Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

17$^{\mathrm{th}}$ International Conference on Process Control 2009

Hotel Baník, Štrbské Pleso, Slovakia, June 9 – 12, 2009

ISBN 978-80-227-3081-5

http://www.kirp.chtf.stuba.sk/pc09

Editors: M. Fikar and M. Kvasnica

Chalupa, P.: Predictive Control Using Self-Tuning Model Predictive Controllers Library, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 17th International Conference on Process Control '09*, Štrbské Pleso, Slovakia, 419–425, 2009.

Full paper online: http://www.kirp.chtf.stuba.sk/pc09/data/abstracts/094.html

PREDICTIVE CONTROL USING SELF-TUNING MODEL PREDICTIVE CONTROLLERS LIBRARY

P. Chalupa

Tomas Bata University in Zlin
Faculty of Applied Informatics
Department of process Control
Nad Straněni 4511, 760 05 Zlin, Czech Republic
e-mail : chalupa@fai.utb.cz

Abstract: This paper is focused on a library of adaptive controllers which use model predictive control design. The Self-Tuning Model Predictive Controllers Library (STuMPCoL) has been designed in the MATLAB / Simulink environment and contains a framework for design and testing of Model predictive control approach combined with on-line identification of controlled process (self-tuning control). The paper presents techniques incorporated into the STuMPCoL and describes structure of self-tunig model predictive controllers. Moreover, the contribution includes some results obtained by simulation and real-time verification of the library.

Keywords: self-tuning control, model-predictive control, Simulink, MATLAB.

1 INTRODUCTION

Model Predictive Control (MPC) is one of possible control approaches based on model of a controlled system. Contrary to most other approaches, MPC is based on future course of inputs of control circuit. Then, a finite number of future control samples can be computed by minimizing a criterion. The computation of future control signal samples can be performed in each sample steps and only the first action is applied to the plant. This approach is called receding horizon MPC (Kwon et al. 2005) and is used further in this papers.

This paper presents a basic frame of the Self-Tuning Model Predictive Controllers Library (STuMPCoL). This library has been created to provide unified framework for testing and design of model predictive controllers for real-time application. The library is intended for usage by both students of process control and control engineers.

The MPC is used to produce either control signal directly or setpoints for simpler controllers (e.g. PID) (Sunan et al. 2002). The first approach is used further in this paper. The computation of the controlled signal is based on a model of a controlled system. There are two basic approaches of obtaining system's model: mathematical-physical analysis of the system and black box approach.

The mathematical-physical analysis of the system and subsequent derivation of the relations between system inputs and outputs provides general model which can be valid for a whole range of system's inputs and states. On the other hand, there is usually a lot of unknown constants and relations when performing mathematic-physical analysis. Therefore, modelling by mathematic-physical analysis is suitable for simple controlled systems with small number of parameters or for obtaining basic information about the system (range of gain, rank of suitable sample time, etc.).

The black box approach to the modelling is based on analysis of input and output signals of the system. The main advantage of this approach lies in the possibility of usage the same identification algorithm for different controlled systems. In addition, the knowledge of physical principle of controlled system and solution of possibly complicated set of mathematical equation is not required. On the other hand, model obtained by black box approach is generally valid only for signals it was calculated from. For example, if only low frequency changes of input signals were used to obtain the model, this model need not be us-

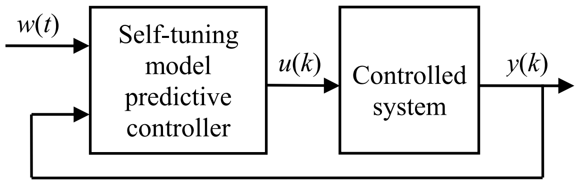

Fig. 1. Control circuit with Self-tuning MPC

able for high frequency changes of input signals (Chalupa and Bobál 2007).

The model of the controlled process can also be obtained by on-line identification. The on-line identification can be encapsulated into model predictive controller. The scheme of a simple control circuit with self-tuning predictive controller is shown in Fig. 1.

