 \rangle$

SOF gain matrix for $\varepsilon = 0.0075$:

$$F_1 = \begin{bmatrix} -1.1873 & 0 & -0.1462 & 0 \\ 0 & -0.7996 & 0 & -0.0586 \end{bmatrix}$$

maximal closed-loop system eigenvalue = -0.0397

maximal eigenvalue of Lyapunov matrices = 999,899.

Results for previously designed procedure – solution of (23):

values of γ providing a feasible solution to (23): $\gamma \in \langle 0.51, 4.71 \rangle$

SOF gain matrix for $\gamma = 1.51$:

$$F_2 = \begin{bmatrix} -1.9138 & 0 & -0.2725 & 0 \\ 0 & -1.4172 & 0 & -0.1026 \end{bmatrix}$$

maximal closed-loop system eigenvalue = -0.0496

maximal eigenvalue of Lyapunov matrices = 942,388.

Example 2

Consider double integrator described by a nominal model

$$A_0 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B_0 = \begin{bmatrix} 0 \\ 0.75 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

with uncertainties:

$$A_{u1} = \begin{bmatrix} 0 & -0.1 \\ 0 & 0 \end{bmatrix}, A_{u2} = \begin{bmatrix} 0 & 0.1 \\ 0 & 0 \end{bmatrix}, B_{u1} = \begin{bmatrix} 0 \\ -0.25 \end{bmatrix}, B_{u2} = \begin{bmatrix} 0 \\ 0.25 \end{bmatrix}$$

Analogically as in Example 1, uncertain system is described by a polytope with four vertices given by (24).

Cost function matrices are: $Q = 0.1I, R = I$.

Parameters for new design procedure has been the same as in Example 1: Upper bound on Lyapunov matrix P_i chosen as $r_0 = 1000$, $H_1 = 0.8r_0$ and $\rho_X = 0.04r_0$. Results obtained using new design procedure and results corresponding to a solution of (23) are summarized below.

New procedure based on (20):

values of ε providing a feasible solution to (20): $\varepsilon \in \langle 0.001, 0.026 \rangle$

SOF gain matrix for $\varepsilon = 0.0055$:

$$F_1 = \begin{bmatrix} -0.3716 & -1.9369 \end{bmatrix}$$

maximal closed-loop system eigenvalue = -0.2275

maximal eigenvalue of Lyapunov matrices = 866.95.

Results for previously designed procedure – solution of (23):

values of γ providing a feasible solution to (23): $\gamma \in \langle 0.51, 10.51 \rangle$

SOF gain matrix for $\gamma = 2.31$:

$$F_2 = \begin{bmatrix} -22.4311 & -266.0514 \end{bmatrix}$$

maximal closed-loop system eigenvalue = -0.0675

maximal eigenvalue of Lyapunov matrices = 889,52.

In both examples the proposed procedure has been successfully applied to compute the robust control gain matrix including guaranteed cost requirement.

5. CONCLUSION

Robust static output feedback control design procedure has been proposed based on new developed sufficient robust stability condition. This condition is in the form of matrix inequality, where the off diagonal terms of testing matrix are linear with respect to unknown matrices and bilinear terms in the matrix diagonal can be readily linearized using upper bound linearization approach. The proposed procedure includes scalar parameters to be chosen by a designer, the proposed values of these parameters have been tested on various examples, and two of them are shown in Section 4.

ACKNOWLEDGMENTS

The work has been supported by the Slovak Scientific Grant Agency, Grant No. 1/0544/09. This support is very gratefully acknowledged.

REFERENCES

Blondel, V. and J.N. Tsitsiklis (1997). NP-hardness of some linear control design problems. *SIAM J. Control Optim.*, **35**, pp.2118-2127.

Boyd, S., L. El Ghaoui, E. Feron and V. Balakrishnan (1994). Linear matrix inequalities in system and control theory. *SIAM Studies in Applied mathematics*, Philadelphia.

deOliveira, M.C., J. Bernussou and J.C. Geromel (1999). A new discrete-time robust stability condition. *Systems and Control Letters*, **37**, pp. 261-265.

de Oliveira M.C., J.F. Camino and R.E. Skelton (2000). A convexifying algorithm for the design of structured linear controllers. *In: Proceedings of the 39 IEEE Conf on Decision and Control, Sydney*, 2000, pp. 2781-2786.

de Oliveira M.C. (2005). A robust version of the elimination lemma. In:16th Triennial IFAC World Congress, Prague, 2005, CD-ROM.

Grman L., D. Rosinová, V. Veselý and A. Kozáková (2005). Robust stability conditions for polytopic systems. *Int.J. of Systems Science,* 36 (N15), pp. 961-974.

Henrion D., J. Löfberg, M. Kocvara and M. Stingl (2005). Solving polynomial static output feedback problems with PENBMI. LAAS-CNRS Research Report No. 05165, March 2005. Proceedings of the joint IEEE Conference on Decision and Control and European Control Conference, Sevilla, Spain, December 2005.

Peaucelle, D., D. Arzelier, O. Bachelier and J. Bernussou (2000). A new robust *D*-stability condition for real convex polytopic uncertainty. *Systems and Control Letters*, **40**, pp. 21-30.

Rosinová, D. and V. Veselý (2003). Robust output feedback design of discrete-time systems – linear matrix inequality methods. In: *2th IFAC Conf. CSD'03* (CD-ROM), Bratislava, Slovakia.

Skelton, R.E., T. Iwasaki and K. Grigoriadis (1998). *A Unified Algebraic Approach to Linear Control Design*, Taylor and Francis.

Veselý, V., Rosinová D., and Kučera V. (2009). Robust controller design for polytopic systems via Robust Elimination Lemma. Journal of Electrical Eng. Vol.60, N1, 34-38.

Online design of SciLab/Xcos block schemes

Zoltán Janík and Katarína Žáková

Faculty of Electrical Engineering and Information Technology
Slovak University of Technology
Ilkovičova 3, 812 19 Bratislava
(Tel: +421 2 60 291 725; e-mail: zoltan.janik@stuba.sk)

Abstract: The paper presents a tool that supports design of simulation structures for online laboratories accomplished in SciLab/Xcos environment. For running such simulations, it is necessary to build a block scheme corresponding to a control of virtual or remote device. The presented tool offers a comfortable way of such solution. It's programmed in widely used technologies to ensure wide compatibility and platform independence. This application can be used as a supporting tool in virtual and remote laboratories.

1. INTRODUCTION

The importance of virtual and remote laboratories becomes more and more significant mainly in the area of technical education. They help students to understand basic problems because they illustrate and visualize controlled dynamics, and the necessity to exercise all design steps starting with the plant identification and ending with the evaluation of the control results achieved with a particular model. The advantage is that such laboratories are available to all interested users 24 hours a day and from any place. Students and interested users can access them via commonly used Web browser on any machine that has Internet connection.

Described web-based application supports block schemes created in SciLab/Xcos desktop application. SciLab is free software for numerical computation and simulation such as Matlab. It is available for all major operating systems: Linux, Mac OS and Windows. The Xcos is a free graphical editor that offers graphical user interface to design and run simulations of dynamical system models. Xcos is distributed together with SciLab.

2. TECHNOLOGY OVERVIEW

The created tool is based on XHTML, CSS and JavaScript language. The incorporated export and import is performed by PHP technology mainly through AJAX[1] requests. These technologies are selected in order to keep the widest availability of this tool. There are no special demands at client's side. The only thing that is required is compatible web browser (the application was tested mainly in FireFox 3.5 and Internet Explorer 8, but it should work with no limitations in IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome).

JavaScript library jQuery is used to speed up the development and simplify the source code. The library is used for the object manipulation, changing CSS[2] properties, visual functions and AJAX requests.

The application internally operates with XML and JSON[3] formats. Therefore the support of XML and JSON handling functions are required. In fact, these functions are natively supported by PHP version 5.2 or higher, thus it is the only requirement to the web server.

3. FRONT-END

The front-end of the application is demonstrated in Fig. 1. Our aim was to prepare a design that would be familiar for users using Matlab/Simulink or SciLab/Xcos environment. The whole application consists of 3 parts: main menu, block toolbar and canvas.

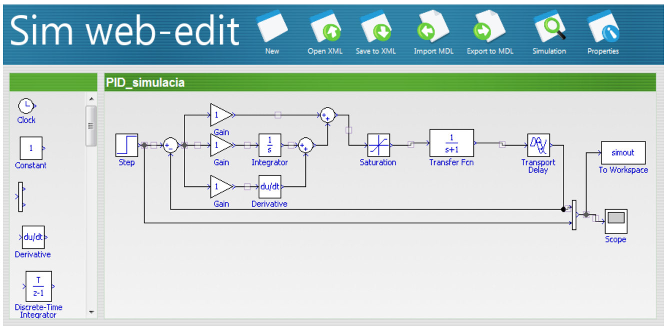


Fig. 1. Web editor interface

Block toolbar contains automatically generated list of all available blocks in the form as they appear in canvas. Each block is set to be draggable. After dragging starts, the clone of selected block is created in order it could be dropped into the canvas. The new block is inserted into canvas exactly in

[1] Asynchronous JavaScript and XML – it enables to exchange data with a server and update parts of a web page without reloading the whole page

[2] Cascading Style Sheets

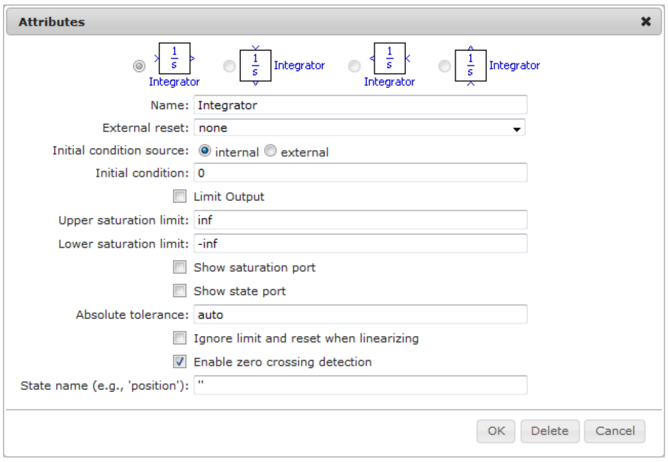
[3] JavaScript Object Notation – set of key-value pairs

place where it has been dropped. In the case that the block is dropped outside canvas, it is not considered as valid and the user can see only visual feedback of moving dropped block back to its original position in toolbar.

In the application we considered 2 types of blocks:

- standard blocks are blocks from the toolbar panel that are used for modeling of dynamical behavior of systems;
- “pseudo” blocks are blocks that are generated automatically by the application for establishing an additional connection feature. In this way we recognize the *node block* that is used for multiple line connection and the *line block* that enables to modify the position of line segment placed between two blocks.

Block attributes can be changed in the pop-up window that is displayed after clicking on any standard block in the canvas. The content of the attributes window is fully dynamic. It is loaded via AJAX request from other script that builds the form (i.e. all window items) from attribute settings included in the configuration file whereby the form contains pre-filled values of current block attributes. The configuration file also defines the type of each attribute input that can be considered as text field, select box, checkbox or radio button group.



Name: Integrator
External reset: none
Initial condition source: ● internal ○ external
Initial condition: 0
□ Limit Output
Upper saturation limit: inf
Lower saturation limit: -inf
□ Show saturation port
□ Show state port
Absolute tolerance: auto
□ Ignore limit and reset when linearizing
☑ Enable zero crossing detection
State name (e.g., 'position'): ''

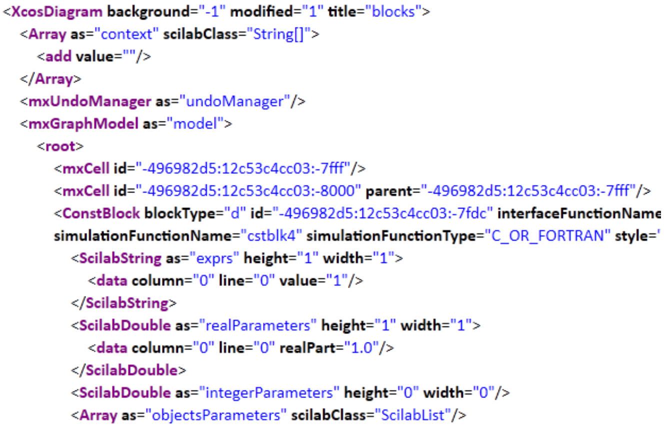
Fig. 2. Dialog window for changing block’s attributes

Each block has at least one connection point – input or/and output port. These are marked by different color when the user moves the mouse cursor over the block. To connect two blocks, two connection points have to be selected by the clicking on them. The connection is possible only between input and output of two different blocks or between input and some other connection line. In this case the new node is created at selected spot on existing connection automatically. The used connection points (input and output ports) appear as inactive and they are not able to be marked and included to a new connection.

If a connection consists of more than two line segments, a line block is inserted automatically at the center of the connection line. This pseudo block can be used to adjust position of the middle line segment of the connection.

