

GP and GA in the Design of a Constrained Control System with Disturbance Rejection

Andrea Soltoggio

Department of Computer and Information Science
Norwegian University of Science and Technology
N-7491, Trondheim, Norway
soltoggi@stud.ntnu.no

Abstract—The design of a robust controller for a constrained SISO¹ linear system is considered. Initially, the study of a solution provided by genetic programming (GP) outlined that the GP search process does not achieve time-optimality.

A genetic algorithm (GA) was chosen and implemented to confront the performance of the GP solution. The system presented in this paper tuned a set of elements that form the controller structure, namely, a PID core, a feed-forward block, a filter on the derivative and a Butterworth filter on the feedback. The proposed approach proved to reach time-optimality thanks to the achievement of bang-bang control. The control synthesis method also showed versatility when considering load and feedback disturbances on the system. GA appears to be preferable for the computational cost and the quality of the solution obtained.

I. INTRODUCTION

With the increase of computational power, evolutionary algorithms (EAs) in control engineering are used to tackle complex problems more effectively than in the past. A recent survey on the present EAs techniques applied to control engineering is provided in [6].

In the EAs panorama, GAs are often regarded as tuning tools performing well on fixed controller structures. Hence, as a general rule, GAs are mostly used as alternative optimisation techniques where the controller and the search space are well defined.

Optimisation of parameters by means of GAs was originally proposed in [8], [32]. In [32], the genotype² was composed by the three PID parameters. The method was tested for optimisation of nonlinear processes and showed robustness and efficiency. Later, several other applications of GAs appeared in [9], [17]–[19], [31]. In most of the previously cited cases, however, the search was limited to the three PID parameters. A more complex optimization is proposed in [20], [21] where adaptive control is reached by means of genetic algorithms.

To gain more flexibility, genetic programming (GP) has been used [5], [7], [15], [16]. In [7], a method for model structure identification has been implemented. With the aim of making the controller structure and the parameter tuning both targets of the search process, in [15], a GP approach is

described. No assumptions on the controller structure were given and the method was free to evolve the most suitable architecture for the current control problem. The plant used as benchmark was a second order linear system. An optimal controller for this plant is proposed in a control engineer textbook [4].

In this paper, the controller proposed in [15], [16] was reproduced and verified with respect to performance. It was found that the GP controller is several times faster than the time-optimal PID proposed in [4] because of a different constraints specification. The proposed comparison [15] of the GP controller with a traditional PID is therefore not relevant.

A GA application was designed to tune a controller with the same constraints used for the GP search process. The application, by evolving a 10-dimensional genotype, is able to tune and, to a certain degree, to shape a controller structure to minimize a time domain index. The solution found improved the performance of the GP controller. Parts of the controller are set to filter noise applied to the system. The filtering is automatically increased or decreased depending on the level of noise. Contrary to the method proposed in [17], which limits itself to tune three PID parameters, in this experiment, disturbance rejection is achieved by the automatic decision of enabling or disabling the filters shaped by 7 additional parameters.

II. THE CONTROL PROBLEM

The plant to be controlled, presented in [4], is expressed by the transfer function

$$G(s) = \frac{K}{(\tau s + 1)^2}, \quad (1)$$

where K and τ vary between the values $1 \leq K \leq 2$ and $0.5 \leq \tau \leq 1$ to obtain robust control. The transfer function represents the mathematical model of a set of processes widely spread in control engineering. Detailed descriptions of physical models are present in control textbooks [2], [4], [29]. In this paper, the mathematical model is used without addressing a particular physical system.

The simulation of the controlled system is carried out for the four states corresponding to the four combinations of the values $K = 1, 2$ and $\tau = 0.5, 1$.

¹Single Input, Single Output

²In GAs the genotype is represented by the array of parameters to be optimized.

For the GP and GA methods, the control problem is constrained by the presence of an actuator with limitation of the control variable to ± 40 Volts. Overshoot within 2% is also required.