Note that the reference signal is marked as $w(t)$. This means that the course of reference signal is sent to the controller, not only the current value $w(k)$. Self-tuning control is based on-line identification of controlled process and controller synthesis which uses results from the identification. (Bobál et al. 2005) Thus, each self-tuning predictive controller consists of two relatively stand-alone parts:

- on-line identification
- model predictive controller

The internal structure of self-tuning model predictive controller is presented in Fig. 2. The output of the on-line identification block – $\hat{\mathbf{\Theta}}(k)$ – represents current estimates of parameters of controlled process.

2 ON-LINE IDENTIFICATION METHODS

Various discrete parametric models are used to describe dynamic behaviour of controlled systems. Overview of these models is given in (Ljung, 2001). A general input-output linear model for a single-

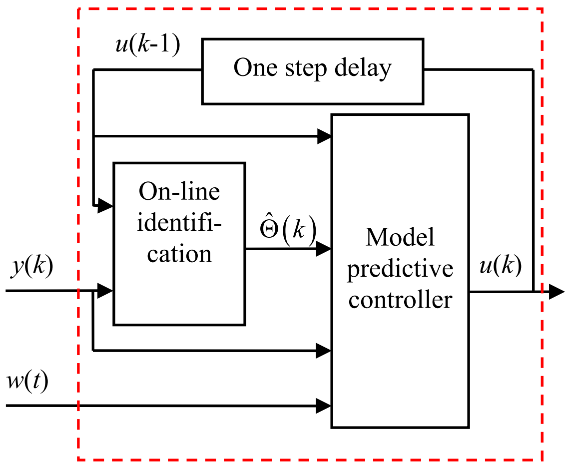

Fig. 2. Internal structure of Self-tuning MPC

output system with input u and output y can be written:

$$A(z^{-1})y(t)=\sum_{i=1}^{N}\frac{B_i(z^{-1})}{F_i(z^{-1})}z^{-d_i}u_i(t)+\frac{C(z^{-1})}{D(z^{-1})}n(t) \quad (1)$$

where N represents the number of measurable input signals, $y(t)$, $u_i(t)$ and $n(t)$ are output signal, input signals and immeasurable error signal respectively and d_i corresponds to the delay of i-th input. $A(z^{-1})$, $B_i(z^{-1})$, $F_i(z^{-1})$, $C(z^{-1})$ and $D(z^{-1})$ are polynomials in the shift operator z^{-1}. Widely used simplification is general model (1) is ARX model:

$$A(z^{-1})y(t)=B(z^{-1})z^{-d}u(t)+n(t) \quad (2)$$

Then the transfer function of model of identified system is assumed to be in the following form:

$$G(z)=\frac{B(z^{-1})}{A(z^{-1})}z^{-d}=\frac{b_1z^{-1}+b_2z^{-2}+\ldots+b_mz^{-m}}{1+a_1z^{-1}+a_2z^{-2}+\ldots+a_nz^{-n}}z^{-d} \quad (3)$$

Then it is possible to write en equation for computing the output of the system in k-th step:

$$y(k)=\mathbf{\Theta}^T(k)\cdot\mathbf{\Phi}(k)+n(k) \quad (4)$$

where $n(k)$ represents the influence of an immeasurable disturbances and the vector of the parameters of the controlled system model $\mathbf{\Theta}(k)$ and the data vector $\mathbf{\Phi}(k-1)$ are formed as follows:

$$\begin{aligned}\mathbf{\Theta}(k)&=\left[a_1,a_2,\ldots,a_n,b_1,b_2,\ldots,b_m\right]^T\\ \mathbf{\Phi}(k)&=\left[-y(k-1),\ldots,-y(k-n),u(k-1),\ldots,u(k-m)\right]^T\end{aligned} \quad (5)$$

The identification problem is formulated as a process of finding the $\mathbf{\Theta}(k)$ vector with respect to some criterion. Exact values of parameters are unknown during the identification process and just the vector of parameter estimations is used:

$$\hat{\mathbf{\Theta}}(k)=\left[\hat{a}_1,\hat{a}_2,\ldots,\hat{a}_n,\hat{b}_1,\hat{b}_2,\ldots,\hat{b}_m\right]^T \quad (6)$$

The aim of the identification process is then make the estimations $\hat{\mathbf{\Theta}}(k)$ as close as possible to the actual parameters $\mathbf{\Theta}(k)$.

