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

PROCEEDINGS

17th 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

Sekaj, I.: Evolution of Continuous-Time Controllers with Various Criterion Functions, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 17th International Conference on Process Control '09*, Štrbské Pleso, Slovakia, 145–151, 2009.

Full paper online: http://www.kirp.chtf.stuba.sk/pc09/data/abstracts/042.html

EVOLUTION OF CONTINUOUS -TIME CONTROLLERS WITH VARIOUS CRITERION FUNCTIONS

I. Sekaj

Institute of Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava
e-mail : ivan.sekaj@stuba.sk,

Abstract: A genetic algorithm based controller design approach is described. The genetic algorithm represents an optimisation procedure, where the cost function to be minimized comprises the closed-loop simulation and a performance index evaluation. Depending on performance index various control aims can be considered. In the paper, several performance indices are analysed in a single-criterial and multi-criterion case. All design methods are experimentally compared.

Keywords: controller design, continuous-time process, genetic algorithm, various performance indices, multiobjective optimisation

1 INTRODUCTION

Control design tasks have to respect several requirements imposed on the static and dynamic behaviour of the controlled system. Controllers often include many searched parameters and their different constraints. The search/optimisation process may be complicated, discontinuous or non-convex, and analytical methods often may not be able to yield satisfactory results. Opposite to this, evolution-based search approaches are able to construct new control laws and non-intuitive solutions as well. One of the most frequently used evolutionary techniques is the genetic algorithm (GA). Recently, genetic algorithms have been applied in the area of process control for solving a wide spectrum of various optimisation problems in several ways and with several aims. In this paper we want to focus on the design of controller parameters (or control algorithm parameters) for continuous system control. In a first group of methods GA's are used as a powerful optimisation or search means in analytically formulated control design methods. Based on mathematical models, the parameters of a controller (or any dynamic system) are designed using different approaches providing stability and/or the required behaviour of the controlled system (Kawabe 1996, Krohling 2001, Man 2001, Sekaj 2002). The second group of methods is characterized by the use of simulation-based time-response evaluation of the closed-loop model, whereby the time-response can be used for various purposes (Herrero 2002, Khatib 1999, Lewin 2005, Mitsukura 1999, Sekaj 2005, Sweriduk 1999, Yang 2005). A multiobjective approach, where the evaluation contains 7 different objectives including analytically formulated and time-response performance based objectives is in (Molina-Cristóbal 2005).

In cases, where the design task represents not only the search for values of a set of parameters of a predefined control structure but also the search for its internal structure, an extension to this approach is possible. For this purpose the Genetic programming (Banzhaf 1999, Koza 1992, Sekaj 2005, Lewin 2005, Yang 2005) is applicable. This problem, however, is beyond the scope of this paper. For controller structure optimisation also a hierarchical GA has been used in (Man 2001). In (Grosman 2005) the genetic programming for Lyapunov function generating has been used. Surveys of evolutionary-based control system design are in (Fleming) and (Lewin 2005).

In this paper a straightforward way of incorporating simulation-based closed-loop time response in the GA is presented. The proposed approach deals with a direct GA-based search/optimisation in the controller parameter space combined with extensive computer simulations of the designed system (Sekaj 1999, Sekaj 2005b). Thus the simulation is an essential part of the minimised objective function. It will be shown that due to this approach, the task of the optimal design of the dynamic system parameters is transformed into a conventional n-dimensional optimisation problem. Various control aims can be obtained using various cost functions. For this purpose several performance indices are used and experimantaly compared. Finally a multicriterial design approach with more design aims is demonstrated.

2 CONTROLLER DESIGN PRINCIPLE

As mentioned above, the aim of the control design is to provide required static and dynamic behaviour of the controlled process. Usually, this behaviour is represented in terms of the well-known concepts referred in the literature: overshoot, settling time, decay rate, steady state error or various integral performance indices (Dorf 1990, Kuo 1991) etc.