The control problem of equation (1) is followed, in [4, pages 697-700], by a traditional method of synthesis for a PID with prefilter. The control variable is unconstrained and the design is linear and time-optimal with respect to the integral of the time-weighted absolute error

$$ITAE = \int_{t_1}^{t_2} t \cdot e(t) dt \quad . \quad (2)$$

In the design, a parameter ω is used to set the intensity of the control variable and achieve a time-optimal response within certain ranges of $u(t)$, $\dot{u}(t)$ and overshoot.

III. THE GP APPROACH

The genetic programming approach has been used in [15], [16] to design from scratch a controller for the plant of equation (1). The genotype was a tree-coded controller mapped into a SPICE code for the simulation of the electric circuit. The fitness function was composed by a weighted sum of the ITAE index and additional penalties for undesired behaviours like overshoot. The controller found as solution is said to be 2.42 times better than the optimal PID with respect to the same ITAE index. The controller has been reproduced and simulated in this experiment. Matlab and Simulink have been used for implementation and simulation.

a) Simulation results: The simulation of the GP controller, compared to the standard PID, shows that the GP controller uses the control variable in a more intensive way than the PID. It makes use of saturated control and higher varying rate of the control variable. Figure 1 shows the control variable for both the controllers.

The use of nonlinear, saturated control helps the GP controller to achieve a faster response. Besides, the GP controller as presented in [15], [16], uses a second derivative that, without the introduction of a proper low-pass filter, produces a very high bandwidth. Moreover, the lack of a constraint for the derivative of the control variable and the unlimited bandwidth in the feedback loop ($Y(s)/Y_{feedback}(s)$) allows a potentially infinite load disturbance suppression but is not implementable on a real system. For the simulation, the derivative was implemented with an embedded first order low-pass filter in order to obtain a finite bandwidth.

From the analysis, the controller presented in [15], [16] has substantially different characteristics from the PID and is therefore not comparable. However, the design method explained in [4, pages 697-700] allows to tune the PID control variable to any desired value. Hence, in a second simulation, the standard PID controller was tuned for a stronger control action tentatively setting the tuning parameter ω from 8 to 16; additionally, a limit on the integral was imposed to 8 Volts and a gain of 3 was added to the

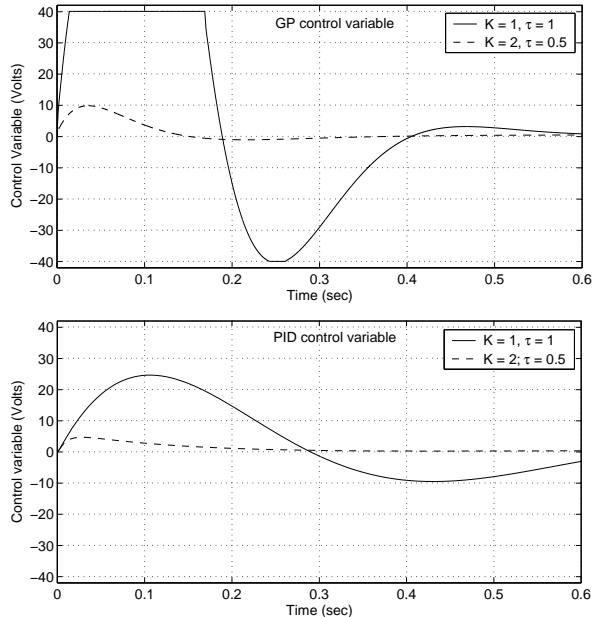


Fig. 1. Control variables for the GP and textbook PID controllers

feedback signal. The roughly tuned controller, compared to the GP controller, obtained better performance under all the considered indices. Table I shows the simulation results for $K=1$ and $\tau=0.5$.

It was observed from the simulations that both the GP controller and the new PID bring the control variable to saturation for the plant parameters $K=1$ and $\tau=1$; i.e. when the plant has the lowest gain (K) and the highest time constant (τ) and needs a stronger control action. For the other three combinations of the parameters, the system response does not change significantly and the control variable remains under the saturation limit: the controllers use saturated control only in one fourth of the cases. This observation gave the notion that performance in the time domain could be improved by using even stronger control within the constraints. The following GA was designed to explore this possibility.