2.1 Recursive least squares method

The recursive least squares method (RLSM) is based on minimization of sum of squares of differences between actual system outputs and outputs estimated on base of system model. If the k-th identification steps is performed and data corresponding to r previous system inputs and outputs are available, the criterion to be minimized can be formulated as follows:

$$J = \frac{1}{2}\left[\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right]^T \left[\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right] \tag{7}$$

where $\boldsymbol{y}(k)$ is the vector of system outputs, and $\hat{\mathbf{y}}(k)$ is the vector of system outputs estimations. With respect to (4), each system output estimation can be written as

$$\hat{y}(i) = \mathbf{\Phi}^T(i)\hat{\mathbf{\Theta}}(k) \tag{8}$$

The resulting equation for parameter estimations update is:

$$\hat{\mathbf{\Theta}}(k+1) = \hat{\mathbf{\Theta}}(k) + \frac{\mathbf{C}(k)\mathbf{\Phi}(k)}{1+\mathbf{\Phi}^T(k)\mathbf{C}(k)\mathbf{\Phi}(k)} \cdot \left[y(k) - \mathbf{\Phi}^T(k)\hat{\mathbf{\Theta}}(k)\right] \tag{9}$$

and the covariance matrix is updated in each sample time according to the following equation:

$$\mathbf{C}(k+1) = \mathbf{C}(k) - \frac{\mathbf{C}(k)\mathbf{\Phi}(k)\mathbf{\Phi}^T(k)\mathbf{C}(k)}{1+\mathbf{\Phi}^T(k)\mathbf{C}(k)\mathbf{\Phi}(k)} \tag{10}$$

The covariance matrix $\boldsymbol{C}$ is usually initialized as a diagonal matrix with elements 10^3 on the main diagonal (Hang *et al.* 1993). The main diagonal of covariance matrix $\boldsymbol{C}$ contains dispersions of identified parameters an thus if the initial parameter estimations are known to be close to the actual values, the initial values of elements on the main diagonal are to be smaller.

2.2 Recursive least squares method with exponential forgetting

When using the least squares method, the influence of all pairs of identified system inputs and outputs to the parameters estimations are the same. This property can be inconvenient for example when identifying the system with time-varying parameters. In this case, it is better to use least squares method with exponential forgetting where the influence of newer data to the parameters estimations is greater then the influence of older data. The criterion to be minimized is in the following form:

$$J = \frac{1}{2}\left[\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right]^T \mathbf{W}\left[\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right] \tag{11}$$

where $\boldsymbol{W}$ is a diagonal weight matrix:

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \varphi & 0 & \dots & 0 \\ 0 & 0 & \varphi^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & \varphi^{r-1} \end{bmatrix} \tag{12}$$

And the φ is a forgetting coefficient which is assumed to be in range $0 < \varphi \le 1$. The RLSM with exponential forgetting can be transferred to pure RLSM by selecting $\varphi=1$. The lower value of φ denotes more rapid forgetting of older data and thus smaller influence of older data to resulting parameter estimations. The weight the data of k-q-th step affect current estimates is:

$$w(k-q) = \varphi^q \tag{13}$$

The derivation of recursive version of the algorithm is similar to the derivation used for pure least squares method and leads to the following equation for covariance matrix update:

$$\mathbf{C}(k+1) = \frac{1}{\varphi}\left[\mathbf{C}(k) - \frac{\mathbf{C}(k)\mathbf{\Phi}(k)\mathbf{\Phi}^T(k)\mathbf{C}(k)}{\varphi+\mathbf{\Phi}^T(k)\mathbf{C}(k)\mathbf{\Phi}(k)}\right] \tag{14}$$

