A Comparison of Genetic Programming and Genetic Algorithms in the Design of a Robust, Saturated Control System

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 control system for a specified second order plant is considered using three different approaches. Initially, a control system evolved by a genetic programming algorithm is reproduced and analysed in order to identify its advantages and drawbacks. The automatic design technique is compared to a traditional one through the analysis of the constraints and performance indices obtained by simulation. A set of unspecified control constraints explored by the GP search process is found to be the cause of a better performance. Hence, giving a better constraints specification, a genetic algorithm is used to evolve an alternative controller. A PID structure is used by the GA to produce and tune the controller. Simulations show a significant gain in performance thanks to a more aggressive and complete exploration of the search space within the constraints. The effectiveness of the two methods compared to the traditional approach is discussed with regard to performance, complexity of design and computational viability.

1 Introduction

In recent years, evolutionary computation has been applied to several control engineering problems. While weaknesses and strengths of traditional approaches of control system design are well known to experts in the field, evolutionary computation offers a designing and tuning tool that is not well investigated with regard to reliability, effectiveness and usability.

The new evolution based methods proposed by several scientists [4] are still not fully considered by the traditional control field. The proposed methods often lack mathematical proofs of stability, guarantees of reliability and applicability of the results. A better knowledge of the characteristics of evolutionary algorithms in control engineering could help the synthesis of more usable and feasible control systems and allow evolutionary computation to gain the status of an effective tool in control system design.

There are several weaknesses and difficulties in the design of a suitable evolutionary algorithm for control synthesis. The determination of a unique fitness

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 174–185, 2004.

value is typically a multi-objective optimization problem [13] and requires particular attention during the setting of initial parameters: a wrong choice can result in poor outcomes or failure of the search. The fidelity in the simulation of the plant is also a key factor, often affected by unknown parameters, unknown plant dynamics or noise. The difference between the real plant and the simulated plant is reflected in the evolved controller that loses reliability and performance once implemented. Therefore, the identification of constraints and characteristics of the desired controller is a decisive factor for a good design.

In this paper, the characteristics of different control systems for a robust[1], linear SISO[2] control problem are discussed. The control problem is presented in a textbook of control engineering [3, pages 697-700] and it is used as a bench mark for the analysed controllers. Initially, a GP-evolved control system presented in [11,12] is reproduced, analysed and simulated in order to highlight its advantages and flaws. A comparison is carried out with respect to the traditional design proposed in [3]. The unspecified limit of the derivative of the control variable and an unlimited bandwidth from the feedback signal to the output result in very high load disturbance suppression. The GP controller gives better performance also using a more intense control action that results in saturated control. Thus, the comparison is not relevant. An alternative controller for the same problem was evolved using a genetic algorithm. The controller structure utilizes a prefilter, a PID core and filters on the feedback and on the derivative. The high gain in performance shown by the GA controller is justified by the use of saturated, bang-bang control. The GA computation explored the search space more effectively and produced a controller that brings to the limit all the constraints. The GA approach is also found less computationally expensive and able to cover different control problems.

2 Methods

2.1 Representation of Controllers and Plant

The results presented in this paper are obtained by the simulation of the controlled systems implemented using Matlab, the Matlab Control System Toolbox and Simulink. The choice of the MathWorks Inc. software is due to the completeness of the available tools for control system engineering. The block diagram of a general controlled system is shown in figure 1.

A controller can be described as the compound of linear and nonlinear components. Linear components can be expressed by transfer functions [3,21]. The design of linear control benefits of well known mathematical theories and methodologies [3,20,21,24,25] . Nonlinear components such as saturation or rate limiter have to be expressed by special blocks or functions. They increase the complexity of the design and justify the use of simulation and evolutionary algorithms.

[1] The parameters of the plant are supposed to be time-varying between certain ranges to guarantee stability and steady performances.

[2] Single Input, Single Output.