IV. THE GA APPROACH

A. Method

The genetic algorithm approach, implemented as part of the experiment presented in this paper, makes use of Matlab and Simulink for running the evolutionary computation and simulating the system respectively. Figure 2 shows the Simulink structure used. The genotype is an array composed by 10 values, which position can be identified in figure 2 and function is described in table II. By means of 10 parameters, the GA tunes a feed-forward block (or prefilter), the three PID parameters, a first order filter on the derivative, a feedback gain and a Butterworth filter on the feedback signal. The anti-windup system, identifiable in

TABLE I
SIMULATION RESULTS FOR THE PID, GP AND NEW PID CONTROLLERS FOR $K = 1$, $\tau = 0.5$

	(PID)	(GP)	(tuned PID)	Characteristic
Overshoot	0.3%	0.4%	1%	limited to 2%
Rise time (ms)	391	239	210	to minimize
Settling time (ms)	629	417	326	to minimize
ITAE (mVolts · sec ²)	49.0	19.8	13.5	to minimize
Load disturbance deviation (mVolts)	6.0	0.64	0.42	to minimize
Maximum $u(t)$ (Volts)	8.6	19.4	36.0	limited to 40Volts
Maximum $\dot{u}(t)$ (Volts/sec)	460	1927	8761	unspecified/free
Bandwidth $Y/Y_{feedback}$ (rad/sec)	57.6	3070	435	unspecified/free
Bandwidth Y/Y_{ref} (rad/sec)	5.4	8.8	10.4	limited to 401rad/sec in [16]

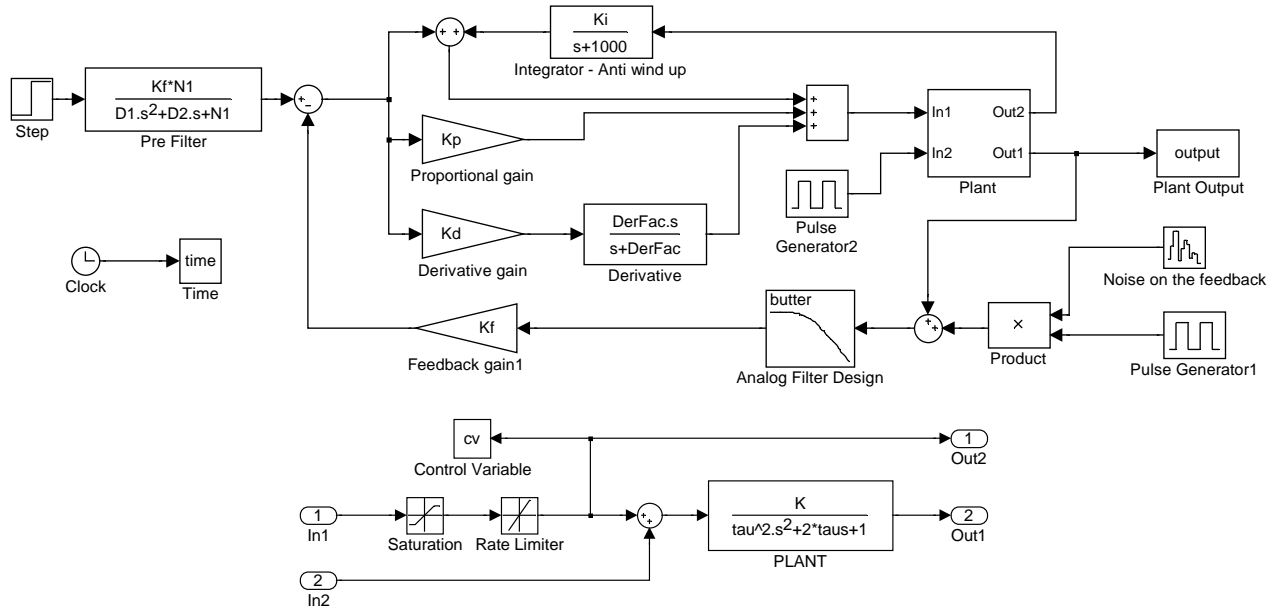


Fig. 2. Simulink model used by the genetic algorithm

the block diagram, was realized using a standard method as described in a control textbook [2, pages 423-427].