Without loss of generality let us consider a simple feedback control loop (closed-loop) (Figure1) where y is the controlled value, u is the control value, r is the reference and e is the control error ($e=r-y$). Consider an appropriate closed-loop simulation model is available. Let us analyse the closed-loop behaviour using the simple integral performance index "Integral of absolute error", which is defined as

$$I_{AE} = \int_0^T |e(t)| dt \qquad (1)$$

where T is the simulation time. The controller design principle is actually an optimisation task - search for such controller parameters from the defined parameter space, which minimise the performance index (1). The cost function (fitness) is a mapping $R^n \rightarrow R$, where n is the number of designed controller parameters. The evaluation of the cost function consists of two steps. The first step is the computer simulation of the closed-loop time-response, and the second one is the performance index evaluation. In case of designing complex, multi-input and multi-output (MIMO) control structures or other controller types (fuzzy controllers, neuro-controllers, etc.) the dimension n of the search space can be large (more than tens or even hundreds), therefore the cost function can be complicated and multi-modal and due to high computational requirements the use of "conventional" optimisation methods may be not feasible. Here, the evolution-based techniques, e.g. genetic algorithms can be used.

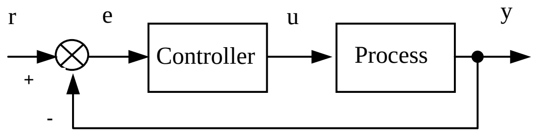

Figure 1 Simple feedback control loop

Genetic algorithms are described in e.g. (Goldberg 1989, Michalewicz 1996, Eiben 2003, Man 2001) and others. A general scheme of a GA can be described by following steps:

1. Initialisation of the population

2. Evaluation of the cost function for entire population.

3. Selection of parent chromosomes.

4. Crossover and mutation of the parents → children.

5. Completion of the new population from the new children and selected members of the old population. Jump to the step 2.

The chromosomes are linear strings, whose items (genes) represent in our case the designed controller parameters. Because the controller parameters are real-number variables and in case of complex problems the number of the searched parameters can be large, GA's with real-coded chromosomes have been used.

Without loss of generality let us consider of a simple PID controller, described in the time domain by the equation

$$u(t) = Pe(t) + I\int e(t)dt + D\frac{de(t)}{dt} \qquad (2)$$

where $P \in R$, $I \in R$, $D \in R$ are the proportional, integral and derivative gains respectively. The chromosome representation in this case can be in form ***ch***$=\{P,I,D\}$. Note, that for an other controller type with the parameters $c_1, c_2, \ldots, c_q$ the appropriate chromosome is a linear string

ch$=\{c_1, c_2, \ldots, c_q\}$. Before each simulation, the corresponding chromosome (genotype) is decoded into controller parameters of the simulation model (phenotype) and after the simulation the performance index is evaluated.

In Figure 2 a PID controller evolution using (1) is demonstrated, where after some generations the best solution from the current population (its closed loop step response) is plotted. As after 100 generations the solution doesn't change considerably, the GA run can be terminated. In Figure 3 the cost function (1) convergence during three independent GA-runs (cost function value vs. generation number) is depicted. Note, that in case of complex system designs the controller design based on dynamic process simulations can be a multi-modal and a time-consuming problem, where often a good sub-optimal solution can be sufficient. The question about the GA-design procedure convergence is similar to other numerical GA-based search/optimisation problems (Goldberg 1989, Michalewicz 1996). The convergence rate depends on the search space size and dimension, on the GA structure and on the used genetic operations.

3 CHOICE OF THE EVOLUTION CRITERION

Consider that the GA has found the optimal (sub-optimal) solution for a defined performance index in the user-defined search space of controller parameters. The choice of the performance index has a fundamental influence on the closed-loop dynamics. Normally, using (1) or (2) brings about fast control responses with small overshoots between 2-5% (Figure 4). For various objectives different performance indices can be used (Sekaj 2003, Sekaj 2005b). If it is necessary to reduce overshoot or to damp oscillations, it is recommended to insert in the integral additional terms, which include absolute values of the first order or also the second order derivatives of the control error

$$J = \int_0^T \left(\alpha|e(t)| + \beta|e'(t)| + \gamma|e''(t)| \right) dt \qquad (3)$$

and to increase β and γ with respect to α, where α, β, γ are weight coefficients. Note, that the control error derivatives can be replaced with the absolute values of output (or output derivatives |y'(t)|, |y''(t)|).

