**Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005**

Robust Controller Design Based on Genetic Algorithms and System Simulation

Ivan Sekaj and Marián Šrámek

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. Based on this, simple design and two robust controller design methods have been proposed. All design methods are experimentally compared. Additionally, for qualitative evaluation and comparison of different controllers a proposed statistical robustness measure is presented.**

I. 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 [8],[11],[14],[21]. 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 [7],[9],[13],[16],[24],[25]. A multiobjective approach, where the evaluation contains 7 different objectives including analytically formulated and time-response performance based objectives is in [17].

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. This problem, however, is beyond the scope of this paper. For this purpose the Genetic programming [10],[1],[23], [13],[25] is applicable. For controller structure optimisation also a hierarchical GA has been used in [14]. In [6] the genetic programming for Lyapunov function generating has been used. Surveys of evolutionary-based control system design are in [4] and [13].

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 [19],[22]. 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.

II. CONTROLLER DESIGN

A. 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: maximum overshoot, settling time, decay rate, steady state error or various integral performance indices [2],[12], etc.

Without loss of generality let us consider a simple feedback control loop (closed-loop) (Fig.1) 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" defined as

Manuscript received February 18, 2005. This work has been supported from the grants of the Slovak grant agency VEGA 1/0155/03 and 1/0158/03.

Ivan Sekaj, Department of Automatic Control Systems, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic, Phone: ++421 2 602 91 585, E-mail: ivan.sekaj@stuba.sk

Marián Šrámek, Department of Automatic Control Systems, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic, E-mail: msramek@nextra.sk

0-7803-9568-9/05/$20.00 ©2005 IEEE

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

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.

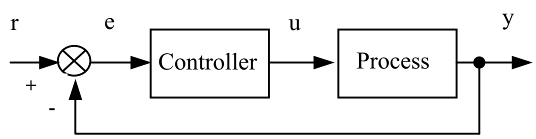

Fig. 1 Simple feedback control loop

Genetic algorithms are described in e.g. [5],[15],[3],[14] and others. A general scheme of a GA can be described by following steps:

1. Initialisation of the population of chromosomes (set of random potential solutions).
2. Evaluation of the cost function for all chromosomes.
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. A block scheme of a GA-based design is in Fig.2.

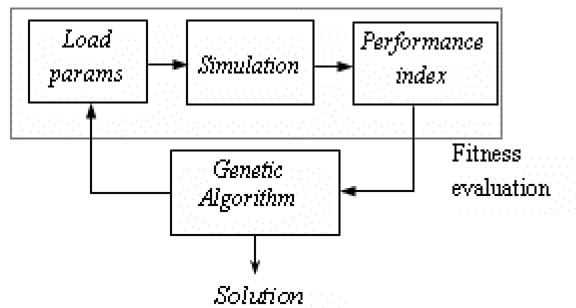

Fig. 2 Block scheme of the GA-based controller design

In Fig.3 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 Fig.4 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 [5],[15]. The convergence rate depends on the search space size and dimension, on the GA structure and on the used genetic operations.

B. *Choice of the performance index*

Consider that the GA finds the optimal (sub-optimal) solution in the user-defined search space of controller parameters. The choice of the performance index has a fundamental influence on the closed-loop behaviour. Using (1) normally brings about fast control responses with small overshoots between 2-5% (Fig.3). Different performance indices are described in [19],[22].

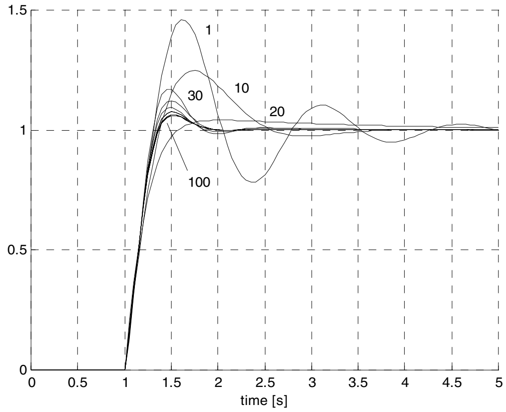

Fig. 3 Evolution of the PID controller parameters after 1, 10, 20, 30 and 100 generations

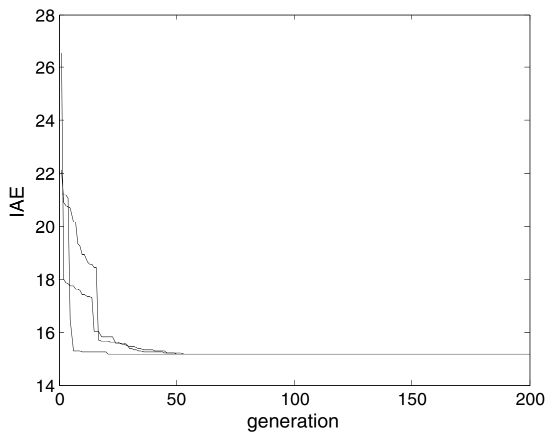

Fig. 4 Cost function convergence during the PID controller design procedure. Three independent GA-runs.