The search space was defined by the following range of values

$$\begin{aligned}
 N1 &= [0,2000]; & D1 &= [0,2000]; & D2 &= [0,2000]; \\
 Kp &= [0,2000]; & Ki &= [0,3000]; & Kd &= [0,1000]; \\
 Kf &= [0,100]; & DerFac &= [0,4000]; \\
 Bw &= [0,4000]; & Ord &= [1,8].
 \end{aligned}$$

The population was randomly initialized and seeded with randomized versions of the solution from the design in [4]

b) *Fitness function*: The fitness was evaluated for the system undergoing a step reference signal from 0 to 1 Volt, a unitary step load disturbance and feedback noise. The main performance index considered was the integral of the time-weighted absolute error (ITAE) between the reference signal and the plant output. Since the plant was simulated in four different parameter configurations, four ITAE values were added in a weighted sum. The weights were automatically adjusted during the computation so that each plant response

had the same influence. To avoid oscillations and instability, a penalty was added for an exceeding number of peaks in the control variable. To favour a smooth response, a penalty was added when, after the first peak, the output variable was moving away of $\pm 2\%$ from the reference value.

c) *Constraints*: An additional constraint was added. The derivative of the control variable, $\dot{u}(t)$, was limited to 10,000 Volts/sec. Although this is a high value, the limitation is determinant to constrain the GA search. In fact, preliminary runs showed that the search process is apt to exploit an unlimited $\dot{u}(t)$ to reach a very high load disturbance suppression until quantization and sampling noise arises and acts as a hidden constraint.

d) *GA setting*: The GA ran on a population of 300 individuals. Selection was based on a tournament mechanism within groups of 10. Mutation and crossover were applied to 20% and 70% of the population. The tournament winners (10%) were kept unmodified in the population as elitists. On 25% of mutated individuals, a vector of likely fitness improvement was added: the vector was calculated

TABLE II
BEST INDIVIDUALS OF THE GA CONTROLLER

Param.	Noise Level			Function
	None	Low	High	
N1	85.72	79.91	60.09	Pre-Filter numerator
D1	0.079	0.089	0.097	Pre-Filter denominator
D2	3.679	3.859	3.967	Pre-Filter denominator
Kp	619.7	465.3	240.2	PID proportional action
Ki	134.4	733.7	847.1	PID integral action
Kd	37.82	34.64	43.25	PID derivative action
Kf	3.34	0.19	0.034	Feedback gain
Ord	1	4	2	Butterworth filter order
DerFac	3153	1013	906.1	Pole for the derivative-filter
Bw	3145	688.9	157.3	Butterworth Bandwidth (Hz)

taking the difference of two individuals genotypes where the first one had better fitness than the second one, repeating the operation all over the population and computing the average. This vector is meant to provide an indication of a likely fitness improvement or likely descendent direction in the fitness landscape.

e) *Simulation runs:* In a first set of runs, the system was simulated without load and feedback disturbances. The search process was monitored and manually stopped. In a second set of runs, the solutions obtained in the first run were trained to adapt to load disturbance and feedback noise at different intensities. Two different levels of noise were applied. The noise was generated by the Band-Limited White Noise Simulink block, which was set to 1ms sampling step and power 2pW and 1nW for low and high noise respectively.

The process ran on a single AMD 2400+ machine with Windows XP as operating system. Given the stochastic nature of EAs, the GA controller was produced in similar versions by running the GA program 20 times for each set of runs. During the first generation, each solution was simulated for an interval of time of 10 seconds. For the following generations, to optimize the computation, the simulation time was set variable and equal to three times the settling time of the best individual.