The parameter estimations are updated in following way:

$$\hat{\mathbf{\Theta}}(k+1) = \hat{\mathbf{\Theta}}(k) + \frac{\mathbf{C}(k)\mathbf{\Phi}(k)}{\varphi+\mathbf{\Phi}^T(k)\mathbf{C}(k)\mathbf{\Phi}(k)} \cdot \left[y(k) - \mathbf{\Phi}^T(k)\hat{\mathbf{\Theta}}(k)\right] \tag{15}$$

Initial value of matrix $\boldsymbol{C}$ is recommended to be chosen as a diagonal matrix with elements 10^3 on the main diagonal (Bobál *et al.* 2005). Choosing of coefficient φ is individual and depends on the relation between identification sample time and speed of identified system but usually is taken from range $\langle 0.90, 0.99 \rangle$.

2.3 Recursive least squares method with adaptive directional forgetting

The exponential forgetting method can be further improved by adaptive directional forgetting (Kulhavý, 1985] which changes forgetting coefficient with respect to changes of input and output signals of identified system. Parameter estimations are updated using recursive equation

$$\hat{\mathbf{\Theta}}(k+1) = \hat{\mathbf{\Theta}}(k) + \frac{\mathbf{C}(k)\mathbf{\Phi}(k)}{1+\xi} \cdot \left[y(k) - \mathbf{\Phi}^T(k)\hat{\mathbf{\Theta}}(k)\right] \tag{16}$$

where the scalar ξ is defines as

$$\xi = \mathbf{\Phi}^T(k)\mathbf{C}(k)\mathbf{\Phi}(k) \tag{17}$$

Covariance matrix $\boldsymbol{C}$ is updated in each identification step according to the following equation:

$$\begin{aligned} \mathbf{C}(k+1) &= \mathbf{C}(k) - \frac{\mathbf{C}(k)\mathbf{\Phi}(k)\mathbf{\Phi}^T(k)\mathbf{C}(k)}{\varepsilon^{-1}+\xi} \\ \varepsilon &= \varphi(k) - \frac{1-\varphi(k)}{\xi} \end{aligned} \tag{18}$$

The forgetting coefficient is adapted with respect to courses of input and output signals according to following equation:

$$\varphi(k+1)=\frac{1}{1+(1+\rho)\left\{\ln(1+\xi)+\left[\eta\frac{\upsilon(k)+1}{1+\xi+\eta}-1\right]\frac{\xi}{1+\xi}\right\}} \tag{19}$$
$$\eta=\frac{\left[y(k)-\widehat{\mathbf{\Theta}}^T(k)\mathbf{\Phi}(k)\right]^2}{\lambda(k)}$$

and the scalars $\upsilon(k)$, $\lambda(k)$ and η are updated in each identification step in the following way:

$$\upsilon(k+1)=\varphi(k)\left[\upsilon(k)+1\right];$$
$$\lambda(k+1)=\varphi(k)\left\{\lambda(k)+\frac{\left[y(k)-\widehat{\mathbf{\Theta}}^T(k)\mathbf{\Phi}(k)\right]^2}{1+\xi}\right\} \tag{20}$$