4. BACK-END

The back-end server is used for running scripts that perform tasks such as import or export. These scripts work with Xcos files. In fact, the Xcos file has the same structure as standard XML file (see Fig. 3). This format makes handling Xcos files very simple.



```
<XcosDiagram background="-1" modified="1" title="blocks">
  <Array as="context" scilabClass="String[]">
    <add value=""/>
  </Array>
  <mxUndoManager as="undoManager"/>
  <mxGraphModel as="model">
    <root>
      <mxCell id="-496982d5:12c53c4cc03:-7fff"/>
      <mxCell id="-496982d5:12c53c4cc03:-8000" parent="-496982d5:12c53c4cc03:-7fff"/>
      <ConstBlock blockType="d" id="-496982d5:12c53c4cc03:-7fdc" interfaceFunctionName
      simulationFunctionName="cstblk4" simulationFunctionType="C_OR_FORTRAN" style='
        <ScilabString as="exprs" height="1" width="1">
          <data column="0" line="0" value="1"/>
        </ScilabString>
        <ScilabDouble as="realParameters" height="1" width="1">
          <data column="0" line="0" realPart="1.0"/>
        </ScilabDouble>
        <ScilabDouble as="integerParameters" height="0" width="0"/>
        <Array as="objectsParameters" scilabClass="ScilabList"/>
```

Fig. 3. Example of Xcos file structure (part of the file)

4.1 Export

The export process is done in four basic steps. Firstly, the scheme in web editor is sent to back-end server. The editor and back-end communicate via AJAX requests using JSON as data exchange format.

In the second phase, the back-end export script transforms the scheme into Xcos format. This step include generation of standard Xcos file header, listing of each block used in scheme together with relevant group of attributes, input and output ports. Block parameters are defined in configuration file (see Section 5) and actual values are taken from the exported scheme. When the blocks are processed, the connections between blocks are specified and the scheme is finalized with Xcos file footer

In the next step, the transformed scheme is saved into file and it’s offered to user to download afterwards.

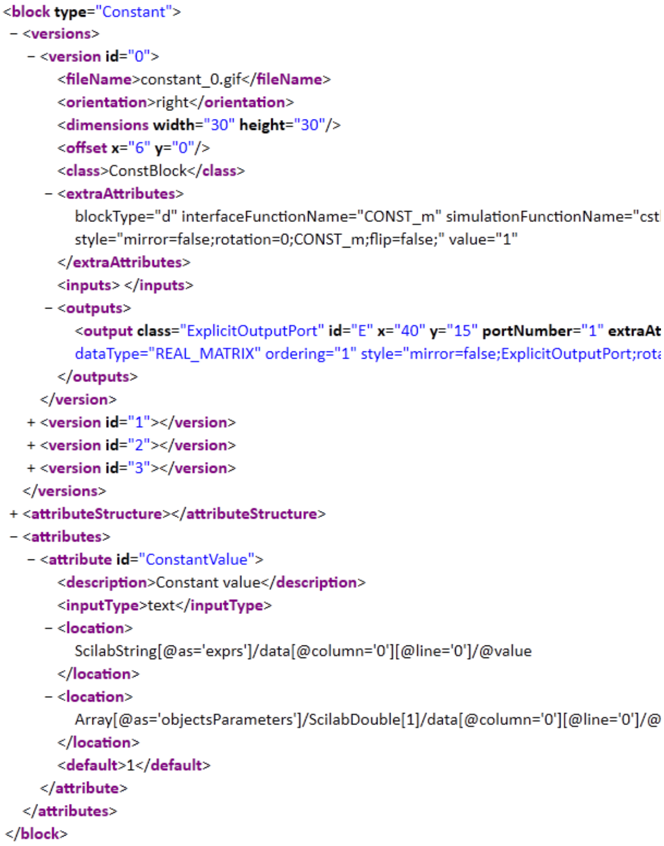
4.2 Import

The import process realizes the same steps as the export but in reverse order. First, user has to upload file containing scheme that is going to be imported. The import script parses the whole file and stores it as XML object.

The loaded content is searched for list of blocks and their attributes. In a case of blocks or attributes that are not present in web editor’s configuration file, the warning message is shown. These unknown blocks will not appear in the imported scheme. Processing of unknown attributes is skipped but the parent block will still appear in the scheme. Blocks are followed by a list of connections among blocks that are also being processed.

5. CONFIGURATION

The key part of application is the common configuration file used by front-end part of the application as well as by the back-end scripts for export and import. The configuration is stored in XML format. The XML file contains settings for each block that is available in the web editor. These settings include information about attributes that are used for displaying blocks in the editor: image file name, dimensions, block class name (e.g. *BasicBlock*, *ConstBlock* or other – used as XML tag that encapsulates block attributes in Xcos document), input and output ports.



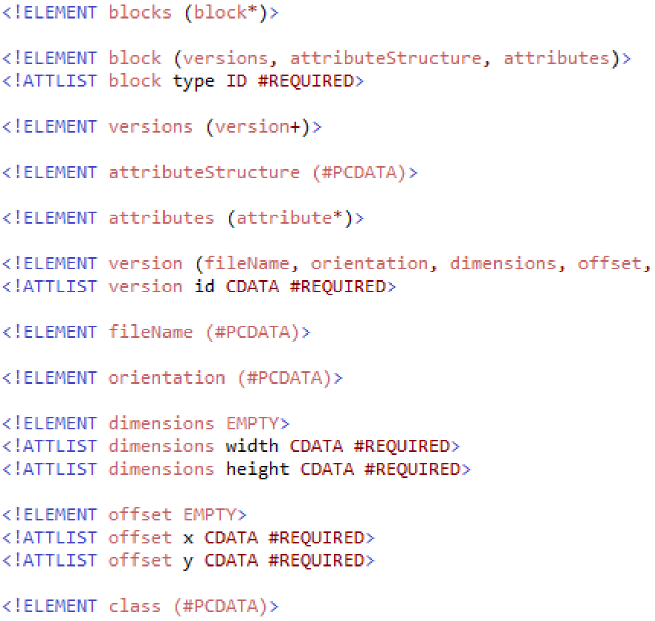
```
<block type="Constant">
 - <versions>
   - <version id="0">
       <fileName>constant_0.gif</fileName>
       <orientation>right</orientation>
       <dimensions width="30" height="30"/>
       <offset x="6" y="0"/>
       <class>ConstBlock</class>
     - <extraAttributes>
         blockType="d" interfaceFunctionName="CONST_m" simulationFunctionName="cst
         style="mirror=false;rotation=0;CONST_m;flip=false;" value="1"
       </extraAttributes>
       <inputs> </inputs>
     - <outputs>
         <output class="ExplicitOutputPort" id="E" x="40" y="15" portNumber="1" extraAt
         dataType="REAL_MATRIX" ordering="1" style="mirror=false;ExplicitOutputPort;rota
       </outputs>
     </version>
   + <version id="1"></version>
   + <version id="2"></version>
   + <version id="3"></version>
   </versions>
 + <attributeStructure></attributeStructure>
 - <attributes>
   - <attribute id="ConstantValue">
       <description>Constant value</description>
       <inputType>text</inputType>
     - <location>
         ScilabString[@as='exprs']/data[@column='0'][@line='0']/@value
       </location>
     - <location>
         Array[@as='objectsParameters']/ScilabDouble[1]/data[@column='0'][@line='0']/@
       </location>
       <default>1</default>
     </attribute>
   </attributes>
</block>
```

Fig. 4. Part of configuration file (Constant block)

Each block has a list of attributes (constant block in Fig. 4 has a single attribute – *ConstantValue*). The attribute definition contains its displayed name, input type (used in web editor's attributes window) and default value. Each attribute has its own specific location in Xcos file inside tags representing current block (e.g. *<ConstBlock>* ... *</ConstBlock>* as seen in Fig. 3). This location needs to be defined in configuration file as an XPath[4] address.

Data contained in the configuration file are used not only for visualization in web editor's interface, but also for correct function of import and export scripts.

The application is easily extendable by editing configuration file by the application administrator. It is not necessary to edit source codes of the application. New blocks can be easily added by modifying configuration XML file. Currently, the web editor includes the most common blocks used in Xcos block schemes. The set of included blocks is being extended simultaneously with further development of the web editor. Since the well-formed and error-free configuration file is very important for the correct function of the whole application, the DTD[5] document for configuration XML is also provided to prevent errors after modifications.



```
<!ELEMENT blocks (block*)>

<!ELEMENT block (versions, attributeStructure, attributes)>
<!ATTLIST block type ID #REQUIRED>

<!ELEMENT versions (version+)>

<!ELEMENT attributeStructure (#PCDATA)>

<!ELEMENT attributes (attribute*)>

<!ELEMENT version (fileName, orientation, dimensions, offset,
<!ATTLIST version id CDATA #REQUIRED>

<!ELEMENT fileName (#PCDATA)>

<!ELEMENT orientation (#PCDATA)>

<!ELEMENT dimensions EMPTY>
<!ATTLIST dimensions width CDATA #REQUIRED>
<!ATTLIST dimensions height CDATA #REQUIRED>

<!ELEMENT offset EMPTY>
<!ATTLIST offset x CDATA #REQUIRED>
<!ATTLIST offset y CDATA #REQUIRED>

<!ELEMENT class (#PCDATA)>
```

Fig. 5. Part of DTD for configuration file

6. CONCLUSION

The introduced application enables to prepare Xcos block scheme in online environment that in next step can be used for control of dynamical systems in frame of virtual and remote laboratories. Since it is developed by wide spread technologies it can be combined with the most of solutions that enable to use SciLab via Internet. It is to say that in the same time we are also developing similar tool for Matlab/Simulink environment (Janík and Žáková, 2009).

[4] XPath is used to navigate through elements and attributes in an XML document.
[5] Document Type Definition – defines legal document structure (list and structure of elements and attributes)

ACKNOWLEDGMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09.

It was was also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Digiteo (1989-2011). SciLab website. http://www.scilab.org

Holzner, S. (2007). Mistrovství v AJAXu. Computer Press. Brno.

Huba, M., P. Bisták, M. Fikar, M. Kamenský (2006). Blended Learning Course "Constrained PID Control". *7th IFAC Symposium on Advances in Control Education ACE'06*. Madrid, Spain.

Janík Z., K. Žáková (2009). Online Design of Matlab/Simulink Block Schemes. *10th International Conference Virtual University*. Bratislava.

Liguš, J., I. Zolotová, P. Karch, J. Ligušová (2010). Information and control system of traverse and its integration into cybernetic centre. Electronics and electrical engineering. No. 6, p. 147-152. ISSN 1392-1215

The PHP Group (2001-2010). PHP Documentation. http://www.php.net/docs.php

Refsnes Data (1999-2010). JavaScript and HTML DOM Reference. http://www.w3schools.com/jsref/default.asp

Resig, J., the jQuery Team (2010). jQuery Documentation. http://docs.jquery.com

Restivo, M.T., J. Mendes, A.M. Lopes, C.M. Silva, F. Chouzal (2009). A Remote Lab in Engineering Measurement, IEEE Trans. on Industrial Electronics, vol. 56, nº12, pp 4436-4843.

Schauer, F., M. Ožvoldová, F. Lustig (2008). Real Remote Physics Experiments across Internet – Inherent Part of Integrated E-Learning. In: *Int. Journal of Online Engineering (iJOE),* 4, No 2.

Schmid, Chr. (2003). Internet - basiertes Lernen, Automatisierungstechnik, 51, No. 11, p. 485-493.

Šimunek, M., P., Bisták, M. Huba (2005). Virtual Laboratory for Control of Real Systems, *Conference Proceedings ICETA*. Košice.

Žáková, K., M. Sedlák (2006). Remote Control of Experiments via Matlab. In: *Int. Journal of Online Engineering (iJOE),* 2, No. 3.

Remote control software for thermo-optical plant

M. Kalúz [*], **Ľ. Čirka** [*], **M. Fikar** [*]

[*] *Slovak University of Technology in Bratislava*
Faculty of Chemical and Food technology
e-mail: {martin.kaluz, lubos.cirka, miroslav.fikar}@stuba.sk
url: http://www.kirp.chtf.stuba.sk

Abstract: This paper describes development of a remote control laboratory. The main aim of this work is to create computer software for remote access and control of thermo-optical device uDAQ28/LT. We describe solutions that have been chosen for meeting the requirements for a fully user-friendly and an easy to use application. We choose the software programming platform Adobe Flash for client side application development and we develop a solution based on technologies PHP, MySQL, and MATLAB for server side.

Keywords: Remote Laboratory, Process Control

1. INTRODUCTION

Today, in a field of automation and process control education, it is necessary to create quality conditions, also by enabling students to use real experimental devices, on which they can carry out their experiments and improving their skills. This aim can be fulfilled by several different ways.

One of the most used solution is real technological laboratory. However, from the students view, it can be used only during school opening hours and it depends on accessibility of teachers. But there is another way how to provide students access to real experiments. It is remote laboratory that can be used for control of real technological plants from any location with access to Internet.

There are some solutions of remote laboratories over the Internet, created by software developers from another universities.

Interesting solution comes from FEI STU in Bratislava, where remote control software for the same thermo-optical device uDAQ28/LT has been developed. This solution is based on Java Applet and Java Servlet communication, and MATLAB used as main control software. Used technology is described in detail in Bisták and Beránek (2006). This solution was later extended by new reservation system (Bisták and Čirka, 2009) and has been applied to another type of device (hydraulic plant) (Žilka et al., 2008).

A DSP(Digital Signal Processor)-based remote control laboratory was proposed by Hercog et al. (2007). Their solution is based on in-house developed control hardware with MATLAB as control software and client application created in LabVIEW graphical programming environment. It is used for control of some real systems like electric motor, robot mechanism, power converter, etc.

Another remote control laboratory was created at Department of Industrial Systems Engineering of University Miguel Hernández in Elche, Alicante, Spain. Puerto et al. (2010) suggested remote control of two mechanical devices, a DC motor and sliding cylinder. The client side of software is built on HTML/PHP technologies. The communication part of software is based on Common Gateway Interface (CGI) and the main control software is MATLAB.