B. Simulation Results

The GA took an average of 30 generations in the first run without noise and approximately the same number of generations for the following two runs with low and high noise. The first generation took between 200 and 300 seconds to be evaluated, depending on the level of noise. Thanks to the adaptive simulation time, the last generations took approximately one tenth of the time to be evaluated.

Three solutions chosen from the runs are presented in table II.

f) *Free of noise system:* In figures 3 and 4, the plant output and the control variable for two plant configurations ($K = 1, \tau = 1$ and $K = 2, \tau = 0.5$) are plotted when the system is not subjected to load and feedback disturbance. The nonlinearities imposed by the constraints were used by the genetic algorithm to increase the performance using the maximum control action allowed by rate limit and

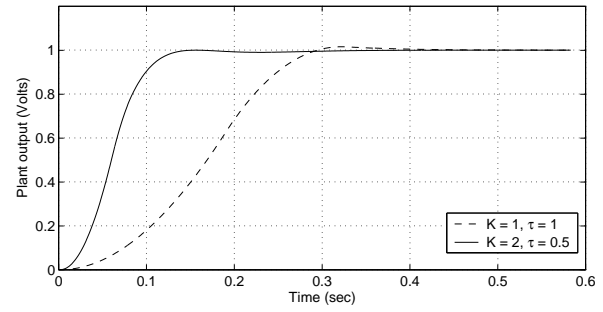


Fig. 3. Plant output of the GA controller

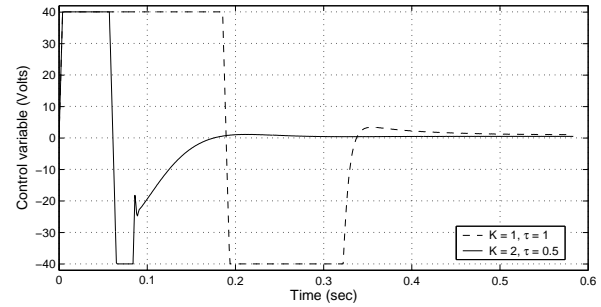


Fig. 4. Control variable of the GA controller

saturation. Hence, the system response gets faster as the system gets more reactive, see figure 3. The control variable (figure 4) shows that the search process reached a complete bang-bang control, where the upper and lower saturation limits are reached using the maximum varying rate in order to obtain the fastest plant response. Table III reports the average performance of solutions obtained in 10 runs for the free of noise system.

g) *Disturbance applied:* When noise was applied, the computation adapted the controller to the new working condition by adjusting its parameters. From table II, comparing the data for different levels of noise, the tuning operated by the GA is evident. The proportional PID parameter (K_p) was progressively lowered as well as the feedback gain (K_f). The Butterworth filter was tuned by considerably lowering the bandwidth and increasing the order. Finally, the first order filter on the derivative was significantly tuned.

By monitoring the evolutionary computation, it was observed that, for individuals of the first generation, the control variable undergoes extreme variations due to the sensitivity to the feedback signal. The control variable for the best individual of the first generation after applying the noise is plotted in figure 5. On a real plant, this behaviour of the control variable might damage the plant or cause fast wear and tear of the mechanical part of the actuator. Hence, the first target is to make the control variable as smooth as possible in spite of a loss in performance. When the controller is adapted to noise, the control variable remain smooth as shown in figure 6. The sensitivity to the feedback

TABLE III
SIMULATION RESULTS OF THE GA CONTROLLER FOR FREE OF NOISE SYSTEM

	(K = 2, $\tau = 0.5$)	(K = 1, $\tau = 1$)	Characteristic
Overshoot	2%	1.9%	limited
Rise time (ms)	75	181	to minimize
Settling time (ms)	128	298	to minimize
ITAE (mVolts · sec ²)	2.8	17.2	to minimize
Load disturbance deviation (mVolts)	2.8	2.6	to minimize
Maximum $u(t)$ (Volts)	40	40	limited
Maximum $\dot{u}(t)$ (Volts/sec)	10000	10000	limited
Bandwidth $Y/Y_{feedback}$ (rad/sec)	391	77.8	unspecified/free