In the discrete-time case the integral is replaced by the sum and the derivatives by the differences. Good results can be obtained also using the performance index as follows

$$J = \alpha\eta + (1-\alpha)t_s \qquad (4)$$

where η is the overshoot, t_s is the settling time and $0 < \alpha < 1$ is the weight coefficient. Tracking the reference variable $y_r(t)$ is achieved via minimizing

$$J = \int (y_r(t) - y(t))^2 dt$$

Control energy minimization can be achieved using performance indices of the type

$$J = \int (\alpha e^2(t) + (1-\alpha)u^2(t))dt$$

where u is the control variable. Closed-loop step responses for a particular closed-loop under a PID are in Figure 4. The following criterions have been used: a) IAE (1); b) criterion (3) with $\alpha = 3, \beta = 2, \gamma = 0$; c) criterion (4) with $\alpha = 0.8$; d) criterion (4) with $\alpha = 0.2$.

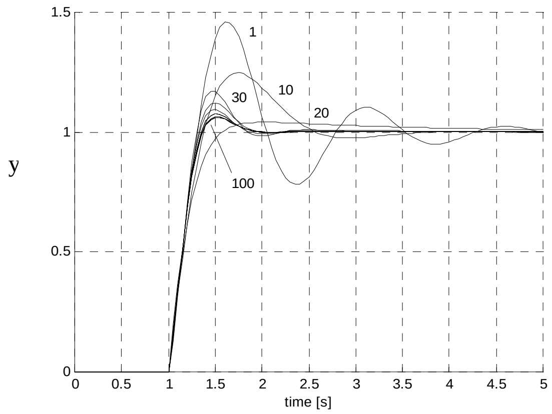

Figure 2 Evolution of the PID controller parameters after 1, 10, 20, 30 and 100 generations

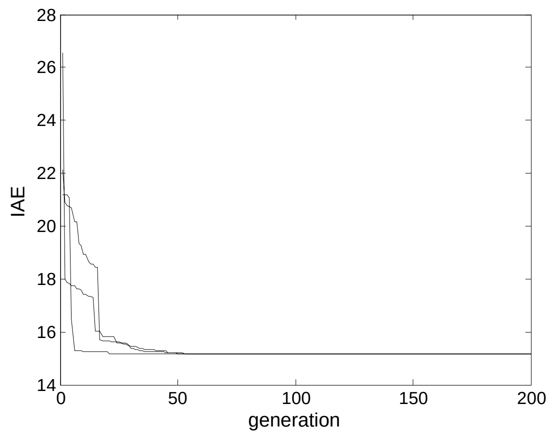

Figure 3 Cost function convergence of the PID controller design procedure for three independent GA-runs

An universal performance index, which combines some of the above criterions is as follows

$$J = \int_0^T \left(|e(t)| + \alpha|e'(t)| + \beta|u(t)| + \gamma|u'(t)| \right) dt \qquad (5)$$

where e' is the control error derivative, u is the control variable and u' is its derivative. This performance index includes oscillation damping (increasing α), minimization of the absolute value of the control signal u (increasing β) and minimization of control signal change u' (increasing γ). In Figure 5 the step responses with various weight constants of the criterion (5) are depicted. The controlled system is

$$P(s) = \frac{1}{4s^5 + 8s^4 + 10s^3 + 6s^2 + 3s + 1}.$$

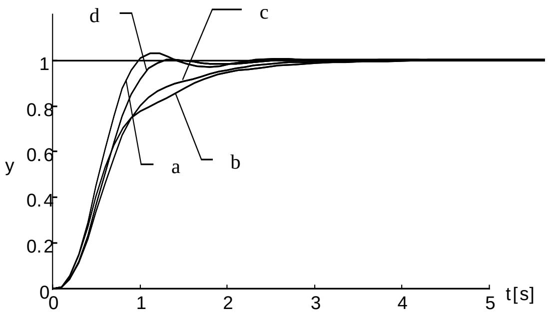

Figure 4 Closed-loop step responses using various performance indices

The weight constants with the obtained PID coeficients for 5 various cases are in Table 1.

Table 1 Evolution results of the PID controller design

	α	β	γ	P	I	D
1	*0*	*0*	*0*	*26.31*	*8.94*	*30.26*
2	*1*	*0*	*0*	*8.70*	*3.15*	*6.59*
3	*10*	*0*	*0*	*6.45*	*2.19*	*4.41*
4	*1*	*0*	*1*	*1.56*	*0.72*	*0*
5	*1*	*0*	*0.5*	*1.56*	*0.71*	*0*

Remark about the stability: Due to the applied performance index minimisation, the closed-loop stability is an implicit attribute of each solution. During the evolution, unstable chromosomes are eliminated because of their high value of performance index and the solution is directed into a stable parameter region. However, if necessary, it is possible to include a stability test into each fitness evaluation. Unstable individuals can additionally obtain high penalty values.