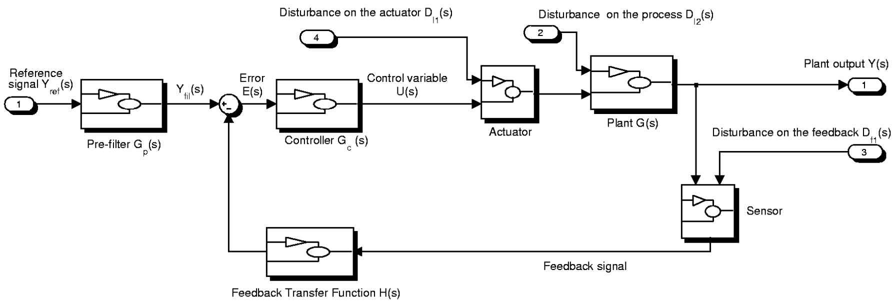

Fig. 1. General model of a controlled system

The plant to be controlled is expressed by the transfer function

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

where K and τ are considered varying between the values $1 \leq K \leq 2$ and $0.5 \leq \tau \leq 1$ to obtain robust control. The simulation of the controlled systems is carried out for the four states corresponding to the four combinations of the values $K = 1, 2$ and $\tau = 0.5, 1$. The measurements were obtained applying a step reference signal from 0 to 1 Volts.

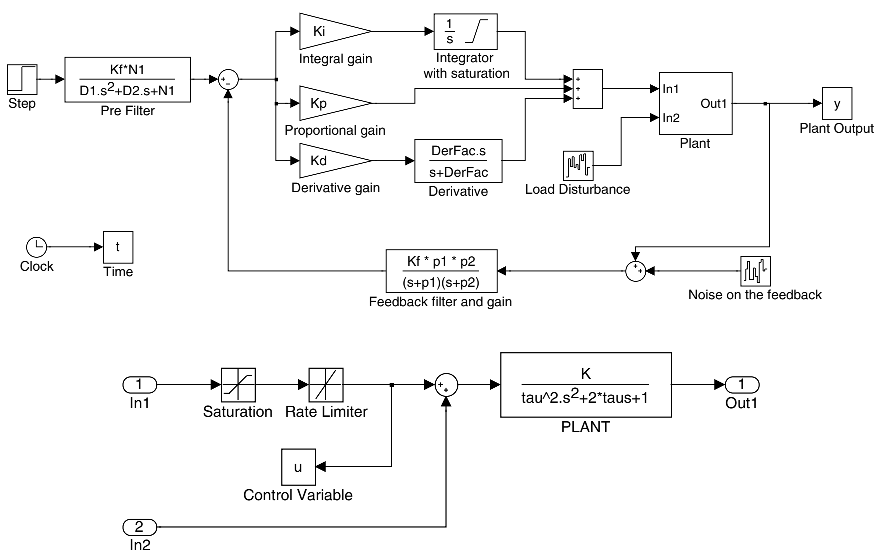

Fig. 2. Simulink model used by the genetic algorithm

2.2 Constraints and Performance Indices for a Control System

The distinction between constraints and performance indices is often blurred. Generally, a constraint is a characteristic of the controlled system that should be kept within specified boundaries. A performance index is a characteristic of the controlled system that should be minimized or maximized. The indices and constraints used in this paper are listed here with a brief description.

- Overshoot: is the amount the system output response proceeds beyond the desired value when applying a step to the reference signal.
- Rise time: is the time taken by the output to rise from 10% to 90% of the input amplitude.
- Settling time: is the time required for the system output to settle within a certain percentage of the input amplitude.
- Maximum $u(t)$: is the maximum value reachable by the control variable.
- Maximum $\dot{u}(t)$: is the maximum derivative reachable by the control variable.
- Bandwidth: is a frequency domain index that describes the reactivity of the system in following the reference signal and the sensitivity to feedback high frequency noise.
- ITAE: is the Integral of Time-weighted Absolute Error between the plant-output and the reference signal. It is the main performance index used in this experiment.
- Load disturbance deviation: is the maximum deviation of the plant-output from the reference signal when a disturbance is applied to the plant-input.