We develop a remote laboratory that can be accessed by Internet, using a web browser, but propose a different combination of technologies than above mentioned. Our remote control application is created as Adobe Flash program, framed in regular web page, accessible through e-learning system Moodle (Dougiamas, 2005). It also provides a video stream from remote web camera that is aimed at controlled device, so the remote user can observe the behavior of the experiment in real time. The Flash application is used on the client side. The server side with the real controlled device uDAQ28/LT consists of PHP/MySQL Apache-type server containing programs serving communication between every parts of our software solution, MATLAB that directly controls remote device, and Java database software drivers for providing access to MySQL from MATLAB.

In comparsion with client applications from Bisták and Beránek (2006), and Puerto et al. (2010), based on HTML/PHP and Java Applet, the Adobe Flash programming platform is more suitable for graphical design development and also it provides more options for graphical user interface (GUI) customization. From the simplicity point of view and software/hardware requirements, Java and HTML/PHP solutions seems to be better, but not so far. Hercog et al. (2007) used for building GUI the LabVIEW environment, which is directly designed for such use, but it is questionable if LabVIEW is an appropriate environment for building features like interface for communication between users, dynamically generated forms and fully animable GUI. The main advantages of our solution is the possibility of direct storage and data managment from experiments. This advantage stems from the using MySQL

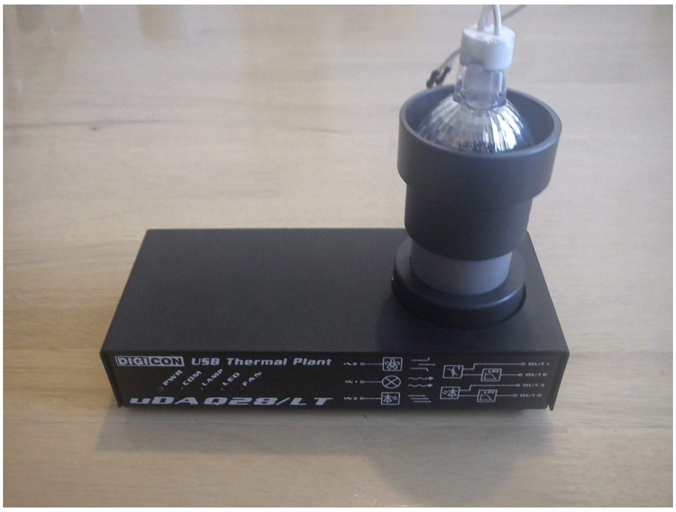


Fig. 1. Thermo-optical device uDAQ28/LT

database system, also as communication channel. Another difference is that we use technology based on asynchronous communication, so we can prevent the crash of experiment due to short-time losses of connection between client and server. MySQL, PHP layer, and MATLAB are located on the same physical server, therefore in the case of link outage between client PC and server, the local communication channels on server side stay unaffected by this issue, so the measuremnet remains running and data are collected in MySQL. In this case, user does not lose the measured data. The experiment is turned off only in case of longer-time link outage. Using PHP is fast and efficient way to secure communication between Flash application on client side and MySQL.

This is not our first attempt of using Adobe Flash platform in software development. Recently, we created an virtual laboratory based on Flash (Čirka et al., 2010), and MATLAB computation online laboratory based on Java (Kalúz et al., 2010).

2. UDAQ28/LT DEVICE

For purposes of remote laboratory and application testing, we chose a thermo-optical device uDAQ28/LT (Fig. 1) created by DigiCon Corporation (Huba et al., 2006). It is originally designed for education in fields of automation and process control. Device can be connected directly to computer by universal serial bus (USB) and after simple driver installation it can communicate with MATLAB. Manufacturer provides with device the installation CD including software drivers, Simulink schemes, and user manual. Plant can be controlled by 3 analog inputs (bulb voltage, fan voltage, and LED voltage), and has 4 primary outputs (light intensity, temperature inside light tube, fan rpm, and voltage taken by fan).

3. SOFTWARE SOLUTIONS

Our software solution combines several technologies, which were chosen depending on selected criteria. The main criteria were data storing ability, connection robustness, program processing rate, easy-to-use graphical user interface, and low system and hardware requirements.

In this section, we will explain the realization of links between every part of chosen technology and its principles.

Every user instruction from Flash application is processed on server side by PHP scripts.

PHP is used in three different ways.

(1) The first is a group of scripts, which serves application timing and simple computations with results directly sent back to application.
(2) The second group of PHP scripts is used to serving connection between Flash application and MySQL database system.
(3) The third group of scripts is used for executing of server system commands.

We use MySQL system for storing data like device states, running experiment states, MATLAB states, connection states, user accounts, and other. The uDAQ28/LT device on server-side is directly controlled by MATLAB/Simulink. We have created a Simulink block function, which sends the measured data to database in every sample time period of running experiment. These data are collected by Flash application and sent to user. The connection between MATLAB and MySQL is performed by Java database connection driver (JDBC).

The communication between every part is shown in Fig. 2. Technology of experiment execution and data collecting is as follows. MATLAB is executed by PHP script that sends execution command directly to operating system command line. This command is built in client-side application and consist of two parts. The first is system command for MATLAB execution with program setup parameters, and the second is command for MATLAB command line. This second part contains all instructions for experiment execution, as names of Simulink schemes and m-files to run, input parameters, and other instructions for MATLAB. When MATLAB/Simulink program is running, the specific m-file script writes to MySQL information about its state (1 – MATLAB is running). Another state is collected from Simulink. These states are periodically checked by client application, and if both are set to 1, it means that experiment is properly running, and client-side application switches to screen for data observation. Data collecting is based on asynchronous technology. Every new measured state from device is sent by Simulink to MySQL and periodically collected by client application. Sampling period for Simulink experiment is set up by user and for data collection by Flash application is set to 0.2 seconds. MATLAB uses JDBC for database connection and Flash application connects to database through PHP layer. After experiment is finished, MATLAB is automatically turned off by its script. It can be turned off also manually by the user from client application.

4. APPLICATION OVERVIEW

The remote laboratory can be used by anyone who is registered as user of Moodle. After sign in to remote laboratory course, user fills a simple registration form to create a new account. After registration, user receives a confirmation e-mail with account activation link and can sign in to remote control application.

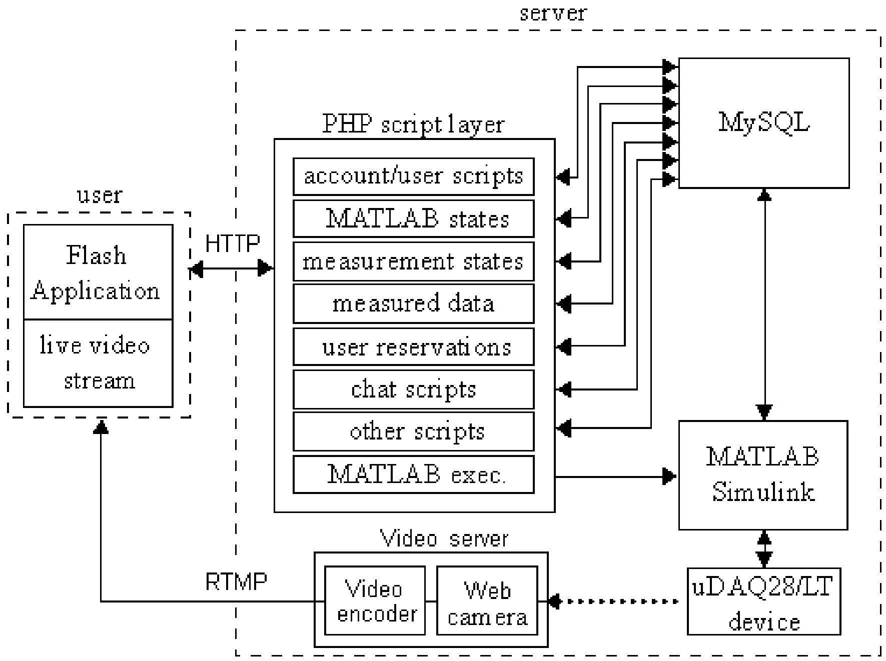


Fig. 2. Scheme showing connections between client and server side of software solution, and controlled device

The Flash application is accessible through any web browser, because it is embedded into regular web page. The only requirement to run the application is Adobe Flash Player (downloadable software plug-in). The application web site is located on e-learning system LMS Moodle, and it is accessible through the main web site of our department.

Application GUI consists of several screens. The first is login interface (Fig. 3), where user has to put his login name and password to sign up for experiment.

In the actual version of our solution, it is necessary to use accout login option directly in client application, because the actual version of application is aimed to be independent on web location. Due to fact that our solution is a part of remote laboratory course, located on Moodle, in future we plan to create extension that will provide access to application user accout directly through Moodle, by using its session cookies.

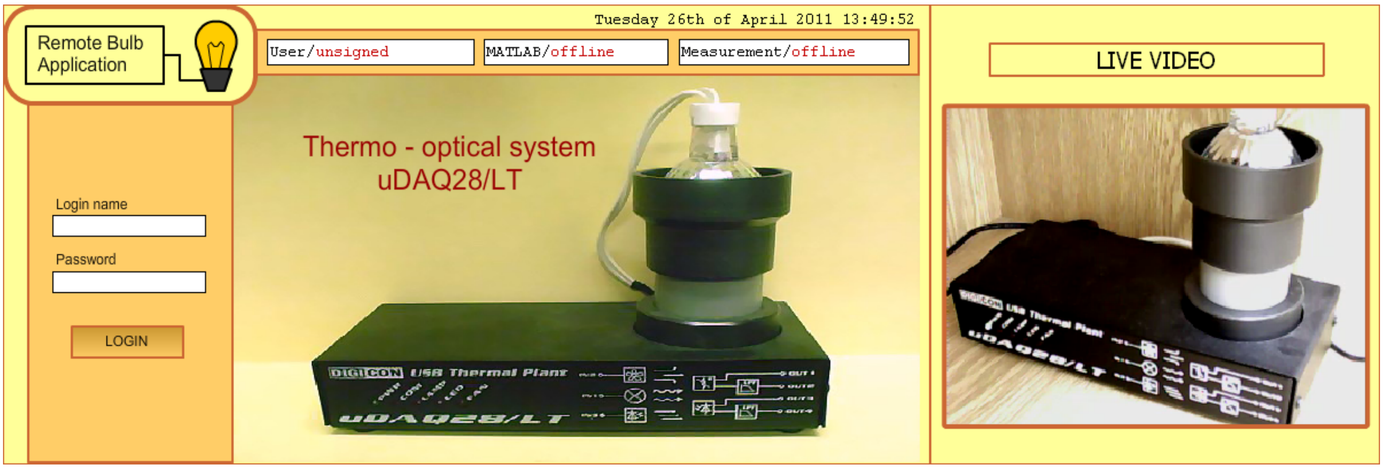
On the second screen (Fig. 4) user can reserve the time session for experiment and communicate with other signed up users by chat. Application provides option for reservations management, like creation, removal, and sort.

Before proceeding to the experiment itself a new time reservation has to be created. Flash program scripts run in background of application to compare new reservation with those which are saved in database to avoid time collision between experiments. One user can register maximum three reservations with maximal duration of 30 minutes (sufficient time for performing the experiment on uDAQ28/LT).

If session reservation is finished, user can switch to another screen, where he can set up the experiment. On setup screen (Fig. 5) user chooses Simulink scheme which will be executed in remote MATLAB. All Simulink schemes are located at server and registered in MySQL database system. Content of each scheme is automatically detected by parsing software. This software part is necessary for gathering informations about Simulink model files (MDL-files) and their content. In application, when user selects scheme for measurement, the parsing PHP script reads the MDL file and detect all important objects in it. The script looks for Simulink blocks and their parameters, and sends information about them back to Flash application, where the input form is dynamically generated. This way, user can view and edit all important parameters that model requires. The script detects most commonly used blocks and their parameters, like constant, step, transfer function, and PID controller blocks. User can change their values before measurement starts and also select parameters that may be changed during experiment.

Application contains all default simulation schemes provided by device manufacturer, but there is also an option to upload custom scheme created by user. For this purpose we have created upload form that provides all necessary features important for keeping software security and operationality. Before a new schema is uploaded to server and saved to database, the security PHP script checks if file has proper extension, size, and if contains all necessary parts, like database connector block and block for data export. In last step PHP script reads file to look for dangerous content (for example system commands writen by user).

When all input parameters for experiment are correctly filled, user can execute the experiment. After connection between user-side application and MATLAB is established and MATLAB/Simulik is properly running, the application switches to experiment screen (Figs. 6 and 8). User can observe experiment results and remote device states and also can send commands for setting up new values of device inputs (Fig. 7).