An universal performance index is as follows

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

where e' is the control error derivative, u is the control value and u' is its derivative. It takes into account control performance optimisation with oscillation damping (increase of α), minimisation of the absolute control signal u (increase of β) and minimisation of control signal change u' (increase of γ).

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, it is possible to include a stability test into each fitness evaluation. Unstable individuals can additionally obtain high penalty values.

III. Robust Controller design

Let us describe two methods, extending the approach from Section 2, which can increase the closed-loop robustness.

A. Set of defined working points

Consider $\boldsymbol{c}=\{c_1, c_2, \dots, c_q\}$ to be the set of the designed controller parameters and $\boldsymbol{s}=\{s_1, s_2, \dots, s_r\}$ the set of parameters of the controlled object. During the operation, each of the parameters s_i can move within the uncertainty space defined by the intervals

$$S: \quad s_{i,\min} \le s_i \le s_{i,\max} \quad ; \quad i = 1,2,\dots,r \tag{3}$$

where $s_{i,min}$ and $s_{i,max}$ are the minimum and maximum possible values of the i-th system parameter. Consider n different (physical) working points of the controlled process, defined by different vectors $\boldsymbol{s}$, which are to be controlled by the same robust controller. For this case let us use the cost function

$$J = \sum_{i=1}^{n} J_i \tag{4}$$

which comprises evaluation in all n working points J_i. It is also recommended to include the measured noise from the real system or other possible disturbances or expected situations into the simulation model.

Remark: Alternatively to the set of n defined physical working points the set of 2^r system parameter vectors located in the vertices of a polytope, which represents the bounds of the parameter space S, can be considered.

B. Randomly generated working points

A next proposed method is as follows. In each GA-generation n random working points (for all chromosomes of the actual population the same) are generated (say $n=100$). That means n vectors $\boldsymbol{s}$ of the system parameters become random values from the space S. The fitness function for each individual of the population is calculated using the performance index (4). In the next generation other n random parameter vectors are generated. The algorithm for cost function evaluation in each generation of the GA is as follows:

1. generate n system parameter vectors $\boldsymbol{s}$ (by random)
2. for each individual of the current population perform a simulation of the closed-loop for each of the n parameter vectors $\boldsymbol{s}$
3. for each individual calculate the cost function (4)

IV. STATISTICAL ROBUSTNESS MEASURE

A similar simulation-based principle described above can be used also for another purpose – for controller robustness evaluation. The proposed Statistical robustness measure (SRM) is based on statistical evaluation of a set of (more than 1000) closed-loop simulation experiments with randomly generated system parameters $\mathbf{s}$ from the uncertainty space S. After evaluating the selected performance index for each experiment (in our case (1)), we obtain a set of points as depicted in Fig.5 in Section 5 (the performance index value vs. the experiment number). Smaller values of the performance index represent a better closed-loop behaviour. A more transparent evaluation of this experiment represents the use of the probability density function (Fig.6) (the probability vs. the performance index). Another possibility is the use of the probability distribution function, which arises by summing (or integrating) the density function (Fig.7). The control performance is better for such controllers, for which the distribution function (the density function) is located to the left within the horizontal axis range. Poor (or unstable) controllers achieve large (or extremely large) values of the performance index and therefore their distribution functions are located in the right hand side part of the horizontal axis range.

If necessary, the SRM can be quantified as a scalar value applying the sum

$$SRM = \sum_{i=1}^{N} J_i$$

where N is the number of closed-loop simulation experiments, and J is the evaluated performance index.

V. COMPARATIVE STUDY

For demonstration of the proposed controller design methods let us consider the controlled object described by the nonlinear differential equation

$$y'' + a_2 y' + a_1 y + a_0 y^3 = b_0 u \tag{5}$$

The gain of this system is non-constant and depends on the position of the working point y. Let the system parameters are moving within the uncertainty intervals

$$\begin{aligned} &b_0 \in \langle 1;5 \rangle, a_0 \in \langle 0.02;5 \rangle, \\ &a_1 \in \langle 0.02;0.2 \rangle, a_2 \in \langle 0.01;20 \rangle \end{aligned} \tag{6}$$