For the control problem examined, an overshoot less than 2% was considered. In [11,12] a saturation limit was imposed to 40 Volts. The derivative of the control variable was unlimited for the GP controller and limited to 10.000 Volts/sec for the GA controller.

2.3 Design Methods

The controllers presented and compared in this paper are designed using three different approaches.

The genetic programming approach has been used in [11,12] to design from scratch a controller for the the plant of equation (1). Several constraints and performance indices were used to build a fitness function. The genotype was a tree-coded controller mapped into a SPICE code for the simulation of the electric circuit. The controller proposed in [11,12] has been reproduced and simulated in this experiment.

The traditional design method is proposed in [3] and makes use of a PID controller with pre-filter to minimize the ITAE index. The design uses a parameter ω to set the intensity of the control variable and achieve optimum non-saturated control with respect to the ITAE index.

The genetic algorithm approach, implemented as part of the experiment presented in this paper, uses a PID controller with 11 parameters for tuning a pre-filter, the PID parameters, a filter on the derivative and a filter on the feedback signal. Figure 2 shows the Simulink model used by the genetic algorithm to optimize the parameters. The figure shows also the position of the 11 parameters with the exception of *IntLim* which is embedded in the integrator block.

In spite of the proposed fixed structure, the number of parameters and the different values that they can assume allow the evolutionary computation to accentuate or disable parts of the controller. Thanks to this characteristic, the method goes beyond the tuning of the three PID parameters.

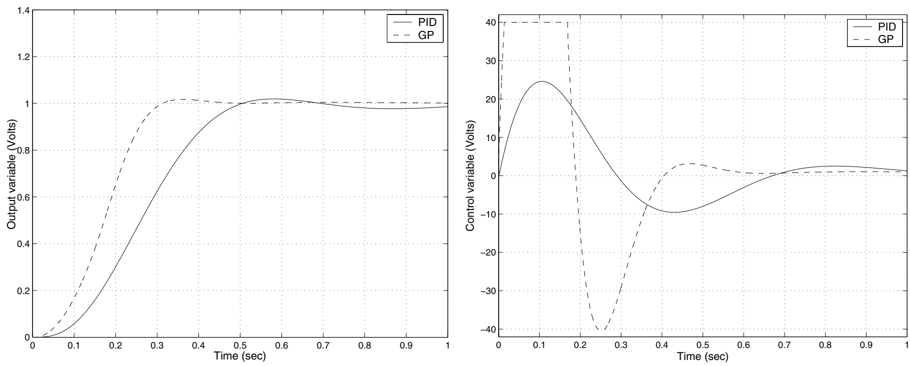

Fig. 3. Plant-outputs (left) and control variables (right) for the PID and GP controllers, plant parameters: $K = 1, \tau = 1$

3 Results

3.1 Simulation Results of the GP and PID Controllers

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 3 shows the output variable and the control variable for both the controllers. It is evident that use of nonlinear saturated control helps the GP controller to achieve better performance. Besides, the GP controller uses a second derivative that produces an infinite bandwidth from the feedback signal to the output. This fact, with the unlimited derivative of the control variable, gives a potential infinite load disturbance suppression but is not applicable to real systems. For the simulation, the derivative was implemented with an embedded low-pass filter, the same shown in figure 2, in order to obtain a finite bandwidth. From the analysis, it is evident that the controller presented in [11, 12] has substantially different characteristics from the PID and is therefore not comparable.

In a second simulation, the standard PID controller was tuned for a stronger control action, setting a tuning parameter ω to 16 instead of 8 as described in

[3, pages 697-700]; additionally, a limit on the integral was imposed to 8 Volts and a gain of 3 was added to the feedback signal. The gain on the feedback was added to increase the bandwidth of the system and obtain a better load disturbance suppression for a unitary step at the plant input. The roughly tuned controller, compared to the GP controller, obtained better performance under all the considered indices. Table 1 shows the simulation results for $K = 1$ and $\tau = 0.5$.