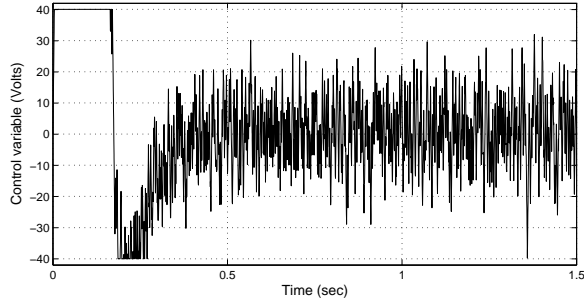


Fig. 5. Noise effect on a controller trained for a free of noise feedback signal. Noise applied: 1nW

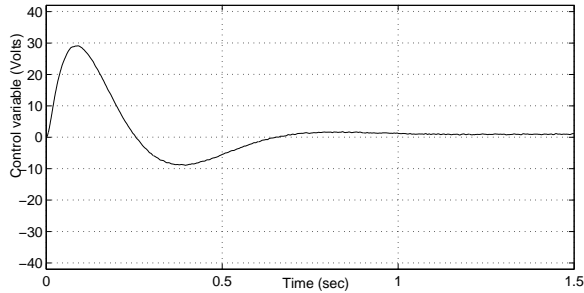


Fig. 6. Noise effect after the training of the controller. Noise applied: 1nW

signal is reduced as well as the performance in the time domain. Therefore, when the feedback signal is affected by high uncertainty, high performance are abandoned in favour of robustness.

C. Comparison with the GP Controller

The GA controller cannot be compared to the GP controller when feedback disturbance is applied. In fact, the GP controller was designed for free of noise signals only. Furthermore, the presence of a second derivative also makes it extremely sensitive to quantization noise during simulation.

Considering free of noise systems, figure 7 shows a comparison between the plant responses of the GP and GA controller for $K = 2$, $\tau = 0.5$. The GA controller has approximately the same performance as the GP controller for $K = 1$, $\tau = 1$. For $K = 2$, $\tau = 0.5$, however, the GA controller showed considerable improvements. The rise time

of 243 ms using the GP controller decreased to 75 ms using the GA controller; the settling time decreased from 419 ms to 128 ms. Finally, the ITAE recorded by the GA controller is 2.8 mVolts · sec² versus 19.9 mVolts · sec² of the GP controller.

V. CONCLUSION

The initial analysis and simulation of the GP and PID controllers outlined several different characteristics of the two controllers. The use of saturated control, not specified as a constraint in [15], [16], allowed the evolutionary computation to reach better fitness values and improve the performance of the standard PID. However, saturated control can be used only in particular control problems. The nonlinearity and the heavy use of the actuator make the controller unsuitable for most industrial applications.

The design cost is a decisive factor. The PID tuning requires few manual calculations as explained in [4, pages 697-700]. The GP controller was synthesized by a parallel computer architecture of sixty six 533MHz elements that took 44.5 hours [15], [16]. Besides, the setting of suitable initial parameters of the evolutionary computation and the design of an effective fitness calculation require additional time and skill. On the other hand, the automatic synthesis for the GP controller, as stated in [16], does not require knowledge in control theory and it is free to evolve a different structure for each kind of control problem. However, lack of knowledge in control theory favors the production of inapplicable controllers because of simulation flaws like unspecified constraints, easily identified by an experienced

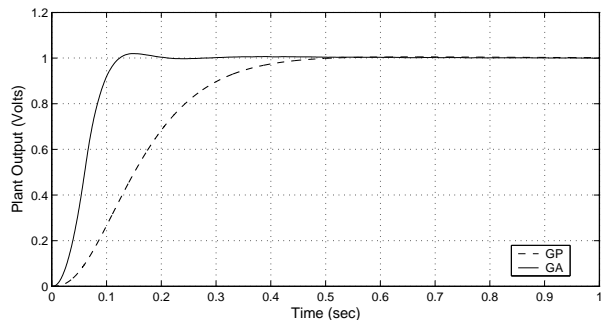


Fig. 7. Comparison of plant outputs for the GP and GA controllers ($K = 2$, $\tau = 0.5$)

control engineer but greedily explored by the evolutionary computation.