Recommended initial values of identification variables are (Bobál *et al.* 2005): $\varphi(0)=1$, $\lambda(0)=0.001$, $\upsilon(0)=10^{-6}$. Initial value of matrix $\boldsymbol{C}$ should be chosen as a diagonal matrix with elements 10^3 on the main diagonal. Parameter ρ states expected value of forgetting coefficient φ^* according to the following equation:

$$\rho=\frac{1-\varphi^*}{2\varphi^*} \tag{21}$$

3 MODEL PREDICTIVE CONTROL CRITERIA

Generally, the computation of control signal of model predictive controller is based on minimization of particular criterion (Kwon *et al.* 2005). General form of the model predictive control criterion can be written as:

$$J_{MPC}=f\left[\mathbf{e}(k),\mathbf{u}(k)\right] \tag{22}$$

where f is a scalar function of vector arguments $\mathbf{e}(k)$ and $\mathbf{u}(k)$. Vector $\mathbf{e}(k)$ represents future control errors while $\mathbf{u}(k)$ represents future samples of control signal:

$$\mathbf{e}(k)=\begin{bmatrix} w(k+1)-y(k+1)\\ w(k+2)-y(k+2)\\ \vdots\\ w(k+N)-y(k+N)\end{bmatrix}$$
$$\mathbf{u}(k)=\begin{bmatrix} u(k)\\ u(k+1)\\ \vdots\\ u(k+N_c-1)\end{bmatrix} \tag{23}$$

where integer N denotes prediction horizon and N_c stands for control horizon. The criterion (22) is minimized by finding optimal course of future control samples $\boldsymbol{u}_k$. The receding horizon is usually used: only finite number of future values is used in criterion and only the first element of the obtained control sequence is applied to the controlled system.

In case of proportional behaviour of the control system, the differences between control samples are widely used in the criterion in stead of control samples itself:

$$J_{MPC}=f\left(\mathbf{e}_k,\mathbf{\Delta u}_k\right) \tag{24}$$

Several basic types of criterion are mentioned in the following subchapters.

2.1 Quadratic criterion

Quadratic criterion (Sunan *et al.* 2002) is the most often used criterion in MPC design. For single input single output (SISO) systems the criterion can be written in general form:

$$J_{MPC}=\mathbf{e}_k{}^T\mathbf{Q}(k)\mathbf{e}_k+\mathbf{\Delta u}_k{}^T\mathbf{R}(k)\mathbf{\Delta u}_k \tag{25}$$

where $\boldsymbol{\Delta u}$ is a vector of future differences of control signal samples and square matrixes $\boldsymbol{Q}$ and $\boldsymbol{R}$ allows to set weighting of individual vector elements. Future outputs of the controlled system, and consequently control errors, are computed on base of its model. Control sequence is obtained by minimizing criterion (25).

Most real-time application uses simple structures of matrixes $\boldsymbol{Q}$ and $\boldsymbol{R}$. If a single input-single output model is used, the criterion (25) is sometimes simplified to the following form:

$$J_{MPC}=\sum_{j=1}^{N}e(k+j)^2+\lambda\cdot\sum_{j=1}^{N}\Delta u(k+j)^2 \tag{26}$$

where λ states ratio between weights of control errors and differences of control samples.

Process of minimizing of the criterion (25) or (26) can be rewritten to a quadratic programming problem:

$$J_{MPC}=\mathbf{u}_k{}^{\mathbf{T}}\mathbf{H}(k)\mathbf{u}_k+\mathbf{p}(k)\mathbf{u}_k \tag{27}$$

where $\mathbf{u}_k$ is a vector of future control signal samples to be computed. $\mathbf{H}$ and $\mathbf{p}$ are matrix and vector derived from λ and model parameters. Quadratic programming problem is usually solved numerically. This allows further constraints to be applied to vector $\mathbf{u}_k$.

2.2 Linear criterion

In case of linear criterion, the absolute values are used in the MPC criterion in stead of quadrates. The linear criterion has the following form:

$$J_{MPC}=\mathbf{Q}^T(k)\left|\mathbf{e}_k\right|+\mathbf{R}^T(k)\left|\mathbf{\Delta u}_k\right| \tag{28}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weight vectors of the size N x 1and N_c x 1respectively. The criterion is usually simplified to weighted sum of future control errors and control signal differences.

$$J_{MPC} = \sum_{j=1}^{N} \left| e(k+j) \right| + \lambda \cdot \sum_{j=1}^{N} \left| \Delta u(k+j) \right| \tag{29}$$