Table 1. Simulation results for the PID, GP and new PID controllers for $K = 1, \tau = 0.5$

	(PID)	(GP)	(new PID)	Characteristic
Overshoot	0.3%	0.4%	1%	limited
Rise time (ms)	391	239	*210*	to minimize
Settling time (ms)	629	417	*326*	to minimize
ITAE $(mVolts \cdot sec^2)$	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
Maximum $\dot{u}(t)$ $(Volts/sec)$	460	1927	8761	unspecified/free
Bandwidth Y/Y_{fil} (rad/sec)	57.6	3070	435	unspecified/free

It was observed from the simulations that both the GP controller and the new PID bring the control variable to saturation when $K = 1$ and $\tau = 1$. That is when the plant has the lowest gain (K) and the longest time constant (τ) and needs the strongest 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 cases. Figure 4 shows the plant-output and the control variable of the GP controlled system for the states ($K = 1$, $\tau = 0.5$), ($K = 2$, $\tau = 1$) and ($K = 2$, $\tau = 0.5$) .

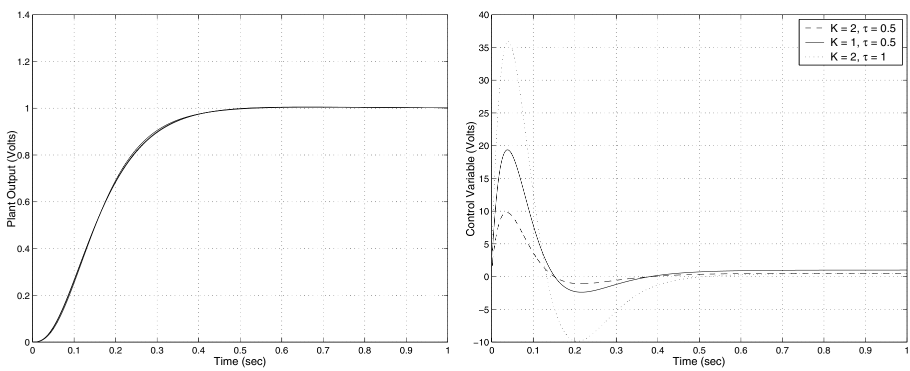

Fig. 4. Plant-outputs (left) and control variables (right) for the GP controller

Table 2. Best individual of the GA controller for noise-free system

Param.	Value	Function	Param.	Value	Function
N1	195.96	Pre-Filter numerator	Kf	3.2033	Feedback gain
D1	0.1744	Pre-Filter denominator 1	IntLim	4.7595	Limit on the integral
D2	7.7851	Pre-Filter denominator 2	DerFac	1613.8	Pole for the derivative filter
Kp	273.78	PID proportional action	p1	978.97	Pole 1 for the feedback filter
Ki	999.21	PID integral action	p2	1543.9	Pole 2 for the feedback filter
Kd	16.988	PID derivative action			

3.2 Simulation Results of the GA Controller

The GA controller was produced in several similar versions by different runs of the algorithm. Preliminary runs during the development of the application were used to identify good settings of the GA parameters. Given the stochastic nature of GAs, the final experiment was carried out by running the program 50 times. To evolve the controller, the GA computation took an average of 34 generations. The first generation took approximately 50 seconds to be evaluated.

The process ran on a laptop with processor AMD Athlon 2400+, 512Mb RAM and Windows XP as operating system.

In a first run the system was simulated without load and feedback disturbances. The population was randomly initialized and seeded with the parameters of the PID controller.

Preliminary experiments proved that the algorithm was able to reach, in a longer time, equivalent solutions without seeding the initial population. However, the quality of the seeds and their initial distribution strongly affect the number of generations required to reach the solution.