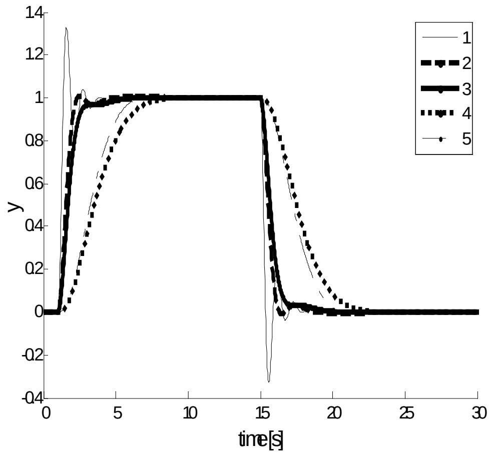

Figure 5 Time responses obtained with various criterion function settings (according Tab.1)

4 MULTIOBJECTIVE CONTROLLER DESIGN

In solving many practical design problems not just a single optimisation objective is considered. Moreover the particular objectives are in contradiction (e.g. performance / energy consumption, etc.). A common way of solving multi-objective tasks is using a single cost function consisting of multiple parts

$$J = w_1 f_1 + w_2 f_2 + \ldots + w_n f_n$$

where each part f_i, $i = 1, \ldots, n$ represents an objective with some weight w_i (as (3),(4),(5) in Section 3). The main disadvantage of this method is the high sensitivity of the solution to the weight coefficients. This can lead to solutions, which do not correspond to our primary requirements.

Another way for solving multi-objective problems is the use of the Dominance principle, which is the search for the Pareto-optimal set of solutions. This is an effective way to overcome the above mentioned problem. Consider a minimisation problem. According to the Dominance principle the individual *x* dominates the individual *y* (or the individual *y* is dominated by the individual *x*) if

$$\forall i = 1,2,\ldots,n \;\; ; \quad f_i(x) \le f_i(y) \quad \text{and}$$
$$\exists j = 1,2,\ldots,n \;\; ; \quad f_j(x) < f_j(y),$$

where n is the number of objectives and f_i, $i=1,2,\ldots,n$ is cost function corresponding to the i-th objective. In case of maximization tasks the formulation is analogical. A set of individuals, which are non-dominated by another individual are members of the Pareto-optimal set of solutions.

Now the objective is not to find a single solution, but as much as possible non-dominated solutions. For these it is not possible to decide, which one is better. Each user will select the individual, which is the best with respect to his requirements. Based on the above approach the search algorithm (GA) is as follows:

1. Generating the initial population.

2. Calculation of all objective functions for each individual of the population.

3. Domination calculation: each individual of the population will obtain such a number of "penalty points" that corresponds to the number of individuals by which it is dominated. The number of penalty points represents the final minimized cost function.

4. Individuals with zero penalty points are stored in the current group of non-dominated individuals.

5. Calculation of the new population (selection, crossover, mutation).

6. Adding the current non-dominated group into the new population.

7. Testing of terminating conditions, jump to Step 2 or end.

The following terminating conditions can be used: satisfaction of the required cost function values (if known), performing of the predefined number of generations or obtaining a predefined number of non-dominated solutions.

In case of controller design applications various objectives can be considered: integral performance indices (Section III), settling time, maximum overshoot, oscillation damping, various stability measures, gain/phase margin, energy consumption, minimization of negative environmental impacts of the controlled process operation, etc.

Consider the design of a DC motor speed controller. The objectives are good quality of transient processes, i.e. over-damped closed loop system with short settling time and on the other hand low control energy consumption. To fulfil the first objective let us minimise the simple integral criterion (1)

$$J_1 = \int_0^T |e(t)| dt$$

The second objective can be represented by minimisation of the integral of the input voltage square, which is proportional to the input energy consumption

$$J_2 = \int_0^T u(t)^2 \, dt$$

The results obtained using the algorithm described in the Section 4 are depicted in Figure 6 (J_2 versus J_1). All individuals, which have occurred during the GA run are marked by "x". The non-dominated solutions from the last generation are in the left bottom part of the area marked with "o". Detail of the non-dominated set is depicted in Figure 7. The individual marked *"Min(J1)"* represents the solution with the best performance with respect to the objective J_1 and the individual marked *"Min(J2)"* represents the best solution with respect to the objective J_2. The individual marked *"Compromise"* is a selected trade off between the both previous extremes. Step responses for all three selected solutions are depicted in Figure 8 and Figure 9.