Consider a PID controller described by the transfer function

$$G_{PID}(s)=P+I/s+Ds$$

Let the control value be limited in the range $-10 \le u \le 10$. For comparison, the controller parameters P,I and D have been designed applying following methods:

1. Robust approach from Section III.B with 100 randomly generated working points from the space S in each generation.
2. Using 2^4=16 working points located in the vertices of the parameter space S (see remark at end of the Section III.A).
3. Robust approach according Section III.A with 3 fix working points. The first working point represents parameters, which values correspond to the lower interval limits of (6) *(WP_1: a_0=0.02; a_1=0.02; a_2=0.01; b_0=1)*, the second to the upper interval limits of (6) *(WP_2: a_0=5; a_1=0.2; a_2=20; b_0=5)* and the third represents the mean values of intervals (6) *(WP_3: a_0=2.51; a_1=0.11; a_2=10.005; b_0=3)*.
4. Simple PID design using GA (Section II, eq. (1)) in the nominal working point WP_3, which parameters are the mean values of (6) *(a_0=2.51; a_1=0.11; a_2=10.005; b_0=3)*.
5. Conventional Optimal Module PID design method (Oldenburg and Sartorius, 1956) using a linearized system in the nominal working point WP_3.

The design results are in Table 1.

TABLE 1
RESULTS OF DIFFERENT PID DESIGN METHODS

Method	P	I	D	SRM
1	99.23	1.68	4.72	2395
2	99.37	1.01	12.52	2532
3	46.27	6.72	11.73	3185
4	98.70	8.66	8.45	2662
5	20.56	0.98	0.36	4774

The SRM for these results has been evaluated from 1200 experiments. The obtained values of performance indices for the method 1 are in Fig. 5. The density functions for all methods are in Fig.6 and the distribution functions in Fig.7. Finally, all methods have been tested on 25 randomly generated test systems from the parameter space S. For illustration, the setpoint step-responses (to 1 in t=0s and to 0.5 in t=20s) for two design methods 5 and 1 are compared in Fig.8 and Fig.9.

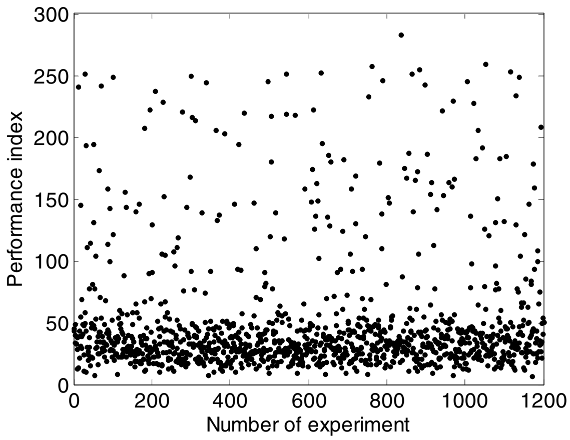

Fig. 5 Performance index values from 1200 experiments for the Random generated working points method (method No.1)