The fitness function was composed by a sum of the ITAEs of the four system responses, a penalty for overshoots greater than 2% and a penalty for a spiked or oscillating control variable to reduce the influence of feedback noise and instability. The additional constraint of 10.000 Volts/sec for the derivative of the control variable was added to make the controller applicable to real control problems. The population was composed of 300 individuals. Selection was based on a tournament within groups of 10 individuals. Two degrees of uniform distributed mutation were applied to both accelerate the initial search and finely tune the parameters. Mutation was applied to 30% of the population. A vector of likely fitness improvement was applied to one third of mutated individuals: the vector was calculated 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 provided an indication of a likely fitness improvement and increased the speed of the search by 2 to 5 times[3]. Crossover was applied to 60% of individuals, combining random parameters of

[3] The increase in speed is strongly dependent on the initial conditions given by the quality of the seeds and their distribution in the search space. This data obtained by preliminary runs could be the central issue of a further study regarding the effectiveness of the method.

two individuals chosen from two different groups. Elitism was applied by copying the best individual of each group into the next generation. The best individual of the run is characterized by the values in table 2.

Table 3. Best individual of the GA controller for a system with noise on the feedback

Param.	Value	Function	Param.	Value	Function
N1	119.8	Pre-Filter numerator	Kf	0.95	Feedback gain
D1	0.076	Pre-Filter denominator 1	IntLim	1.29	Limit on the integral
D2	4.59	Pre-Filter denominator 2	DerFac	90.02	Pole for the derivative filter
Kp	112.6	PID proportional action	p1	786.3	Pole 1 for the feedback filter
Ki	11.75	PID integral action	p2	295.3	Pole 2 for the feedback filter
Kd	13.55	PID derivative action			

During the computation, some initial parameters were adapted to adjust and direct the multi-objective search. In particular, the weights of the ITAE values, initially set to 1, appeared to be unbalanced as soon as the computation reached saturated control, giving better performance for the plants with higher gain and lower time constant. The nonlinearity in the controlled system was being used by the genetic algorithm to increase the performance using the maximum control action allowed by rate limit and saturation. Hence, the system response gets faster as the system gets more reactive. The results are shown in figure 5. The control variable shows that the computation 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. Bang-bang control provided minimum ITAE, settling time and rise time and was chosen as manual termination criterion.

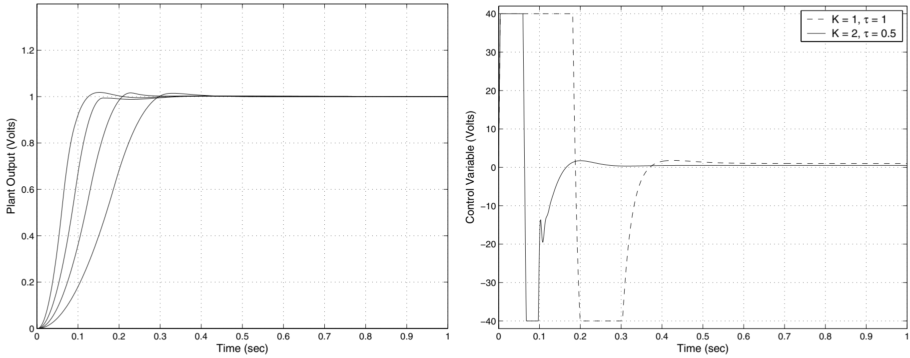

Fig. 5. Plant-outputs (left) and control variables (right) for the GA controller. In the left graph, the fastest response corresponds to $K = 2, \tau = 0.5$, the slowest to $K = 1, \tau = 1$