The synthesis of a GA controller requires knowledge in control theory in order to set the proper controller architecture. However, in this experiment, it is proved that the GA computation is not only capable of tuning controller parameters, but has also the potential of acting in the controller design process. In particular, the Butterworth filter is designed by the GA on the base of the level of noise present in the system. Also the first order low-pass filter on the derivative and the feedback gain were tuned according to the noise. This fact suggests the possibility of using more complex structures, without limiting to PID control, and allowing the GA to shape the optimal sub-structure that better fits the current control problem.

The simulation of the solutions from the run for the free of noise system outline the effectiveness of the GA approach compared to GP. Saturation and bang-bang control were achieved by the evolutionary process. The use of nonlinear constrained control, which leads to difficulty for the traditional design, was easily exploited by the GA for all the four plant configurations. The dramatic improvements recorded qualify the approach to optimize solutions of control problems with high performance requirements.

The computational aspect makes the GA approach feasible. The time of magnitude one or a few hours, depending on the complexity of the fitness calculation and the presence of noise, makes the method attractive for several applications. Considering a future improvement in this technique and an increase in computational power, the capability of adapting to different levels of noise could be used to design an online adaptive control, as also proposed in [20]. In fact, if the evolutionary computation speed of adaptation is faster than the rise time of noise condition, an online GA could guarantee robustness and adaptability.