Process of minimizing of the criterion can be rewritten to a linear programming problem:

$$J_{MPC} = \mathbf{q}(k)\mathbf{u}_k; \qquad \mathbf{A}(k)\mathbf{u}_k < \mathbf{b}(k) \tag{30}$$

where $\mathbf{u}_k$ is a vector of future control signal samples to be computed. $\mathbf{A}$ is a matrix and $\mathbf{q}$ and $\mathbf{b}$ are matrix and vector derived from λ and model parameters. Linear programming problem is usually solved numerically, which allows further constraints to be applied to vector $\mathbf{u}_k$.

2.3 Min-max criterion

The min-max criterion was proposed to minimize the maximum absolute value of control error and control signal difference:

$$J_{MPC} = \max_{j=1}^{N} \left| e(k+j) \right| + \max_{j=1}^{N} \left(\lambda_j \cdot \left| \Delta u(k+j) \right| \right) \tag{31}$$

Likewise linear criterion, process of minimizing of the min-max criterion can be rewritten to a linear programming problem:

$$J_{MPC} = \mathbf{q}(k)\mathbf{u}_k; \qquad \mathbf{A}(k)\mathbf{u}_k < \mathbf{b}(k) \tag{32}$$

where $\mathbf{u}_k$ is a vector of future control signal samples to be computed. $\boldsymbol{A}$ is a matrix and $\boldsymbol{q}$ and $\boldsymbol{b}$ are matrix and vector derived from λ_j and model parameters.

4 THE STUMPCOL

The Self-Tuning Model Predictive Controllers Library encapsulates identification methods and MPC designs presented in previous chapters.

The models of the controlled process are mot restricted to ARX. The STuMPCoL provides on-line identification of following model types:

- ARX (AutoRegressive with Exogenous input)
- ARMAX (AutoRegressive Moving Average with Exogenous input)
- OE (Output Error)

These models can be identified using one of the following methods:

- Recursive Least Squares Method (RLSM)
- RLSM with exponential forgetting
- RLSM with directional forgetting

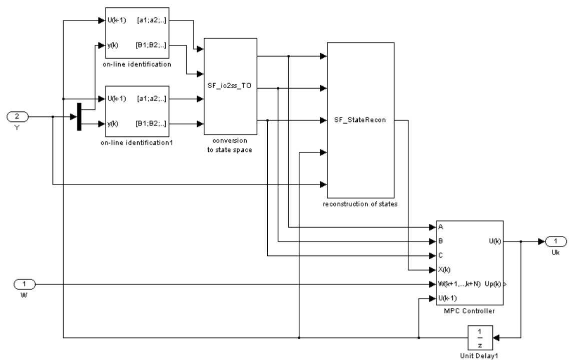

Fig. 3. Simulink scheme of Self-tuning MPC

- RLSM with adaptive directional forgetting

The identification function is designed for multi input single output (MISO) systems. This extends the area of applicability of the library even to multi input multi output (MIMO) systems.

The future control signal courses are computed on by minimizing one of following criteria:

- Quadratic
- Linear (sum of absolute values)
- Min-max (minimization of maximal absolute value)

The library controllers are not restricted to single input single output (SISO) systems. Following combinations of inputs and outputs of controlled system are covered in the library:

- SISO (Single Input Single Output)
- TISO (Two Inputs Single Output)
- TITO (Two Inputs Two Outputs)

Additional controllers for more complicated configurations can be designed by user on the basis of some appropriate library controller. Only standard MATLAB / Simulink techniques have been used during the library design and thus user-designed controllers can be derived relatively easily. A Simulink scheme of an internal structure of self-tuning model predictive controller for the TISO controlled system is presented in Fig. 3. The parameters of the controllers are entered using just standard Simuling dialogs as presented in Fig. 4.

5 SIMULATION VERIFICATION

The controllers from the STuMPCoL have been verified by control of various Simulink models. This chapter presents an illustrative example of control of a TITO linear continuous system.