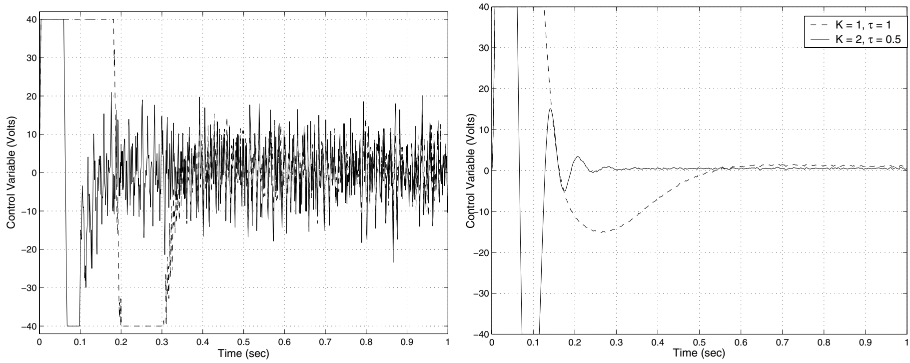

Fig. 6. Control variables at the first (left) and last (right) generation for the run with noise

In a second run, a disturbance to the feedback signal was applied. The disturbance was generated by the white band-limited noise block of the Simulink library with power $0.1nW$ and sampling time $1ms$. Randomized seeds of the best individual of the first run and the original PID controller were chosen to initialize the population. The control variable at the first generation undergoes extreme variations due to the sensitivity to the noise added to the feedback signal. 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.

To follow the optima in the dynamic fitness landscape that changes when introducing noise, the heuristic given by the vector of likely fitness improvement provided a high percentage of best individuals.

A final solution found in the second run is reported in table 3. Figure 6 shows the control variable at the first and last generation.

The GP controller was designed only for noise-free signals. 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. Figure 7 shows a comparison between the responses of two controllers for $K = 2$, $\tau = 0.5$. 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 $mVolts \cdot sec^2$ versus 19.9 $mVolts \cdot sec^2$ of the GP controller. Table 4 reports the performance of the GA controller.

4 Discussion

The 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 [11,12], allowed the evolutionary computation to

Table 4. Simulation results of the GA controller for the system without noise

	($K = 2,\tau = 0.5$)	($K = 1,\tau = 1$)	Characteristic
Overshoot	2%	1.9%	limited
Rise time (ms)	75	181	to minimize
Settling time (ms)	128	298	to minimize
ITAE ($mVolts \cdot sec^2$)	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_{fil} (rad/sec)	430	77.8	unspecified/free

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 tuning of a new PID and the synthesis of the GA controller were done considering a reduction of the set of control problems to the one where saturated control is applicable.

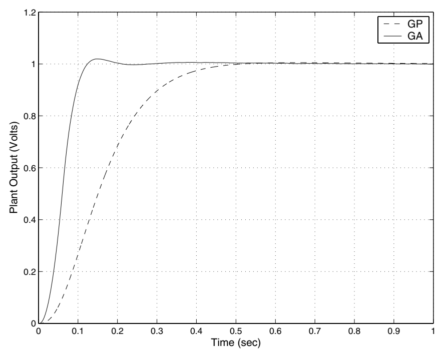

Fig. 7. Comparison of plant-outputs for the GP and GA controllers

The design cost is a decisive factor. The PID tuning requires few manual calculations as explained in [3, pages 697-700]. The GP controller was synthesized by a parallel computer architecture of sixty six 533MHz elements that took 44.5 hours [11,12]. Besides, the setting of suitable initial parameters of the evolutionary computation and the design of an effective fitness calculation require additional time and skill. The choice of wrong parameters can lead to poor results or failure [4]. On the other hand, the automatic synthesis for the GP controller, as stated in [12], 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 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 the GA computation does not only tune parameters but has some freedom to utilize particular elements in the architecture only when they are needed. This fact suggests the possibility of using a more complex structure, without limiting to PID control, and allowing the GA to shape the optimal sub-structure that better fits the present control problem. In the first run, when the system was simulated without disturbances, the low-pass filters were automatically disabled by placing poles at high frequencies. Conversely, when feedback disturbance was applied, the filters were tuned to play an important role in filtering high frequency noise. The feedback gain was also automatically lowered.