REFERENCES

- [1] W. Banzhaf, P. Nordin, R. E. Keller, F. D. Francone: *Genetic Programming - An Introduction*, Morgan Kaufmann, San Francisco, CA and dpunkt, Heidelberg (1998)
- [2] P. Bolzern, R. Scattolini, N. Schiavoni: *Fondamenti di controlli automatici*, McGraw-Hill Libri Italia srl, piazza Emilia, 5, 20129 Milano (1998)
- [3] Richard C. Dorf, Robert H. Bishop: *Modern Control System Analysis & Design Using Matlab & Simulink*, Seventh Edition, Addison Wesley Longman, Inc. Menlo Park California (1997)
- [4] Richard C. Dorf, Robert H. Bishop: *Modern Control Systems*, Ninth Edition, Upper Saddle River. N.J.: Prentice Hall (2001)
- [5] D. C. Dracopoulos, S. Kent: *Genetic Programming for Prediction and Control*, Neural Computing and Application, Vol.6, No.4, pages 214-228 (1997)
- [6] P. J. Fleming, R. C. Purshouse: *Evolutionary algorithms in control system engineering: a survey*. Control Engineering Practice, Vol 10, 2002, (2002) Pages 1223-1241
- [7] G. J. Gray, D. J. Murray-Smith, Y. Li, K. C. Sharman, T. Weinbrenner: *Nonlinear model structure identification using genetic programming*. Control Engineering Practice, Vol 6. 1998. (1998), Pages 1341-1352
- [8] J. Grefenstette: *Optimization of control parameters for genetic algorithms*, IEEE Transactions of Systems, Man and Cybernetics, Volume 16, Issue 1 (1986)
- [9] M. Jamshidi, L. dos Santos Coelho, R. A. Krohling, P. J. Fleming: *Robust Control System with Genetic Algorithms*, CRC Press (2002)
- [10] H. S. Jones: *Control Apparatus*. United States Patent 2,282,726. Filed October 25, 1939. Issued May 12, 1942
- [11] J. R. Koza: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press: Cambridge, MA, 1992
- [12] J. R. Koza: Introduction to genetic programming. In Kinner, Kenneth E. Jr (editor). *Advances in Genetic Programming*, Cambridge, MA, The MIT Press. Pages 21-45. Chapter 2 (1994)
- [13] J. R. Koza: *Survey of Genetic Algorithms and Genetic Programming* Proceedings of 1995 WESCON Conference. Piscataway, NJ: IEEE. (1995) 589 - 594
- [14] J. R. Koza, Forrest H. Bennet, David Andre: *Method and Apparatus for Automated Design of Complex Structures Using Genetic Programming* U.S. Patent 5,867,397. Issued Feb 2. 1999
- [15] J. R. Koza, M. A. Keane, J. Yu, F. H. Bennett III, W. Mydlowec: *Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming*. Genetic Programming and Evolvable Machines, 1, (2000) Pages 121-164.
- [16] J. R. Koza, M. A. Keane, J. Yu, F. H. Bennet, W. Mydlowec: *U.S. Patent 6,564,194. Method and Apparatus for Automatic Synthesis Controllers*. Issued May 13. 2003
- [17] R. A. Krohling, J. P. Rey: *Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithms*, IEEE Transactions of Evolutionary Computation, Vol. 5, No. 1. February 2001.
- [18] R. A. Krohling, H. Jaschek, J.P.Rey: *Designing PI/PID Controllers for a Motion Control System Based on Genetic Algorithms*, Proceedings of the 12th IEEE International Symposium on Intelligent Control, 16-18 July 1997, Istanbul, Turkey (1997)
- [19] Renato Krohling: *Genetic Algorithms for Synthesis of Mixed H_2/H_∞ Fixed-Structure Controller*, Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference, Gaithersburg, MD, September 14-17, 1998.
- [20] W. K. Lennon, K. M. Passino: *Genetic adaptive identification and control*, Engineering Applications of Artificial Intelligence, Vol 12. (1999), pages 185-200
- [21] W. K. Lennon, K. M. Passino: *Intelligent Control for Brake Systems*, IEEE Transactions on Control Systems Technology, Vol 7, No 2, March (1999)
- [22] G. P. Liu, S. Daley: *Optimal-tuning PID control for industrial systems*. Control Engineering Practice, Vol 9, 2001. Pages 1185-1194
- [23] The MathWorks Inc.: *Using the Control System Toolbox* Version 5, Online only, Revised for Version 5.2 (Release 13). July 2002
- [24] The MathWorks Inc.: *Using Simulink* Version 5, Fifth Printing, Revised for Simulink 5 (Release 13). July 2002
- [25] The MathWorks Inc.: *Simulink Reference* Version 5, Fifth Printing, Revised for Simulink 5 (Release 13). July 2002
- [26] T. Morimoto, Y. Hashimoto: *AI approaches to identification and control of total plant production systems*. Control Engineering Practice, Vol 8. 2000. Pages 555-567
- [27] Z. Michalewics: *Genetic Algorithms + Data Structures = Evolution Programs* Third, Revised and Extended Edition, Springer-Verlag (1996)
- [28] Chester L. Nachtigal: *Instrumentation and Control, Fundamentals and applications* Wiley Interscience Publication, John Wiley & Sons, Inc (1990)
- [29] Katsuhito Ogata: *Modern Control Engineering*, Third Edition, Upper Saddle River. N.J.:Prentice Hall (1997)
- [30] G. Olsson, G. Piani: *Computer Systems for Automation and Control* Prentice Hall International (UK) Ltd (1992)
- [31] C. Vlachos, D. Williams, J.B. Gomm: *Solution to the Shell standard control problem using genetically tuned PID controllers*. Control Engineering Practice, Vol 10. 2002, Pages 151-163
- [32] P. Wang and D. P. Kwok *Optimal design of PID process controllers based on genetic algorithms* Control Engineering Practice, Volume 2, Issue 4, August 1994, Pages 641-648
- [33] K. J. Åström, T. Hägglund: *The future of PID control*. Control Engineering Practice, Vol 9, 2001. Pages 1163-1175
- [34] K. J. Åström, T. Hägglund: *PID Controllers: Theory, Design, and Tuning* Second Edition. Research Triangle Park, N.C (1995)