Remote Bulb Application

Tuesday 26th of April 2011 13:49:52

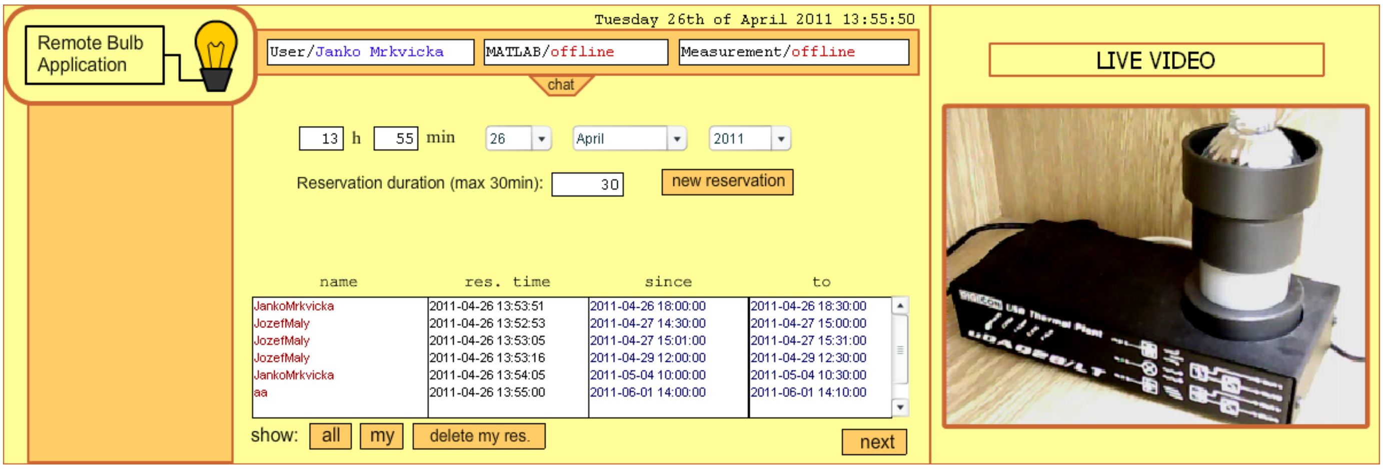
User/unsigned | MATLAB/offline | Measurement/offline

Thermo - optical system uDAQ28/LT

Login name

Password

Fig. 3. First screen of Flash application, showing the picture of remote device and login form



Remote Bulb Application

Tuesday 26th of April 2011 13:55:50

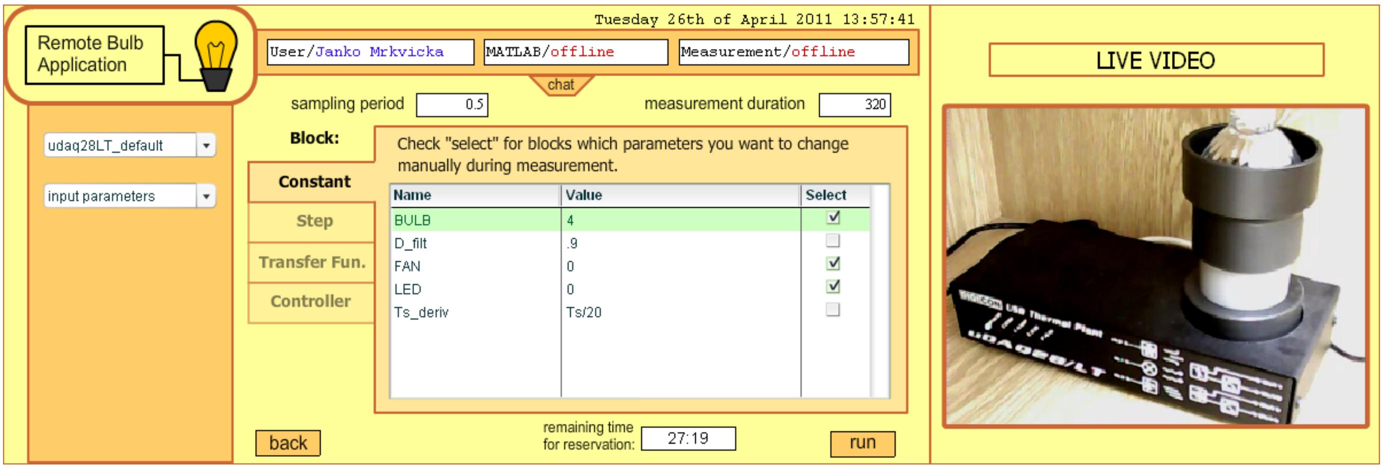
User/Janko Mrkvicka | MATLAB/offline | Measurement/offline

13 h 55 min 26 April 2011

Reservation duration (max 30min): 30

name	res. time	since	to
JankoMrkvicka	2011-04-26 13:53:51	2011-04-26 18:00:00	2011-04-26 18:30:00
JozefMaly	2011-04-26 13:52:53	2011-04-27 14:30:00	2011-04-27 15:00:00
JozefMaly	2011-04-26 13:53:05	2011-04-27 15:01:00	2011-04-27 15:31:00
JozefMaly	2011-04-26 13:53:16	2011-04-29 12:00:00	2011-04-29 12:30:00
JankoMrkvicka	2011-04-26 13:54:05	2011-05-04 10:00:00	2011-05-04 10:30:00
aa	2011-04-26 13:55:00	2011-06-01 14:00:00	2011-06-01 14:10:00

Fig. 4. Second screen of Flash application, showing reservation system for experiment



Remote Bulb Application

Tuesday 26th of April 2011 13:57:41

User/Janko Mrkvicka | MATLAB/offline | Measurement/offline

sampling period 0.5 measurement duration 320

udaq28LT_default

input parameters

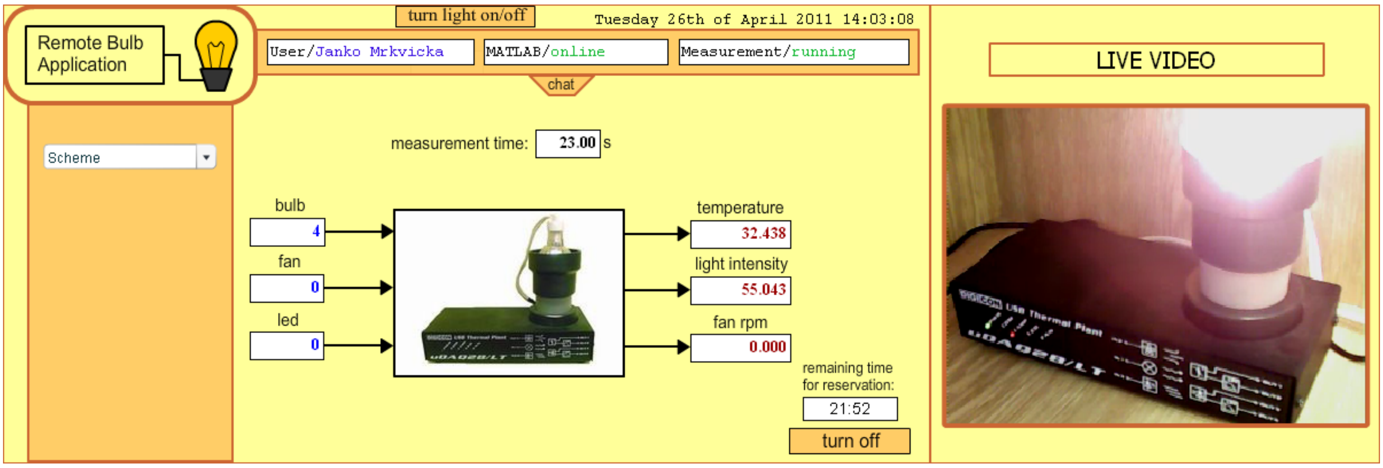
Block:

Check "select" for blocks which parameters you want to change manually during measurement.

Name	Value	Select
BULB	4	☑
D_filt	.9	□
FAN	0	☑
LED	0	☑
Ts_deriv	Ts/20	□

remaining time for reservation: 27:19

Fig. 5. Input form for experiment parameters associated to chosen scheme



Tuesday 26th of April 2011 14:03:08

User/Janko Mrkvicka

MATLAB/online

Measurement/running

measurement time: 23.00 s

bulb: 4

fan: 0

led: 0

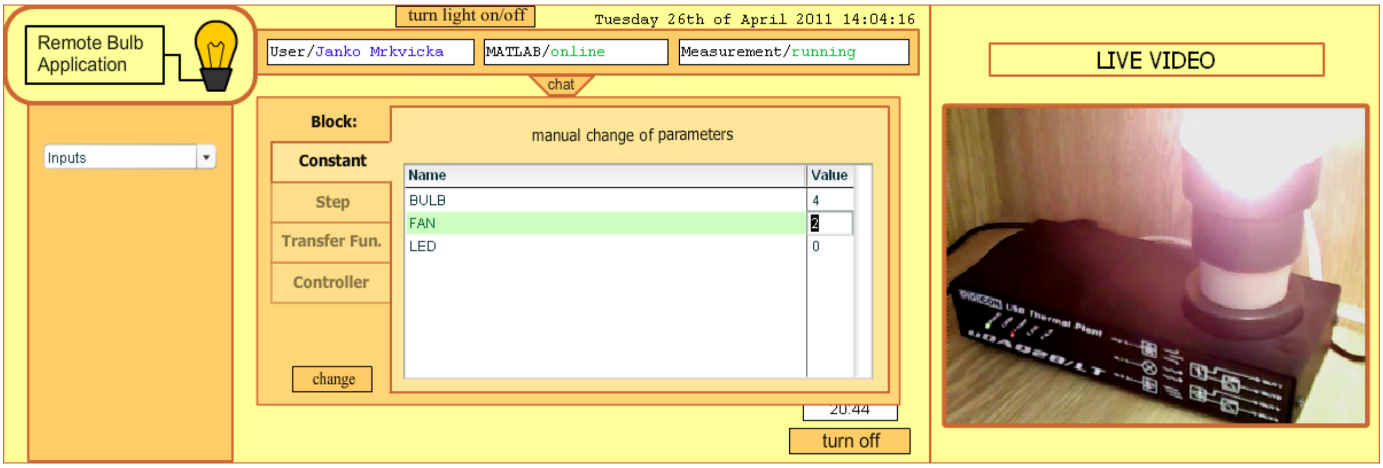
temperature: 32.438

light intensity: 55.043

fan rpm: 0.000

remaining time for reservation: 21:52

Fig. 6. Running experiment (input/output view)



Tuesday 26th of April 2011 14:04:16

User/Janko Mrkvicka

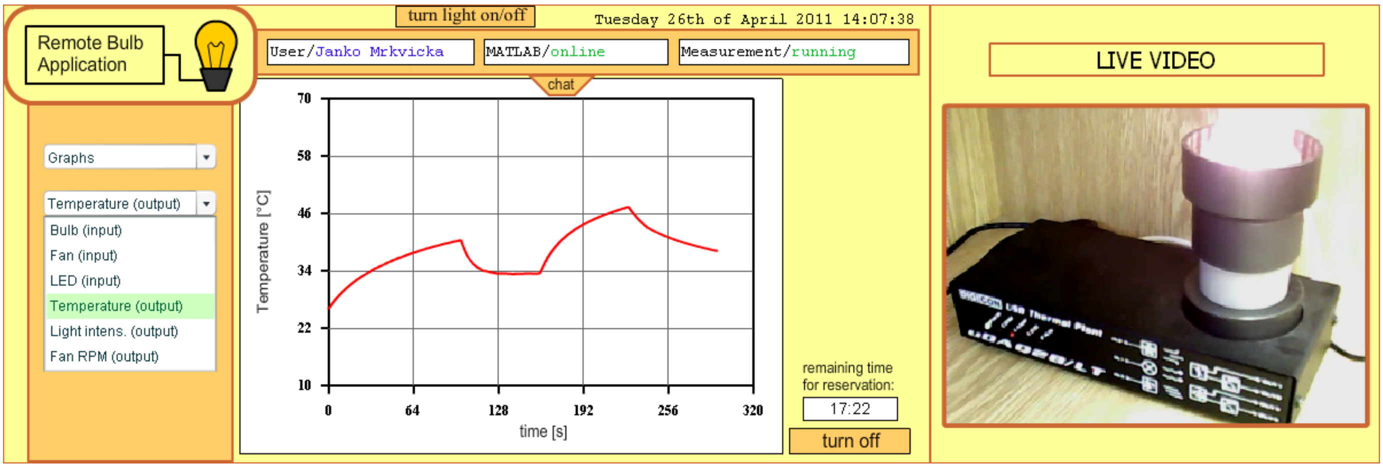
MATLAB/online

Measurement/running

manual change of parameters

Name	Value
BULB	4
FAN	2
LED	0

Fig. 7. Running experiment (input form where user can change parameters in real time)



Tuesday 26th of April 2011 14:07:38

User/Janko Mrkvicka

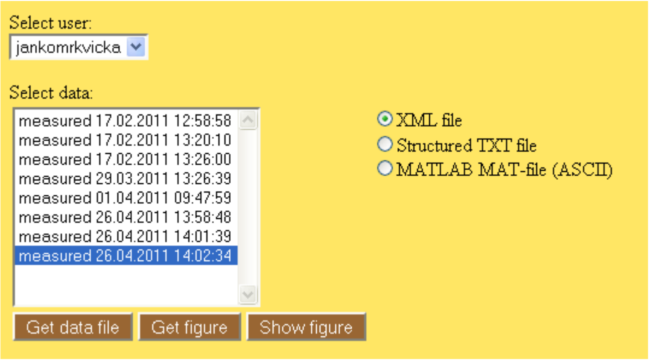
MATLAB/online

Measurement/running

remaining time for reservation: 17:22

Fig. 8. Running experiment (selected variable graph view)

When experiment is finished, user can download all measured data from database in chosen format. Application can provide data in the form of structured XML file, plain text file or MATLAB m-file. Every experiment result is stored in database with unique identificator, and can be accessed later. Web module for data export is shown in Fig. 9.



Select user:
jankomrkvicka

Select data:
measured 17.02.2011 12:58:58
measured 17.02.2011 13:20:10
measured 17.02.2011 13:26:00
measured 29.03.2011 13:26:39
measured 01.04.2011 09:47:59
measured 26.04.2011 13:58:48
measured 26.04.2011 14:01:39
measured 26.04.2011 14:02:34

● XML file
○ Structured TXT file
○ MATLAB MAT-file (ASCII)

Fig. 9. Data export form. Results are sorted by user name and time of measurement

5. CONCLUSION

Remote laboratories can be a suitable way to improve education in field of automation engineering and process control. Our proposed solution presents one of many different ways how can remote laboratory be realized. We have chosen technologies which can easily handle features that are required for this kind of solution. MySQL database system is suitable for fast storage and data management and it is also robust enough for programs with high database access traffic. Adobe Flash Application on user side provides easy-to-use GUI and can be run directly through web browser without any installation procedure.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency of the Slovak Republic under the grant 1/0071/09.