The automatically driven search forward area of likely good fitness reflects the human attitude to move in the direction that gives a likely improvement. It can easily follow the movements of a dynamic fitness landscape and here it is proven to be a decisive factor in speed and effectiveness.

From the simulation results, the dramatic improvements recorded by the GA approach qualify the method to optimise solutions for control problems with high performance requirements.

The computational aspect makes the GA approach feasible. The time of the magnitude of one or a few hours on a single machine, depending on the complexity of the fitness calculation and availability of good seeds, makes the method attractive for several applications. With an increment of computational power and a plant with a long time constant, it would be possible to apply the method to online optimisation or synthesis of adaptive control systems.

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. R. C. Dorf, R. H. Bishop: *Modern Control System Analysis & Design Using Matlab & Simulink*, Seventh Edition, Addison Wesley Longman, Inc. Menlo Park California (1997)
3. R. C. Dorf, R. H. Bishop: *Modern Control Systems*, Ninth Edition, Upper Saddle River. N.J.: Prentice Hall (2001)
4. P. J. Fleming, R. C. Purshouse: *Evolutionary algorithms in control system engineering: a survey.* Control Engineering Practice, Vol 10. 2002, (2002) Pages 1223-1241
5. 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
6. M. Jamshidi, L. dos Santos Coelho, R. A. Krohling, P. J. Fleminig: *Robust Control System with Genetic Algorithms*, CRC Press (2002)
7. J. R. Koza: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press: Cambrigde, MA (1992)

8. 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)
9. J. R. Koza: *Survey of Genetic Algorithms and Genetic Programming* Procedings of 1995 WESCON Conference. Piscataway, NJ: IEEE. (1995) 589 - 594
10. 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
11. J. R. Koza, Martin A. Keane, Jessen Yu, Forrest H. Bennett III, William Mydlowec: *Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming.* Genetic Programming and Evolvable Machines, 1, (2000) Pages 121-164.
12. J. R. Koza, Martin A. Keane, Jessen Yu, Forrest H. Bennet, William Mydlowec: *U.S. Patent 6,564,194. Method and Apparatus for Automatic Synthesis Controllers.* Issued May 13. 2003
13. G. P. Liu, J. B. Yang, J. F. Whidborne: *Multiobjective Optimisation and Control* Research Studies Press LTD. Baldock, Heartfordshire, England. (2003)
14. G. P. Liu, S. Daley: *Optimal-tuning PID control for industrial systems.* Control Engineering Practice, Vol 9, 2001. Pages 1185-1194
15. The MathWorks Inc.: *Using the Control System Toolbox* Version 5, Online only, Revised for Version 5.2 (Release 13). July 2002
16. The MathWorks Inc.: *Using Simulink* Version 5, Fifth Printing, Revised for Simulink 5 (Release 13). July 2002
17. The MathWorks Inc.: *Simulink Reference* Version 5, Fifth Printing, Revised for Simulink 5 (Release 13). July 2002
18. Z. Michalewics: *Genetic Algorithms + Data Structures = Evolution Programs* Third, Revised and Extended Edition, Springer. (1996)
19. T. Morimoto, Y. Hashimoto: *AI approaches to identification and control of total plant production systems.* Control Engineering Practice, Vol 8. 2000. Pages 555-567
20. Chester L. Nachtigal: *Instrumentation and Control, Fundamentals and applications* Wiley Interscience Publication, John Wiley & Sons, Inc (1990)
21. Katsuhito Ogata: *Modern Control Engineering*, Third Edition, Upper Saddle River. N.J.:Prentice Hall (1997)
22. 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
23. 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
24. K. J. Åström, T. Hägglund: *The future of PID control.* Control Engineering Practice, Vol 9, 2001. Pages 1163-1175
25. K. J. Åström, T. Hägglund: *PID Controllers: Theory, Design, and Tuning* Second Edition. Research Triangle Park, N.C (1995)