5 CONCLUSION

The presented GA-based controller design approach is minimising such a cost function, which comprises system simulation and performance index evaluation. In this way the controller design is transformed into a search problem in a n-dimensional parameter space. The subjects of design/optimisation may be complex systems with control structures of various types. The main (and practically the only) limitation of this approach is the computational time, which is higher in comparison to conventional approaches. In the presented multiobjective approach it is possible to consider more than one objective. The proposed GA-based approach has been in our department successfully applied for the design of various types of controllers for various system types (linear, non-linear, stable, unstable, non-minimum phase, SISO and MIMO) in simulation as well as in real-time applications. The design approach is powerful, robust, widely applicable and simple to use.

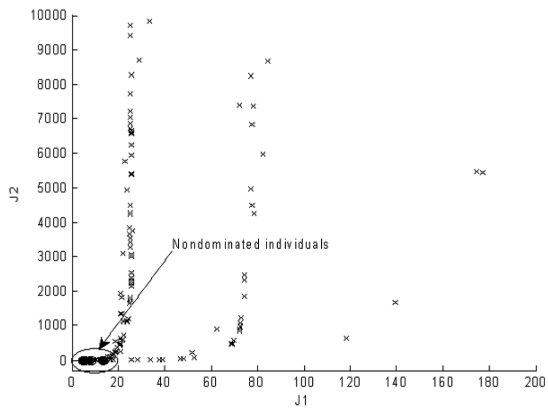

Figure 6 Individuals occurred during the solution

x - all individuals, o - non-dominated individuals

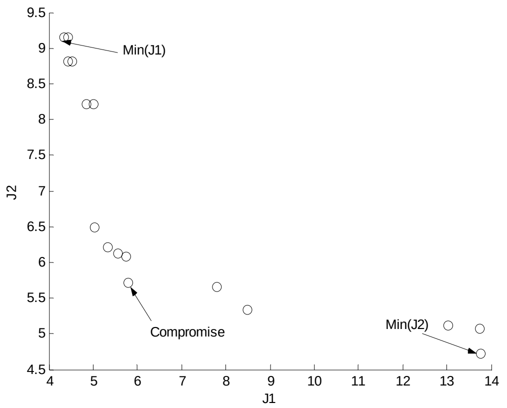

Figure 7 Set of non-dominated individuals - detail (Pareto-optimal set)

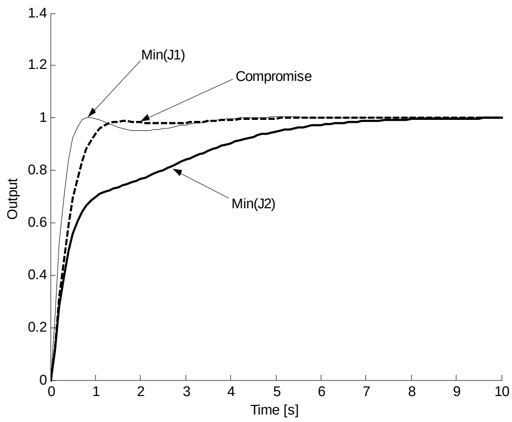

Figure 8 Time-responses of three selected individuals

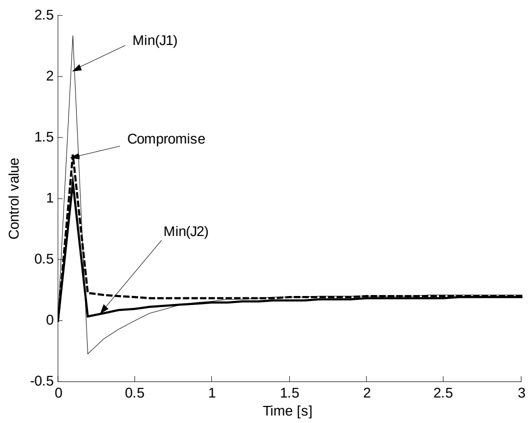

Figure 9 Control variable time-responses of the three selected individuals

ACKNOWLEDGMENT

The work has been supported by the grants VEGA no. 1/0544/09 and no. 1/0690/09.