The paper is supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Bisták, P. and Beránek, M. (2006). Java client server applications for remote laboratory based on matlab and com. In *7th Int. Conference Virtual University VU06*, 135–139. Bratislava, Slovakia.

Bisták, P. and Čirka, Ľ. (2009). Reservation of remote laboratory using Moodle LMS. In M. Huba (ed.), *Proceedings of 10th International Conference Virtual University*. FEI STU in Bratislava, E-academia Slovaca, Bratislava.

Čirka, Ľ., Kalúz, M., Kvasnica, M., and Fikar, M. (2010). Virtual laboratory. In *Proceedings of the 9th International Scientific–Technical Conference Process Control 2010*, C029a–1–C029a–8. University of Pardubice, Kouty nad Desnou, Czech Republic.

Dougiamas, M. (2005). Moodle – a free, open source course management system for online learning. `http://moodle.org`.

Hercog, D., Gergič, B., Uran, S., and Jezernik, K. (2007). A DSP-based remote control laboratory. *IEEE Transactions on Industrial Electronics*, 54, 3057–3068.

Huba, M., Kurčík, P., and Kamenský, M. (2006). *Thermo-optical device uDAQ28/LT*. STU Bratislava, Illkovičova 3, Bratislava.

Kalúz, M., Čirka, Ľ., and Fikar, M. (2010). MATLAB builder JA in control engineering education at FCFT STU. In *Technical Computing Bratislava 2010*, volume 18, 1–5.

Puerto, R., Jiménez, L.M., and Reinoso., O. (2010). Remote control laboratory via internet using matlab and simulink. In *Computer Applications in Engineering Education*, volume 18, 694–702.

Žilka, V., Bisták, P., and Kurčík, P. (2008). Hydraulic plant remote laboratory. *International Journal of Online Engineering*, 4, 69–73.

Comparison of Supervisory and Networked Control in Remote Laboratories

P. Folvarčík

*Slovak University of Technology,
Faculty of Electrical Engineering and Information Technology,
Institute of Control and Industrial Informatics,
Ilkovičova 3, 812 19 Bratislava, Slovak Republic
(e-mail: pavol.folvarcik@stuba.sk)*

Abstract: This paper presents some problems of remote control of real systems. Firstly, it compares the quality of local control (controller are located on the server PC) and remote control (controller are located on client PC) for systems with small time constant. Consequently it deals with solving the problems in the remote control. Solution is realized by modification of communication between client and server and reduction of the quantity of transferred and processed data. After that, communication will be faster and the application will be usable for systems with a lower time constant.

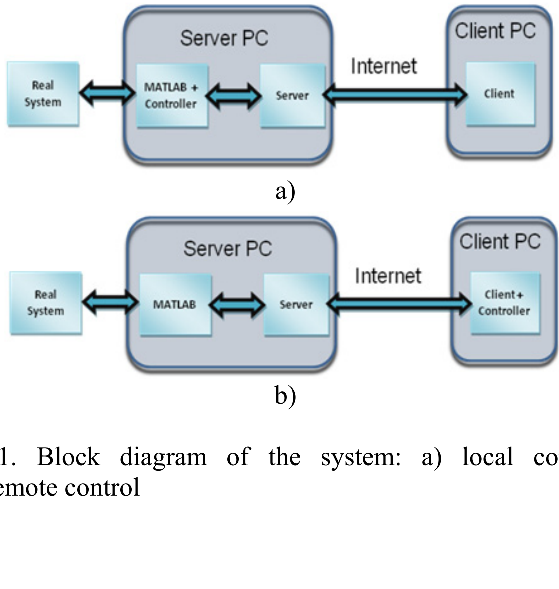
Keywords: remote control, MATLAB, client-server, time delay

1.INTRODUCTION

The existence of transport delays is a normal feature of many technological processes in the input-output relations. Production devices with a time delay often can't be controlled using standard controller designed without considering the presence of transport delays. Action value generated by controller faces to destabilize the feedback loop. This paper is dedicated to analyze the influence on the quality of the control system and eliminate their effects.

2. TRANSPORT DELAY OF SYSTEM

Our control system consists of several parts, between which delays of data can occur and it can cause reduction of control quality. The whole system is composed of client-server application, the computer with running Matlab and the real system, which is connected to a computer.



a)

b)

Fig.1. Block diagram of the system: a) local control; b) remote control

In the early stages of my work was control scheme created in Matlab, that was controlled the system (Fig. 1.a). The task of server has to been transfer the necessary data between client and Matlab, especially the parameters necessary to run the simulation. Consequently it sent the measured data from the system to the client that can visualize it for the users. This solution of telematics control was restrictive for the user, because they could not design their own controller. More appropriate solution is to move the controller from the scheme in Matlab directly into the client application, where user can freely modify it (Fig. 1.b). This solution has brought to the feedback loop a transfer delay that was before minimal. It is only an academic example and it can never be used in practice, because it does involve significant security risk and eliminates the possibility to achieve quality closed loop behavior. It was developed only to show an impact of transport delays on quality of control process for students. In the final version of the application the user can choose between server-side control (in Matlab scheme) and the client-side control (its own controller).

2.1Transport delay between client and server

Seeing that client and server run in simulations on the same computer, network delay is minimal. Network delay will increase when running a client application on another computer on the network. Its duration depends on your connection speed and also the network traffic. When the network is without traffic, the network delay is around several tens of millisecond. When the network is with traffic, the value of transport delay is increasing on the value of hundreds of milliseconds, which can lead to the instability of the system for systems with small time constant. Seeing that client can connect to the server from anywhere on the Internet, it is necessary to consider which type of control in simulation will be better.

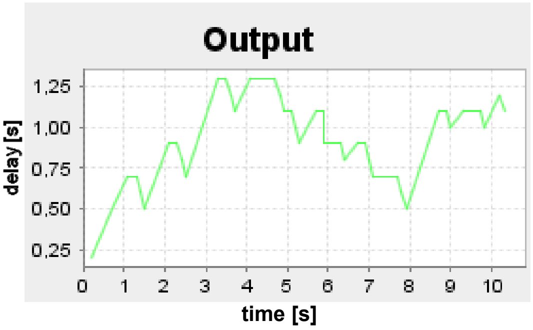
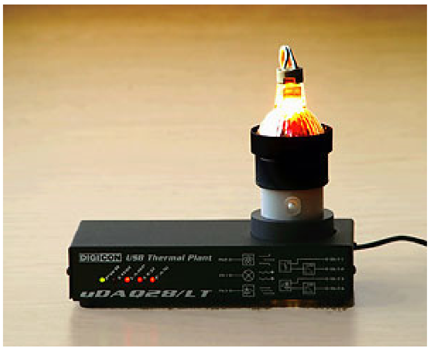


Fig.2. Transport delay between client and server

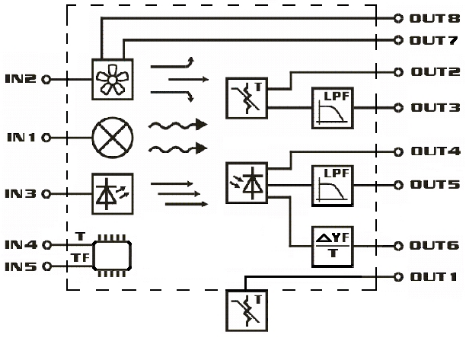
In Fig. 2 you can see evolving of time delay between running the simulation and the time at which the sample was created. The values are in second. Time was measured by creating timer and run it on the client side at a time when it was confirmed to launch simulation on the server side. Their difference was the transfer delay with which the sample was received from the server.

These delays are total, and it includes itself all parts of the delays which occurring during the simulation. Highest part of delay takes the delay of the communication between server and Matlab. Seeing that client and server run in simulations on the same computer, network delay is minimal. Network delay will increase when running a client application on another computer on the network using a video stream from the camera. Delays in some points are more than 1 second. It is necessary to reduce this delay or to use it only to systems with large time constant.

In Fig. 4 is shown the delay if the network is loaded. To made a traffic, we used a video stream, that transmits data over the network at 6Mbps. The picture shows that delays in some samples adding up to less than 3 seconds. This delay leads to system instability. We will show the impact of this delays on the control quality on real thermo-optical plant uDAQ28/LT (Fig. 3) (Huba, 2008).



a)



b)

Fig.3. a) Thermo-optical plant uDAQ28/LT b) Basic electric diagram of thermo-optical plant uDAQ28/LT

This system was designed to support of education of process control. System has three manipulated inputs: bulb voltage (0-5V) which represents heater and light source, fan voltage (0-5V) which can be used for temperature decreasing and voltage of led diode (0-5V) which represents another source of light. On the output is possible to measure seven variables: temperature insight the system (direct or filtrated), outsight temperature, light intensity (direct or filtrated), fan velocity and fan current. In the next two figures (Fig. 5. and Fig. 6.) is a comparison of the output of optical channel with control on the server side (without transport delay) and control on the client side (with transport delay). The network was with traffic 6 Mbps.

After elimination of network traffic delay was reduce (Fig. 2.) and the quality of regulation was better (Fig. 7.). Values of transfer delays are stil quite high and the system we can not regulate.

Control process for this delay may not work properly because it is a system with time constant less than 2 second. In Fig. 8 and Fig. 9 are shown traces of the outputs of the system with time constant of more than 20 seconds. It's unfiltered thermal channel of thermal-optical plant.

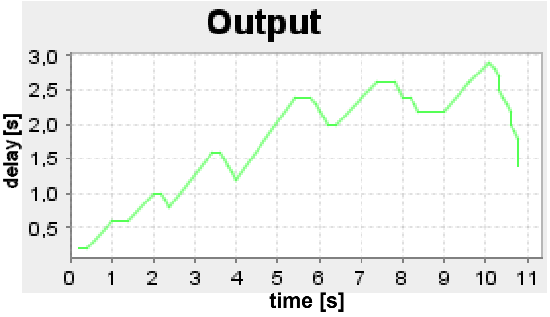


Fig.4. Transfer delay on the network with traffic

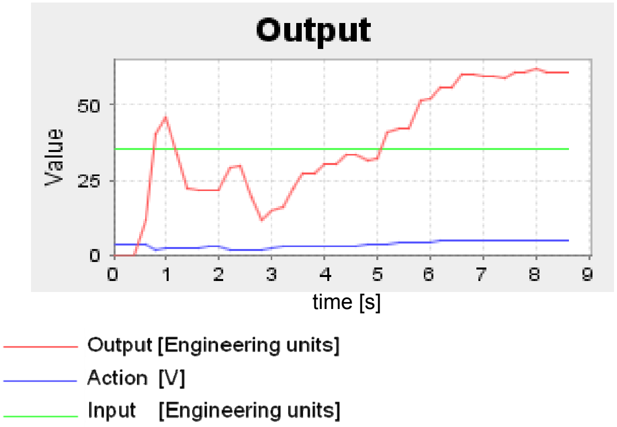


Fig.5. Simulation on real system with client control

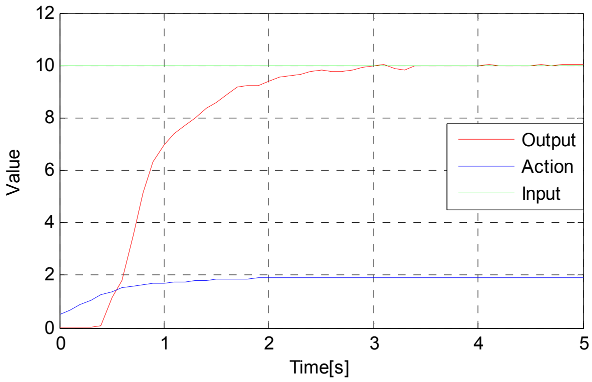


Fig.6. Simulation on real system with server control

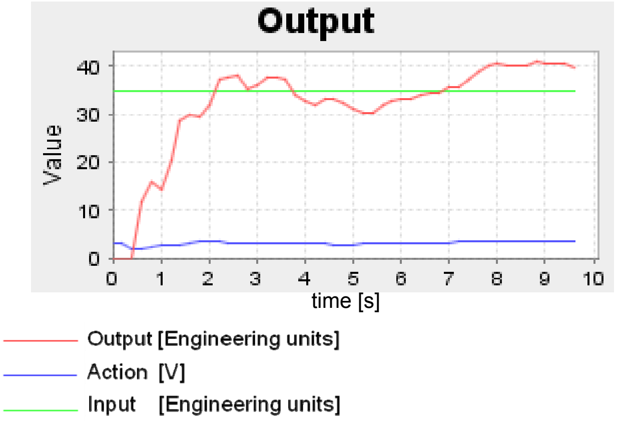


Fig. 7. Output value on network without trafic

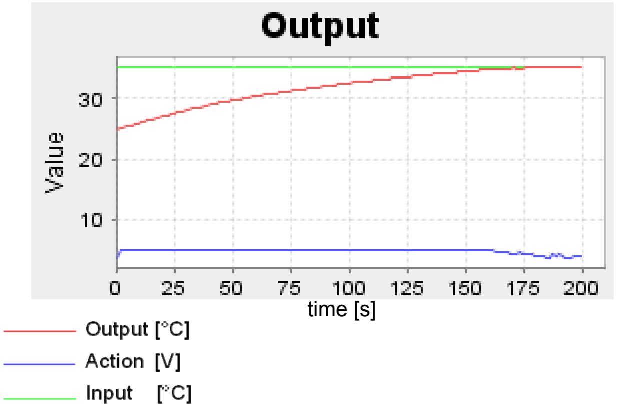


Fig. 8. Output value with server control

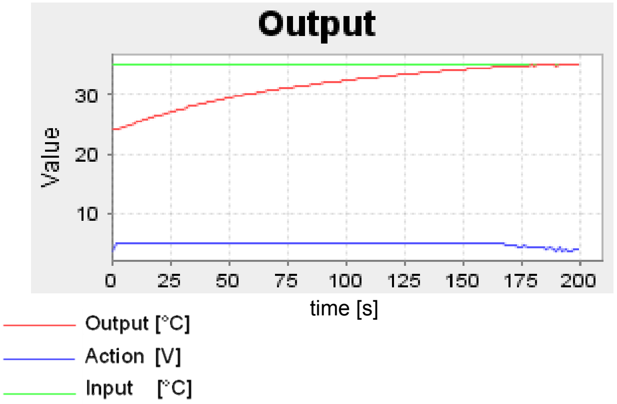


Fig. 9. Output value with client control

In the chart we can see that delays caused small overshoot on output values. The course of action values is different for both graphs, which is caused by transfer delay. As this is a system with high time constant, did not influence the course of output variables. Our application works reliable on systems with high time constant.