The transfer matrix of the system was selected as:

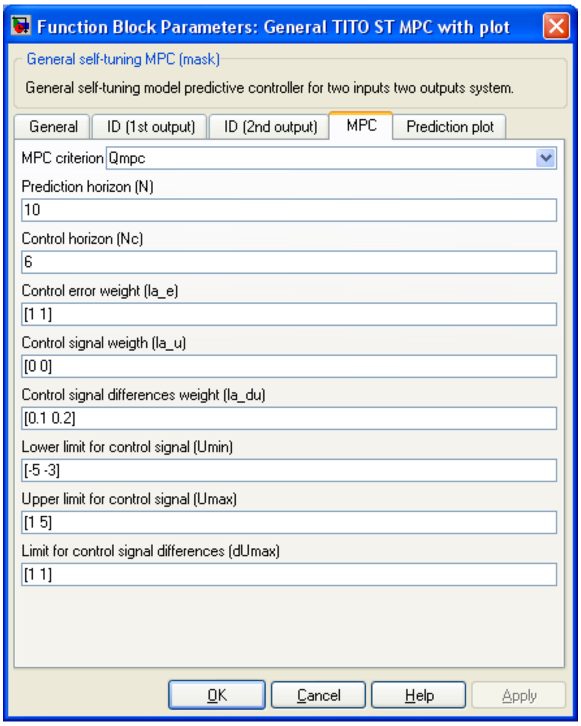

Fig. 4. Dialog for setting controller parameters

$$G(s)=\begin{bmatrix} \dfrac{-2}{s^2+3s+1} & \dfrac{3s+2}{s^2+3s+1} \\ \dfrac{3s+2}{s^2+s+1} & \dfrac{2}{s^2+s+1} \end{bmatrix} \qquad (33)$$

It can be seen that cross transfers play a great role in this system. Therefore it is impossible to control the system as two stand-alone SISO systems. The ARX model and RLSM with adaptive directional forgetting were used for on-line identification. No-apriori information was provided to the on-line identification function. Control signal sequence was obtained by minimizing quadratic criterion. Resulting control courses are presented in Fig. 5.

It can be observed that after initial phase, where parameter estimates did not represent the system behaviour well, a satisfactory control courses were obtained and the closed control loop behaves as an almost decoupled system.

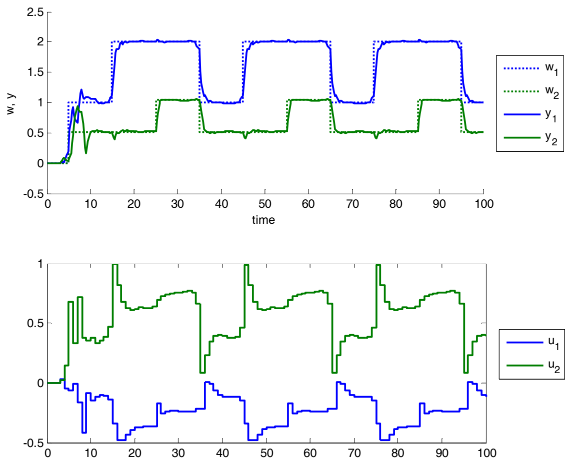

Fig. 5. Simulation control courses

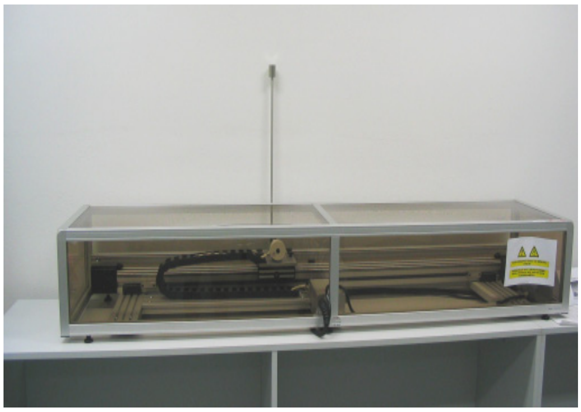

Fig. 6. PS600 Inverted Pendulum system

6 REAL TIME EXPERIMENT

The controllers from the library have been verified by real-time control of laboratory model. Control of Amira PS600 Inverted Pendulum is presented in this chapter. A photo of the system is shown in Fig. 6.