REFERENCES

Banzhaf, W., P. Nordin, R. E. Keller, F. D. Francone (1999). *Genetic Programming: An introduction.* San Francisco: Morgan Kaufmann.

Dorf, R. C. (1990). *Modern Control Systems.* Addison-Wesley publishing Company, 5th edition.

Eiben, A. E. and J. E. Smith (2003). *Introduction to Evolutionary Computing*. Springer.

Fleming, P. J. and R. C. Purshouse. Evolutionary Algorithms in Control System Engineering: A Survey. *Control Engineering Practice*, 10(11), 1223

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addisson-Wesley.

Grosman, B. and D. R. Lewin (2005). Automatic Generation of Lyapunow Functions Using Genetic Programming, *16th World Congress of IFAC*, Prague, July 3-8.

Herrero, J. M., X. Blasco, M. Martínez, J. V. Salcedo (2002). Optimal PID Tuning with Genetic Algorithms for Non Linear Process Models. *Proceedings on the 15th World Congress of IFAC,* Barcelona, July 21-26.

Kawabe, T., T. Tagami, T. Katayama (1996). A Genetic Algorithm based Minimax Optimal Design of Robust I-PD Controller. *UKACC Int. Conference on Control '96,* Conf. Publication No. 427, IEE, pp.436-441.

Khatib, W., V. Silva, A. Chipperfield, P. Fleming (1999). Multidisciplinary Optimisation with Evolutionary Computing for Control Design. *Proceedings on the 14th World Congress of IFAC,* Beijing, July 5-9.

Koza, J. R. (1992). *GeneticProgramming*. Cambridge, MA: MIT Press.

Krohling, R. A. and J. P. Rey (2001). Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms, *IEEE Trans. On Evolutionary Computation,* Vol.5, No.1.

Kuo, B. C. (1991). *Automatic Control Systems.* Prentice-Hall International Editions.

Lewin, D. R. (2005). Evolutionary Algorithms in Control System Engineering. *16th World Congress of IFAC*, July 3-8, Prague, Czech Rep.

Man, K. F., K. S. Tang, S. Kwong (2001). *Genetic Algorithms, Concepts and Deign.* Springer.

Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolutionary Programs.* Springer.

Mitsukura, Y., T. Yamamoto, M. Kaneda, K. Fujii (1999). Evolutionary Computation in Designing a PID Control System. *Proceedings on the 14th IFAC World Congress,* Beijing, 5-9 july, P.R.China, pp. 497-502.

Molina-Cristóbal, A., I. A. Griffin, P. J. Fleming, D. H. Owens (2005). Multiobjective Controller Design: Optimising Controller Structure with GA. *16th World Congress of IFAC*, July 3-8, Prague.

Sekaj, I. (1999). Genetic Algorithm-based Control System Design and System Identification. *Proceedings on the Int. Conference Mendel'99*, June 9-12, , Brno, Czech Republic, pp.139-144.

Sekaj, I., M. Foltin, M. Gonos (2002a). Genetic Algorithm Based Adaptive Control of an Electromechanical MIMO System. *Proceedings of the GECCO Conference*, July 9-13, New York, pp. 696

Sekaj, I. and V. Veselý (2002b). Robust Output Feedback Controller Design Via Genetic Algorithms. *Proceedings on the 15th World Congress of IFAC*, Barcelona, 21-26 July.

Sekaj, I. (2003). Genetic Algorithm Based Controller Design. *2nd IFAC conference Control System Design'03*, Bratislava, Slovak Republic, September 7-10.

Sekaj, I., J. Perkácz, M. Nídel (2005a). Controller design based on genetic programming. *Proceedings of the Int. conference Mendel'05*, Brno, Czech Republic, June 15-17.

Sekaj, I. and M. Šrámek (2005b). Robust Controller Design Based on Genetic Algorithms and System Simulation. *Proceedings on the 44th IEEE CDC and ECC conference 2005*, Seville, Spain, 12-15 December

Sweriduk, G. D., P. K. Menon, M. L. Steinberg (1999). Design of a pilot-activated recovery system using genetic search methods. *Optimal Synthesis.*

Yang, Z. Y., C. W. Chan, M. S. Xue, G. J. Luo (2005). On-line Temperature Control of an Oven Based on Genetic Algorithms. *16th World Congress of IFAC, Prague, July 3-8.*