2.2 Transfer delay betweet Matlab and server

In communication between Matlab and the server it causes delays in transmission in both directions. Communication has been made by using COM objects (Beránek, 2006). Calculation of the action value is performed on the client side and therefore it is necessary send required data to the client to calculate. Server application receives by the method *getRealtimeData()* measured values of Matlab, that it sends to the client.

Data are obtained from the *scopeData* variable in that Matlab Scope object stores data during simulation. This is the only way to get the measured values during simulation. Its disadvantage, however, is that Matlab sends to the server all previously stored data. Data is sent as a matrix whose number of rows is equal to the number of samples and number of columns is equal to the number of sent variables. During the simulation it's changing the number of rows in the matrix. The server then must recognize what is already sent to the client and what is not his yet. For this task we have *upperBorder* and *lowerBorder* variables. In to the *lowerBorder* variable is written number of samples received in the previous matrix. In to the *upperBorder* variable is written number of samples in new matrix. Client receives only the samples with indexes between *lowerBorder* and *upperBorder*. This method of sending was designed for the application with server side control and it was sufficient for the user, because the delay, that it caused, has no effect on the quality of control. However, when we wanted used the same method to the client side control, we encountered a few problems.

The most significant was that the server sent at once 3 samples on average and client calculate an action variable for each of them in the order they were received. First it calculated an action value for the sample which was calculated at time t-2Ts, then t-Ts and finally for the sample at time t. In calculating the control value it calculated value from the sample, which was no longer current. The most accurate calculation was only at the last received sample. On the control process that had the most impact in systems with small time constants. The impacts of this delay, we have analyzed in section 2.1, where we showed that delay is relatively high also without the delays in transfer over the network. Delays in some places are more than 1 second. That is why it is necessary to reduce this delay or to use it only for systems with high time constant. Because we want a system with the widest possible use, we looked for a way to reduce this delay.

3.MINIMIZING THE TRANSFER DELAY

The biggest time delay was identified when server has sent more measured samples at the same time. Because we can not control the speed of taking samples of Matlab, the only possibility to speed up the transfer is sent only the most current sample to the client. This ensures that the client will not receive samples that are not current and it will not calculate an action value. At the same time we reduce the number of transferred data and communication will be faster. Previous server sent measured samples by the index value from *lowerBorder* to *upperBorder*. After modifying the servers code, server sends only the value with index *upperBorder*, consequently client receive only the most current value. Client calculates action value and sent it to the server. Server sent the value to Matlab and wait for next sample. In the equation for calculating the action value acts parameter Ts. It is sampling time. Client, but does not receive all of the the samples, but only some of them. In practice this means that is sending approximately every third sample, so the actual sampling time is approximately 3Ts.So that client calculate an action value always at the correct sampling time it must calculate sampling time at each step itself. Sampling time is thus variable. This function provides the following code:

```
multiple=(t[0]-t[1])/Ts;
if(multiple==0) multiple=3;
t[1]=t[0];
```

Action value is then calculates:

```
u[0]=u[1]+P*e[0]+multiple*Ts*I*e[1]-P*e[1];
```

`u[0]` – actual control action
`u[1]` – control action in time t-Ts
`e[0]` – actual control error
`e[1]` – control error in time t-Ts

After these modifications, we minimize the delays that arose in our application. In Fig.10 we can see time delay of the sample. It is seen that the delay is significantly reduced. Before the modifications it fluctuated between 0.5 and 1.25 seconds. After the modifications it fluctuated between 0.2 and 0.4 seconds.

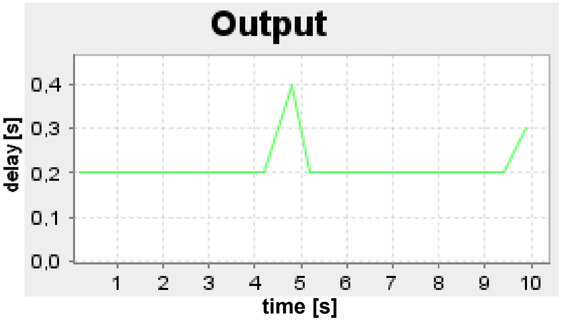


Fig. 10. Time delay of sample

This delay was measured without network load. After adding the traffic like in section 2.1, thus the video stream with speed of 6Mbps, we measured delay between 0.2 and 0.8 seconds (Fig. 11). In the previous server application it was between 1 and 3 seconds. In Fig. 12 and Fig. 13 is a comparison of control on the client side with control on the server side for unfiltered optical channel of thermal-optical system (system with small time constant).

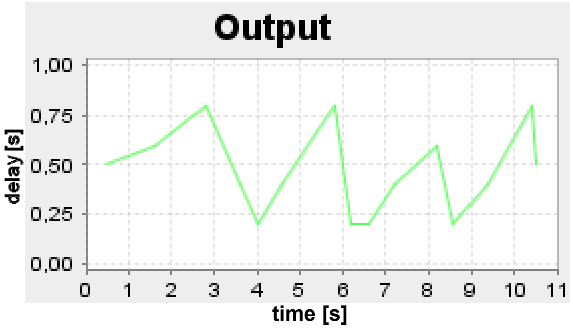


Fig. 11. Delay of sample on network with traffic

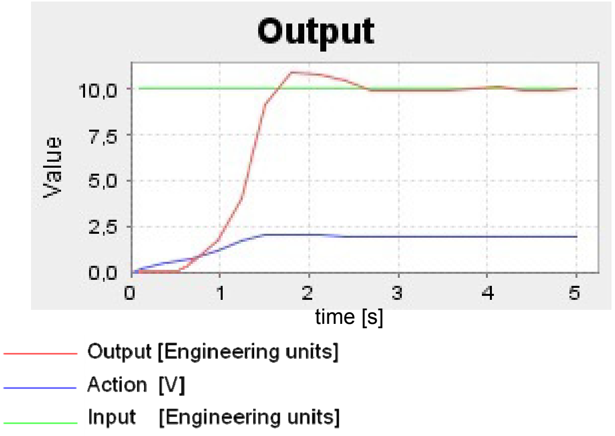


Fig. 12. Output value with client control

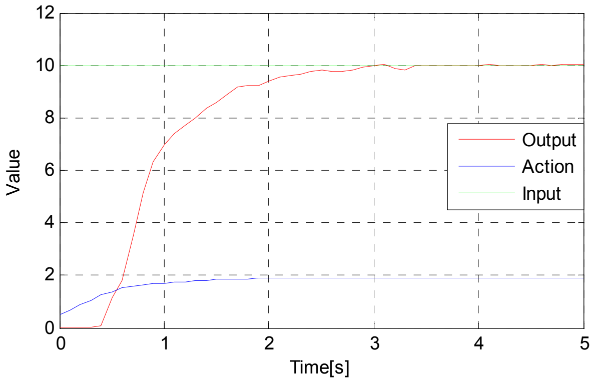


Fig. 13. Output values with server control

The chart shows that the delay of the sample has an impact on the quality of control. Transport delays caused overshoot in our case. Controller, despite delay, regulates the output value to the desired value. In the previous server application was system unstable. On the quality of the regulatory process had influence two basic components. First was the delay and the second was sampling time. On the server side we used Ts = 0.2 seconds. But n the client side not. Although Matlab counted with sample time Ts = 0.2 seconds, but the client receives every second or third sample. Thus, sampling time was at the client two to three times greater. Thus, if we want to compare most accurate, we must set the sampling time for the client minimal to the value Ts / 2. Fig. 14 shows the course of output variables with Ts = 0.1 seconds.

The output is similar to the output of server-side control. This showed that the sampling time in this system had a greater impact to quality of control them the transfer delay. This simulation showed that the server side control is working correctly.

Server and client are modified so that we can send to the server information, which type of control we want to use (client side or server side). The user can choose this parameter before running simulations directly in the window of client application. Hereby he can set parameters of PI controller that is used in simulation with client-side control.

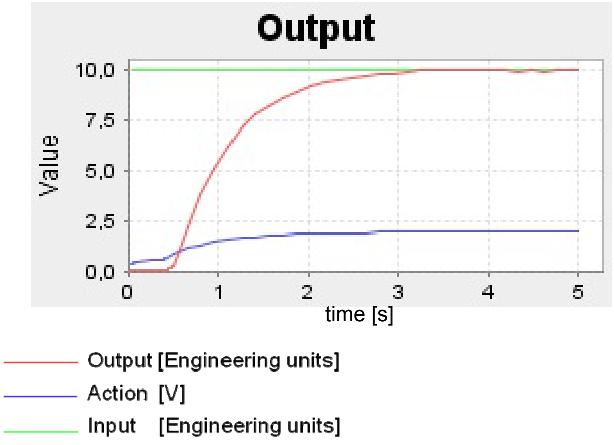


Fig.14. Output of system with Ts= 0.1 second

Each of these options uses a different control scheme, it is therefore necessary to choose the correct startup scheme. Otherwise the simulation will not run and Matlab sends an error message.

4. CONCLUSION

Simulations proved that an application for remote control works correctly also for systems with small time constant (approximately 2 second). The theory, about instability of feedback loop with high transfer delays, was confirmed. The quality of control process was much better after the identification of transfer delays and minimization of them. This solution will enable greater use of application, especially in educational process. It also allows users to create their own controller and simulate its behavior on different real systems. Subject of next research would be to eliminate any delays in communication on the server side.

ACKNOWLEDGMENTS

The work has been partially supported by the Grant KEGA No. 3/7245/09, by the Grant VEGA No. 1/0656/09 and by the Grant VEGA No. 1/0369/10. It was also supported by a grant (No. NIL-I-007-d) from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is also co-financed from the state budget of the Slovak Republic.

REFERENCES

Beránek, M. (2006). Client-server application for remote control of real. *Diploma thesis*. STU Bratislava.

Bisták, P. (2008) Remote Laboratory Java Server for Data Exchange with Matlab Automation Server, *Proceedings of International Conference REV 2008*, Düsseldorf, Germany.

Bistá, P. (2009). Matlab and Java based virtual and remote laboratories for control engineering. In: *17th Mediterranean Conference on Control & Automation*.

Gomes L., S. Bogosyan (2009). Current Trends in Remote Laboratories, *IEEE Trans. Industrial Electronics*, vol. 56, No. 12, pp. 4744- 4756.

Huba, M. (2008) Thermo-Optical Laboratory Plant uDAQ28/LT, Technical and Users Guide, http://www.eas.sk/members/2/file/070212%20opticko_te pelna_sustava.pdf

Müller S., H. Waller (1999). Efficient integration of real-time hardware and Web based services into MATLAB, *Proceedings of 11th European Simulation Symposium*, Erlangen, Germany.

Restivo, M. T. et al. (2009). A Remote Laboratory in Engineering Measurement. In: *IEEE transactions on industrial electronics*, Vol. 56, No.12 [online].

Safaric, R., S. Uran, M. Trunic, I. Hedrih (2004). Remote controlled mechatronics device via internet using Matlab, *Proceedings of 1st Int. REV Symposium*, Villach, Austria.

Sheng Y., W. Wang, J. Wang, J. Chen (2008). A Virtual Laboratory Platform Based on Integration of Java and Matlab, *Li, F. et al (Eds.) ICWL 2008*, LNCS, vol. 5145, Springer-Verlag Berlin Heidelberg.

ABS/TCS Simulator

M. Juhás *, P. Seman **, S. Bodi ***

** Institute of automation, measurement and applied informatics*
Faculty of Mechanical Engineering
Slovak University of Technology in Bratislava
(Tel: +421 917 825280; e-mail: martin.juhas@stuba.sk)
*** Institute of automation, measurement and applied informatics*
Faculty of Mechanical Engineering
Slovak University of Technology in Bratislava
(e-mail: pavol.seman@stuba.sk)
**** ATEC automatizačná technika s.r.o*
(e-mail: stanislav.bodi@atec-at.sk)

Abstract: This paper gives information about development of laboratory device - simulator for automotive applications. The design of simulator allows the use of different setups for simulating braking of vehicle's front and rear wheel as well as acceleration of front and rear wheel driven vehicle. Main use of simulator is development and evaluation of control strategies for anti-lock braking system and traction control system, but device can also be used as general mechatronic system with non-linear behaviour for testing of control system designs.

1. INTRODUCTION

The anti-lock braking system (ABS) and traction control system (TCS) are important safety systems that monitors and controls wheel slip during vehicle braking and acceleration (Gillespie 1992). ABS improves vehicle stability and reduces stopping distances when braking on slippery road surfaces. Rolling wheels usually have more friction available, than locked wheels. This is also used by TCS to reach better traction during acceleration. Control algorithms for both systems are being continuously developed and optimized for better performance. The aim of this work was to design laboratory device that can be used for experimental evaluation of suggested improving control techniques.