The pendulum can be considered a SITO system where input is control voltage of cart motor and outputs are cart position and pendulum angle. Pendulum should be held in the upright position (w_2=0) during control process. The resulting courses are presented in Fig. 7.

It can be observed that pendulum was stabilized in upright position and the cart moved along reference trajectory. The oscillations of control signal are caused by a block for compensating the friction of the cart. Control signal in range $\langle -2\text{V}, 2\text{V}\rangle$ does not cause any cart movement and thus only control signal outside this range were applied. Maximal allow-

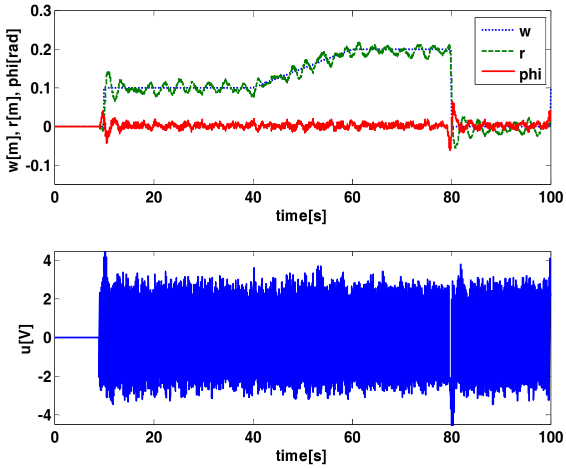

Fig. 7. Control courses of PS600 system

able range of control signal is from -10V to +10V but as can be seen in Fig. 7. The compensation of the friction was accomplished by the “Coulomb & Viscous Friction” Simulink block.

CONCLUSION

The Self-Tuning Model Predictive Controllers Library (STuMPCoL) was introduced in the paper. The library contains various predictive controllers based on discrete linear model of the controlled system. The controllers are based on quadratic, linear or min-max criterion. The controllers from the library were successfully used in both simulations and the real-time control of laboratory models. The library is available on web pages of Tomas Bata University in Zlin (Chalupa, 2008).

ACKNOWLEDGEMENT

This work was supported in part by the Ministry of Education of the Czech Republic under grant MSM 1M0567, and in part by the Grant Agency of the Czech Republic under grant No. 102/06/P286.

REFERENCES

Bobál, V, J. Böhm, J. Fessl and J. Macháček (2005). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*. Springer-Verlag. London.

Chalupa, P. and V. Bobál (2007). Adaptive Predictive Control of Interconnected Liquid Tanks. In: *Proceedings of The International Symposium on System Theory, Automation, Robotics, Computers, Informatics, Electronics and Instrumentation*, Craiova, Romania, October 18-20, 2007, pp. 72-77.

Chalupa, P. (2008). STuMPCoL *Self-Tuning Model Predictive Controllers Library* [on line]. http://www.fai.utb.cz/people/chalupa/STuMPCoL

Hang, C.C., H.T. Lee and W.K. Ho (1993). *Adaptive Control*, Instrument Society of America.

Kulhavý, R. (1985). *Kandidátská disertační práce*, Institute of Information Theory and Automation, Czechoslovak Academy of Science. (in Czech).

Kwon, W. H. and S. HAN (2005). *Receding Horizon Control*. Springer-Verlag. London.

Ljung, L. (2001). *System Identification Toolbox for Use with MATLAB.* The MathWorks, Inc.

Sunan, H., T.K. Kiong and L.T. HENG (2002). *Applied Predictive Control*. Springer-Verlag. London.