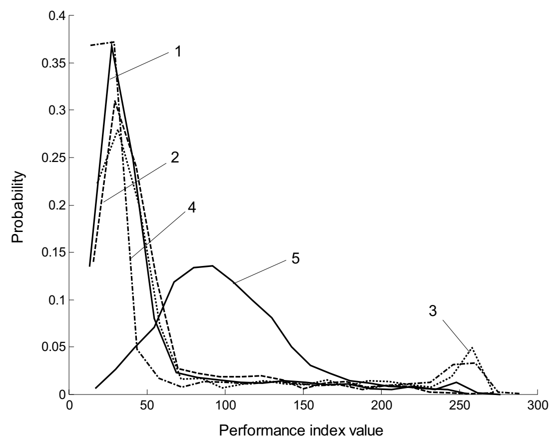

Fig. 6 Probability density function for all 5 methods

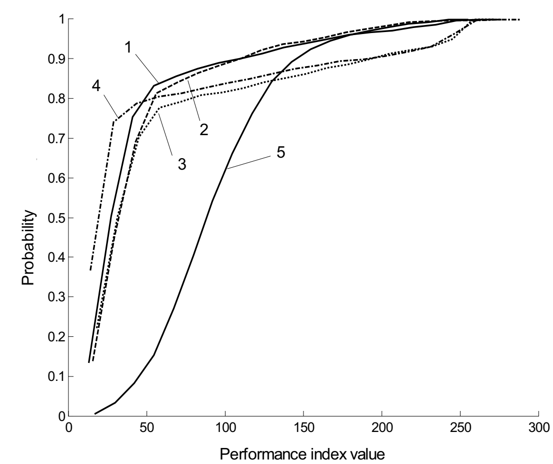

Fig. 7 Probability distribution function for all 5 methods

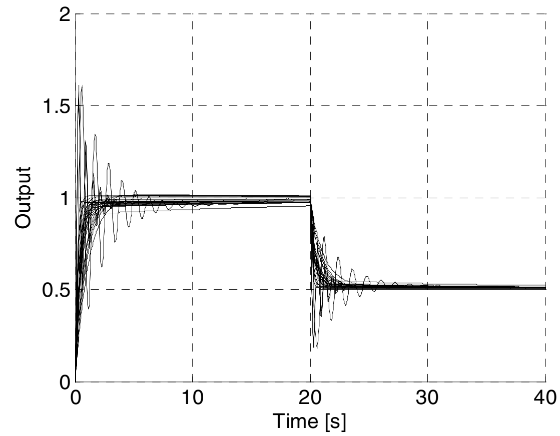

Fig. 8 Closed-loop responses of 25 test systems for the Optimal module PID design method (method No.5)

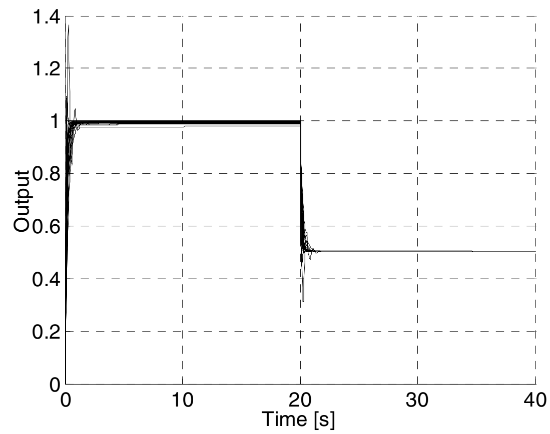

Fig. 9 Closed-loop time responses of 25 test systems for the GA design method with random generated working points (method No.1)

VI. 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. The design approach is simply extendable to robust controller design, for which two different methods have been proposed. Additionally, for robustness measure evaluation a statistical robustness measure method is proposed, which can be considered as an objective tool for robustness comparison of different controller types or different controller parameters.

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, in laboratory and also in the practice. The design approach is powerful, robust, widely applicable and really simple to use.

REFERENCES

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

[2] R. C. Dorf, *Modern Control Systems.* Addison-Wesley publishing Company, 5th edition, 1990

[3] A. E. Eiben, and J. E. Smith, *Introduction to Evolutionary Computing*. Springer, 2003

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

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addisson-Wesley, 1989

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

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

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

[9] W. Khatib, V. Silva, A. Chipperfield, P. Fleming, "Multidisciplinary Optimisation With Evolutionary Computing for Control Design", In: *Proceedings on the 14th World Congress of IFAC,* Beijing, July 5-9, 1999

[10] J. R. Koza, *GeneticProgramming*. Cambridge, MA: MIT Press, 1992

[11] R. A. Krohling, and J. P. Rey, "Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms", In: *IEEE Trans. On Evolutionary Computation,* Vol.5, No.1, 2001

[12] B. C. Kuo, *Automatic Control Systems.* Prentice-Hall International Editions, 1991

[13] D. R. Lewin, "Evolutionary Algorithms in Control System Engineering", In *16th World Congress of IFAC*, July 3-8, Prague

[14] K. F. Man, K. S. Tang, S. Kwong , *Genetic Algorithms, Concepts and Deign.* Springer, 2001

[15] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolutionary Programs.* Springer, 1996

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

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

[18] L. C. Oldenburg, H. Sartorius, "A uniform approach to the optimum adjustments of control loops", *Frequency response*. The MacMillan Co., N.Y., 1956

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

[20] I. Sekaj, M. Foltin, M. Gonos, "Genetic Algorithm Based Adaptive Control of an Electromechanical MIMO System", In: *Proceedings of the GECCO Conference*, July 9-13, 2002, New York, pp. 696

[21] I. Sekaj, and V. Veselý, "Robust Output Feedback Controller Design Via Genetic Algorithms", In: *Proceedings on the 15th World Congress of IFAC*, Barcelona, 21-26 July, 2002.

[22] I. Sekaj, "Genetic Algorithm Based Controller Design", In: *2nd IFAC conference Control System Design'03*, Bratislava, Slovak Republic, September 7-10, 2003

[23] I. Sekaj, J. Perkácz, M. Nídel, "Controller design based on genetic programming", In: *Proceedings of the Int. conference Mendel'05*, Brno, Czech Republic, June 15-17, 2005

[24] G. D. Sweriduk, P. K. Menon, M. L. Steinberg, "Design of a pilot-activated recovery system using genetic search methods", In: *Optimal Synthesis,* 1999

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