2. ABS/TCS SIMULATOR CONCEPT AND DESIGN

For purposes of anti-lock braking system analysis a quarter vehicle model is often considered (Solyom 2004). This model consists of single wheel attached to a mass – Fig. 1.

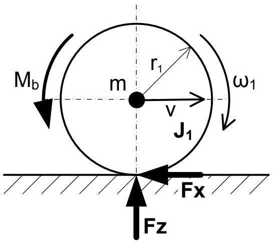


Fig. 1. Quarter model of vehicle: m – vehicle mass, v – vehicle longitudinal velocity, J_1 – wheel inertia, r_1 – wheel radius, ω_1 – angular velocity of wheel, M_b – brake torque, F_x – wheel/road friction force, F_z – vertical force

The equations of motion of quarter vehicle model are given by:

$$J_1\dot{\omega}_1 + M_b - F_x r_1 = 0 \tag{1}$$

$$m\dot{v} - F_x = 0 \tag{2}$$

Maximum braking force must be less then available friction:

$$|F_x| \le F_z \mu \tag{3}$$

where μ is the friction coefficient between road and tire.

For the simulator design the vehicle dynamics is substituted using second wheel as shown in Fig. 2 .

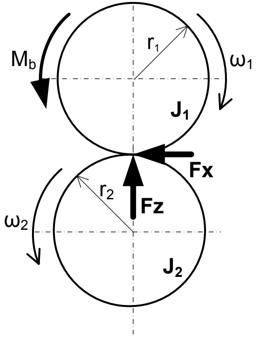


Fig. 2. Quarter model of simulator: J_1 – vehicle wheel inertia, r_1 – wheel radius, ω_1 – angular velocity of wheel, M_b – brake torque, F_x – wheel/road friction force, F_z – vertical force, J_2 – vehicle dynamics substituting wheel inertia, r_2 – substituting wheel radius

Equation (1) is valid for new system too, while (2) is changed:

$$\frac{J_2\dot{\omega}_2}{r_2} - F_x = 0 \tag{4}$$

New configuration is well suited for construction of physical laboratory model and forms a basis for ABS/TCS simulator design – Fig. 3, 4.

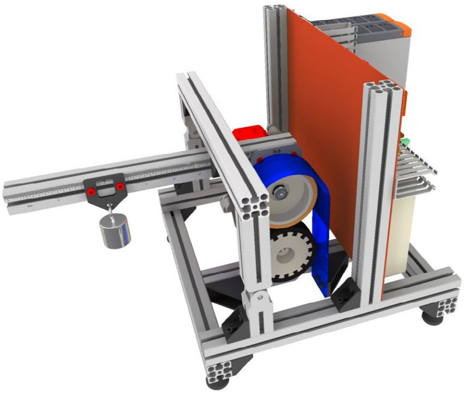


Fig. 3. Mechanics of ABS/TCS simulator – concept

The design of simulator allows the use of different setups for simulating braking of vehicle's front and rear wheel as well as acceleration of front and rear wheel driven vehicle. This functionality is provided by simulator frame kinematics and by possibility to control the torque of both wheels via servo-drives. Table I shows available simulator configurations for different vehicle types and driving conditions.

TABLE I
Description of ABS/TCS Simulator Configurations

Configuration	Upper wheel rotation	Upper wheel torque	Lower wheel torque
Front wheel braking	positive	negative, ABS controlled	0
Rear wheel braking	negative	positive, ABS controlled	0
Front wheel braking downhill	positive	negative, ABS controlled	positive, constant
Rear wheel braking downhill	negative	positive, ABS controlled	negative, constant
Front wheel acceleration	positive	positive, TCS controlled	0
Rear wheel acceleration	negative	negative, TCS controlled	0
Front wheel acceleration uphill	positive	positive, TCS controlled	negative, constant
Rear wheel acceleration uphill	negative	negative, TCS controlled	positive, constant
Friction estimation	positive	positive, velocity control	negative, velocity control

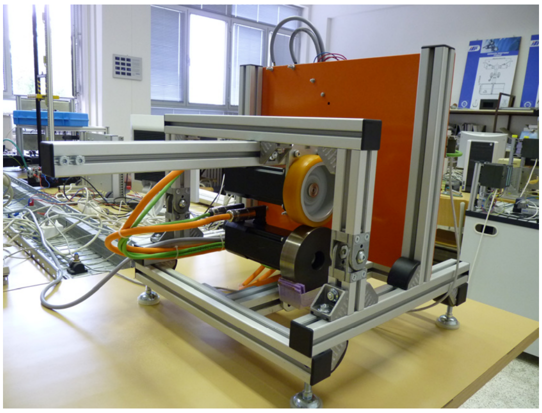


Fig. 4. Mechanics of ABS/TCS simulator – real system

3. CONTROL SYSTEM ARCHITECTURE

Hardware of control system for simulator is based on standard industrial components of B&R automation. Central part of solution is X20 PLC which runs the control program and controls the torques of both simulator wheels via ACOPOS servo drives. Device utilizes utilises Ethernet Powerlink bus, which is used for fast torque and velocity closed loop control of servo drives, as well as for data acquisition of process data into PLC memory during runtime. Human machine interface runs as a task on the same PLC and is available over Ethernet network via VNC console. Main parts of simulator's control system architecture are shown at Fig. 5 and Fig. 6.

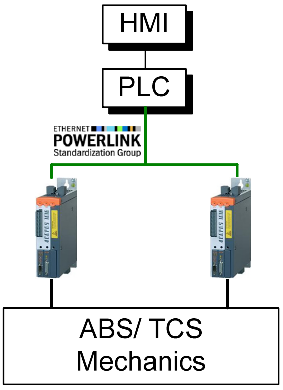


Fig. 5. Control system architecture for simulator

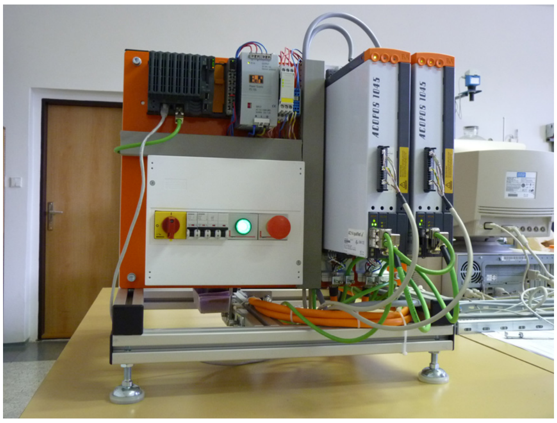


Fig. 6. Control system of ABS/TCS simulator – real system

Software of simulator can be split into two parts – system software and user software. System software deals with safe control of servo drives, data acquisition and data logging tasks and human machine interface services. User part of software implements actual control strategies for anti-lock braking or traction control. This software is prepared within MATLAB/Simulink environment. The code for PLC is then automatically generated using Real Time Workshop addition of and incorporated into the structure of automation project for industrial system. Architecture of software is at Fig. 7.

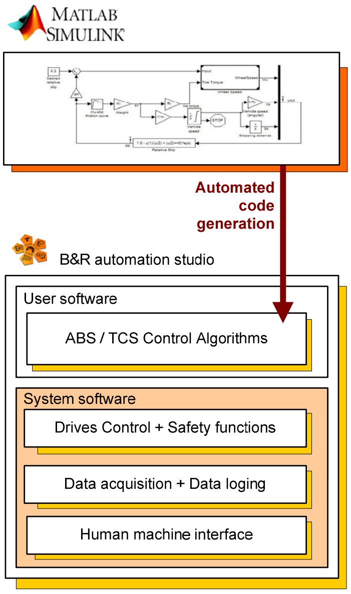

MATLAB
SIMULINK
Automated code generation
B&R automation studio
User software
ABS / TCS Control Algorithms
System software
Drives Control + Safety functions
Data acquisition + Data loging
Human machine interface

Fig. 7. Software solution architecture for simulator

4. EXPERIMENTAL WORKS

First experimental works were aimed at estimation of slip friction curves for different tire/road conditions – dry rough surface and wet smooth surface. Afterwards implementation of anti-lock braking was experimentally verified on wet surface.

4.1 Slip friction estimation for dry concrete road surface

The task of this experiment was the estimation of friction torque between wheel with rubber surface (representing tire) and wheel with concrete surface (representing road). Photo of experimental setup is at Fig. 8.

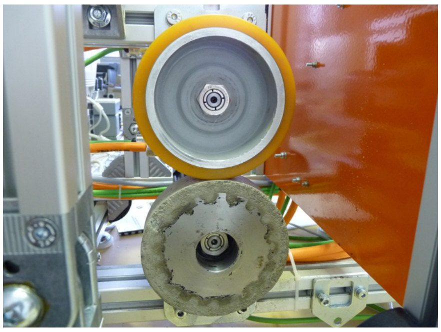


Fig. 8. Experiment setup for dry concrete road friction curve estimation

During experiment rotational velocity of “road” wheel was maintained at constant value. Starting rotational velocity of “tire” wheel was the same with opposite direction of rotations. During experiment “tire” wheel velocity was decreased in steps while measuring torques at both servo drives. These values were corrected for the effect of mechanical losses. Resulting slip friction curve is at Fig. 9.

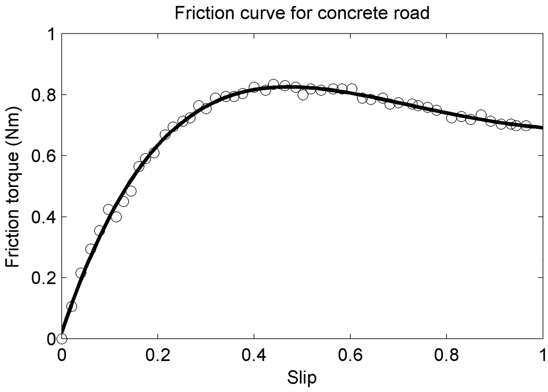


Fig. 9. Friction torque curve for rubber and concrete

4.2 Slip friction estimation for wet slippery road surface

The task was to estimate the available friction between wheel with rubber surface and wheel with slippery wet surface. Photo of experimental setup is at Fig. 10, which also shows maintaining of thin wet film at “road” wheel. Experimental procedure and data processing was the same as for previous experiment. Resulting slip friction curve is at Fig. 11.

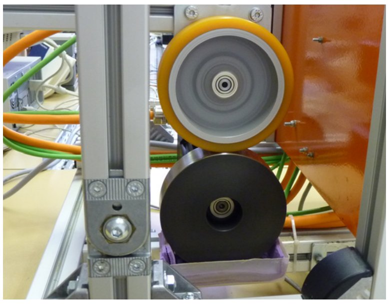


Fig. 10. Experiment setup for wet smooth road friction curve estimation

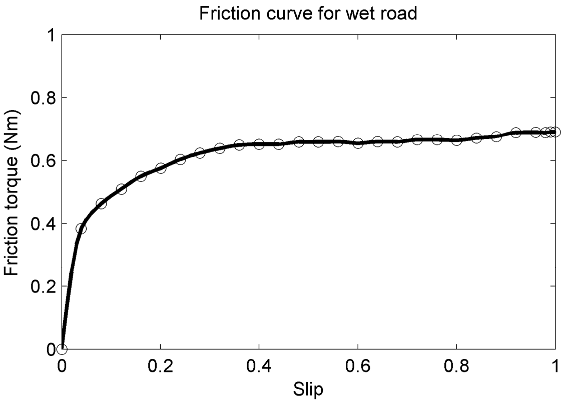


Fig. 11. Friction torque curve for rubber and for wet smooth surface

4.3 Anti-lock braking on wet slippery road

Experimentally estimated friction curves were used during ABS control synthesis. Resulting PI based controller was verified in front wheel braking configuration with and without anti-lock function active. Results are given in Fig. 12 and 13.

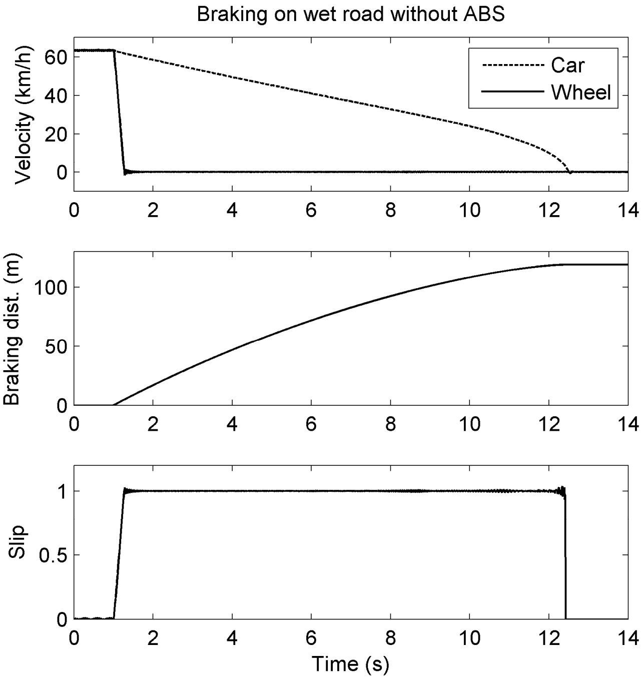


Fig. 12. Braking with ABS function off

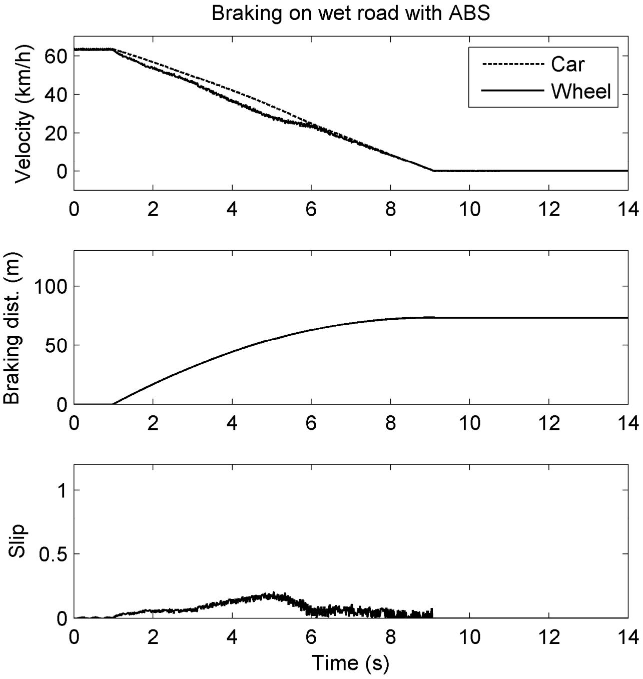


Fig. 13. Braking with ABS function on

In both cases simulator wheels were slowly accelerated to starting speed. Braking was activated in time of 1 s. During braking with ABS off the upper braking wheel locked after short time and remained in this state throughout all braking process (Fig. 12). ABS controller in next experiment tried to maintain the slip at value of 0.4. This effort was only partially successful, but effect of active ABS is clearly visible (Fig. 13) – braking time was reduced by app. 30 % and braking distance was reduced by app. 40 %, compared to experiment without ABS function.

5. CONCLUSIONS

Paper describes laboratory device for anti-lock braking system and traction control system simulation. Main use of simulator is development and evaluation of control strategies for anti-lock braking system and traction control system.

First experimental results are promising and proved functionality of simulator design. Simulator is still in development phase and works are in progress to improve its features.

Future works include creation and adjusting of mathematical model of simulator, optimization of simulator software and development of HMI interface.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support granted by the Slovak Research and Development Agency under the contracts VMSP-P-0030-09 and APVV-0160-07, and reviewers' comments. This research is also supported by the grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism. This project is co-financed from the state budget of the Slovak Republic.

REFERENCES

Gillespie, T. D. (1992). Fundamentals of Vehicle Dynamic. SAE International, ch. 10.

Solyom, S. (2004). Control of Systems with Limited Capacity. PhD thesis. Lund Institute of Technology, Sweden.

Author Index

List of Participants

Belgium

Krommer, Gabor, Brussels, Poslanecký klub EL'S v Európskom parlamente

Croatia

Bencic, Goran, Zagreb, Koncar – Power Plant and Electric Traction Enginnering / Faculty of Electrical Enginnering and Computing, University of Zagreb
Bobanac, Vedran, Zagreb, Faculty of Electrical Engineering and Computing, University of Zagreb
Iles, Sandor, Zagreb, University of Zagreb, Faculty of Electrical Engineering and Computing (FER)
Petrović, Vlaho, Zagreb, Faculty of Electrical Engineering and Computing, University of Zagreb
Spudic, Vedrana, Zagreb, University of Zagreb Faculty of Electrical Engineering and Computing Department of Control and Computer Engineering

Czech Republic

Bobál, Vladimír, Zlin, Department of Process Control, Faculty of Applied Informatics, Tomas Bata University in Zlin
Cvejn, Jan, Pardubice, Univerzita Pardubice, Fakulta elektrotechniky a informatiky
Doležel, Petr, Pardubice, University of Pardubice, Faculty of Electrical Engineering and Informatics, Department of Process Control
Dostál, Petr, Zlín, Department of Process Control, Faculty of Applied Informatics, Tomas Bata University in Zlín
Dušek, František, Pardubice, University of Pardubice Faculty of Electrical Engineering and Informatics Department of Process Control
Fribert, Miroslav, Pardubice, Univerzita Pardubice, katedra elektrotechniky
Haniš, Tomáš, Prague, Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Control Engineering
Havlíček, Libor, Pardubice, University of Pardubice, Department of Process Control and Computer Techniques
Honc, Daniel, Pardubice, Department of Process Control Faculty of Electrical Engineering and Informatics University of Pardubice
Javůrek, Milan, Pardubice, University of Pardubice, Faculty of Electronics and Informatics, Department of Control Systems
Jirkovský, Jaroslav, Praha 8, HUMUSOFT s.r.o.
Krynský, David, Praha, Nuclear Research Institute Řež plc, division Energoprojekt Praha
Kubina, Adam, Praha, Nuclear Research Institute Řež plc, division Energoprojekt Praha
Kucera, Vladimir, Praha 6, Czech Technical University in Prague, FEE, Department of control engineering
Kučera, Vladimír, Praha 6, Faculty of Electrical Engineering Czech Technical University in Prague
Macků, Lubomír, Zlín, Faculty of Applied Informatics, Tomas Bata University in Zlín
Macháček, Jiří, Pardubice, Univerzita Pardubice
Mareš, Jan, Prague 6, Institute of Chemical Technology Prague Faculty of Chemical Engineering Department of Computing and Control Engineering
Petrová, Jana, Praha 6, Department of Computing and Control Engineering Institute of Chemical Technology in Prague
Rathouský, Jan, Praha 6, Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
Severa, Ondřej, Plzen 1, University of West Bohemia in Pilsen, Czech Republic
Slavíková, Petra, Prague 6, Institute of Chemical Technology Prague Department of Computing and Control Engineering
Sobota, Jaroslav, Plzeň 1, University of West Bohemia in Pilsen, Czech Republic
Šebek, Michael, Praha 6, Czech Technical University in Prague, Faculty of Electrical Eng.
Trnka, Pavel, Prague, Czech Technical University in Prague Faculty of Mechanical Engineering Department of Instrumentation and Control Engineering

Estonia

Belikov, Juri, Tallinn, Control Systems Department, Institute of Cybernetics at TUT
Kaldmäe, Arvo, Tallinn, Department of Control Systems, Institute of Cybernetics at Tallinn University of Technology
Kotta, Ülle, Tallinn, Dept. of Control Systems Institute of Cybernetics at TUT
Mullari, Tanel, Tallinn, Institute of Cybernetics at Tallinn University of Technology
Tőnso, Maris, Tallinn, Dept. of Control Systems Institute of Cybernetics at TUT

France

Henrion, Didier, Toulouse, LAAS-CNRS Toulouse

Germany

Al-Rashedi, Naef, Hagen, Fern-University in Hagen, Faculty of Informatic and Mathematic, Process Control Research Group.
Bahník, Pavol, Hagen, FernUniversität in Hagen
Behrendt, Stefan, Wismar, Hochschule Wismar, University of Applied Sciences, Faculty of Engineering
Gerke, Michael, Hagen, FernUniversitaet in Hagen Department of Electrical Engineering Control Systems Engineering group
Haber, Robert, Köln, University of Applied Science Cologne, Institute Process Engineering
Hippe, Peter, Erlangen, Lehrstuhl für Regelungstechnik Universität Erlangen-Nürnberg
Jelenčiak, František, Hagen, FernUniversität in Hagen
Masar, Ivan, Hagen, FernUniversität in Hagen MI/PRT
Seibold, Peter, Hagen, FernUniversität in Hagen

Norway

Sivalingam, Selvanathan, Trondheim, Post doctoral researcher, Department of Engineering Cybernetics, NTNU, Norway

Poland

Kowalewski, Adam, Kraków, Institute of Automatics, AGH University of Science and Technology
Ogonowski, Zbigniew, Gliwice, Institute of Automatic Control, Silesian University of Technology

Russia

Alexeev, Alexander, Tomsk, Department of Integrated Computer Control Systems, Tomsk Polytechnic University
Bobko, Yevgeny, Vladivostok, Far Eastern Federal University

Slovak Republic

Bisták, Pavol, Bratislava, Slovak University of Technology (Organization unit: Fac. of El. Eng. and Inf. Technology)
Blahová, Lenka, Bratislava, Slovak University of Technology, Faculty of Chemical and Food Technology. Institute of Information Engineering, Automation, and Mathematics. Department of Information Engineering and Process Control
Csambal, Jozef, Bratislava, Slovak University of Technology in Bratislava, Faculty of Mechanical Engineering Institute of Automation, Measuring and Applied Informatics Námestie slobody 17 812 31 Bratislava
Čirka, L'uboš, Bratislava, Department of Information Engineering and Process Control FCFT STU
Čižniar, Michal, Bratislava, Honeywell Process Solutions, Honeywell s.r.o.
Duchoň, František, Bratislava, FEI STU
Dvoran, Ján, Bratislava, Department of Information Engineering and Process Control FCHPT STU
Fikar, Miroslav, Bratislava, Department of Information Engineering and Process Control FCFT STU
Filasová, Anna, Kosice, Technical University of Kosice, Dept. of Cybernetics and AI
Folvarčík, Pavol, Bratislava, Slovenská technická univerzita, Fakulta informatiky a informačných technológií,Ústav riadenia a priemyselnej informatiky.
Fonod, Robert, Kosice, Technical University of Kosice, Department of Cybernetics and AI
Halás, Miroslav, Bratislava, Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and IT, Slovak University of Technology
Holič, Ivan, Bratislava, Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and IT, Slovak University of Technology, Slovak Republic
Huba, Mikuláš, Bratislava, STU FEI Bratislava
Janík, Zoltán, Bratislava, Faculty of Electrical Engineering and Information Technology Slovak University of Technology

Jendroľ, Vladimír, Giraltovce, Slovenská technická univerzita v Bratislave Fakulta elektrotechniky a informatiky
Juhás, Martin, Bratislava, Department of Automation, Measurement and Applied Informatics; Faculty of Mechanical Engineering; Slovak University of Technology in Bratislava
Kalmárová, Andrea, Bratislava, Department of Information Engineering and Process Control
Kalúz, Martin, Bratislava, Department of Information Engineering and Process Control FCFT STU, Slovak Republic
Kardoš, Ján, Bratislava, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava
Karovič, Jaroslav, Bratislava, Country Manager,Invensys Systems sro, Division Foxboro
Karšaiová, Mária, Bratislava, Department of Information Engineering and Process Control, Faculty of Chemical and Food Technology, Slovak University of Technology
Keseli, Roland, Sala, ProCS Ltd. Sala
Kopačka, Matúš, Bratislava, Slovak University of Technology in Bratislava Faculty of Mechanical Engineering Institute of Automation, Measuring and Applied Informatics
Kozáková, Alena, Bratislava, URPI FEI STU
Krokavec, Dušan, Kosice, Department of Cybernetics and AITechnical University of KosiceFaculty of Electrical Engineering and Informatics
Kvasnica, Michal, Bratislava, Department of Information Engineering and Process Control FCFT STU
Makýš, Miroslav, Bratislava, Slovenská technická univerzita Fakulta informatiky a informačných technológií
Mészáros, Alojz, Bratislava, KIRP STU
Nguyen, Quang Thuan, Bratislava, Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology , Slovak University of Technology in Bratislava
Oravec, Juraj, Bratislava, FCHPT STU
Osuský, Jakub, Bratislava, FEI STU
Paulen, Radoslav, Bratislava, Institute of Information Engineering, Automation and Mathematics STU in Bratislava
Pestun, Ivan, Bratislava, Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and IT, Slovak University of Technology
Petrík, Matej, Bratislava, Institute of Applied Informatics at FIIT SUT
Podmajerský, Marian, Bratislava, Department of Information Engineering and Process Control FCHPT STU
Rauová, Ivana, Bratislava, Institute of Information Engineering, Automation, and Mathematics; Faculty of Chemical and Food Technology; Slovak University of Technology in Bratislava
Rosinová, Danica, Bratislava, Department of Automatic Control Systems, STU, Faculty of Electrical Engg. and IT Bratislava Slovakia
Sedlák, Michal, Bratislava, Faculty of Electrical Engineering and Information Technology Slovak University of Technology
Szucs, Alexander, Bratislava, Department of Information Engineering and Process Control FCFT STU, Slovak Republic
Števek, Juraj, Bratislava, Phd student
Ťapák, Peter, Bratislava 1, FEI STU
Vagač, Stanislav, Bratislava, Department of Information Engineering and Process Control
Vachálek, Ján, Bratislava, Ústav automatizácie, merania a aplikovanej informatiky, Strojnícka fakulta, Slovenská technická univerzita v Bratislave
Valo, Richard, Bratislava, Institute of Information Engineering, Automation, and Mathematics FCFT STU in Bratislava
Vasičkaninová, Anna, Bratislava, KIRP, FCHPT
Veselý, Vojtech, Bratislava, Institute of Control and II, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology
Vörös, Jozef, Bratislava, Slovak Technical University Faculty of Electrical Engineering and Information Technology Department of Automation and Control
Závacká, Jana, Bratislava, Department of Information Engineering and Process Control , Faculty of Chemical and Food Technology, STU, Slovakia
Zuščíková, Monika, Bratislava, Slovenská technická univerzita, Strojníka fakulta, Ústav automatizácie, merania a aplikovanej informatiky, Bratislava
Žáková, Katarína, Bratislava, ÚRPI FEI STU

Slovenia

Vrancic, Damir, Ljubljana, J. Stefan Institute

South Africa

Asante, Kwaku Effa, Kyalami, None

Switzerland

Bonvin, Dominique, Lausanne, EPFL Lausanne
Jones, Colin, Lausanne, EPFL Lausanne
Noga, Rafal, Geneva, European Organization for Nuclear Research
Papafotiou, George, Baden-Dättwil, ABB Switzerland
Zeilinger, Melanie, Lausanne, EPFL Lausanne

Thailand

Sraman, Liton, Khon Kaen, Mahamakut Buddhist University, Thailand