A Case Study of GP and GAs in the Design of a 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[1] control system is considered using a traditional approach, a genetic programming and a genetic algorithm method. Initially, an existing GP-evolved control system is reproduced and compared to a traditional PID[2] in order to identify its advantages and drawbacks. 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 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

The work presented in this paper started in August 2003 as a case study project. The aim was to investigate the application of evolutionary algorithms to control engineering and identify particular topics to be considered for a master thesis and possible future research. The study of a previous work [3, 4] and the presentation of the result was followed by the author's orginal implementation of a GA for the design of a control system.

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 [2] often lack mathematical proofs of stability or optimality, guarantees of reliability and applicability of the results.

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

[2] Proportional, Integral, Derivative control

There are several weaknesses and difficulties in the design of a suitable evolutionary algorithm for control synthesis. The determination of a unique fitness value is a complex multi-objective optimization problem [5]. The fidelity of the plant's simulation is also a key factor, often affected by unknown parameters, unknown plant dynamics or noise.

In this paper, the characteristics of different control systems for a robust, linear SISO[3] control problem are discussed. The control problem is presented in a textbook of control engineering [1, pages 697-700] and it is used as bench mark for the analysed controllers.

A more detailed description of this experiment can be found in [7].

2 Method

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.

Linear components can be expressed by transfer functions within blocks avaiable from the Simulink library. Nonlinear components such as saturation or rate limiter are also included in the Simulink library. Nonlinear elements increase the complexity of the design and justify the use of simulation and evolutionary computation.

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 $1 \le K \le 2$ and $0.5 \le \tau \le 1$ 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.

2.2 Constraints and Performance Indices

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.

In this experiment, the indices used were overshoot, rise time, settling time, limited control variable, limited derivative of the control variable, a bandwidth index, the integral of time-weighted absolute error between the plant output and the reference signal (ITAE) and the plant output deviation when subject to load disturbance. Accurate descriptions of performance indices and constraints can be found in the literature [1, 6, 9, 10].

[3] Single Input, Single Output.

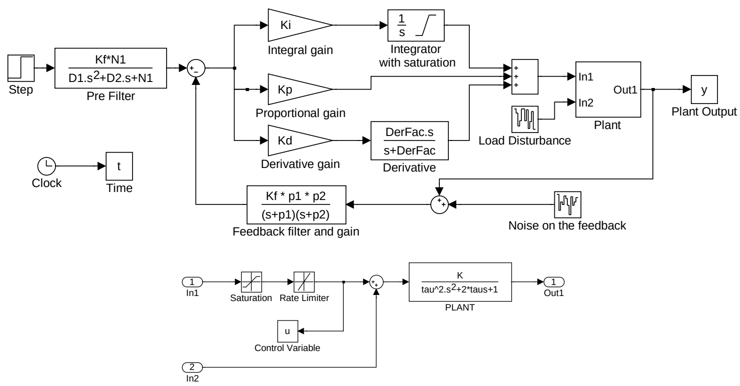

Fig. 1. Simulink model used by the genetic algorithm

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 [3, 4] to design from scratch a controller for the the plant of equation (1). The controller proposed in [3, 4] has been reproduced and simulated in this experiment.

The traditional design method is proposed in a control engineering textbook [1].

The genetic algorithm approach, implemented as part of the experiment presented in this paper, uses a PID controller with 11 parameters for tuning a prefilter, the PID parameters, a filter on the derivative and a filter on the feedback. Figure 1 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.

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 process was manually stopped. A population of 300 individuals was randomly initialized and seeded with the parameters of the textbook controller. The fitness function was composed by a weighted sum of the ITAEs of the four system responses, an index to keep the overshoot less than 2% and a measurement of 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.

Selection was based on a tournament within groups of 10 individuals. Following, mutation and crossover were applied. Two degrees of uniform distributed mutation were applied to both accelerate the initial search and finely tune the parameters. Mutation was applied to 20% of the population. On 10% of individuals a vector of likely fitness improvement was added: 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. Crossover was applied to 70% of individuals.

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 makes use of saturated control and higher varying rate of the control variable. Figure 2 shows the output variable and the control variable for both the controllers. It is evident that use of nonlinear saturated control helps

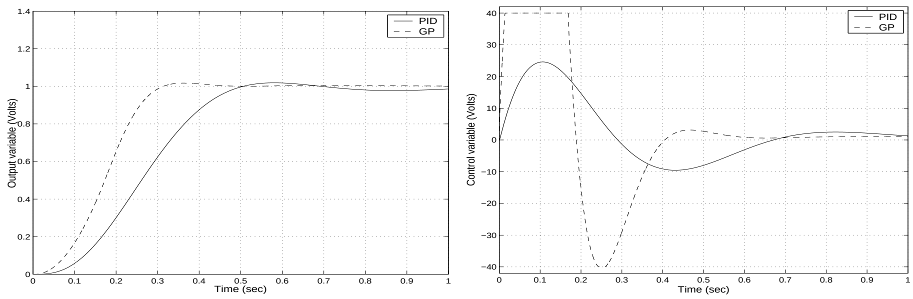

Fig. 2. Plant outputs (left) and control variables (right) for the PID and GP controllers, plant parameters: $K = 1$,$\tau = 1$

the GP controller to achieve a better performance. The controller presented in [3, 4] 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 [1, 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 newly tuned controller, compared to the GP controller, obtained better performance under all the considered indices. This result proves clearly that the GP controller does not beat the text book controller as claimed in [3, 4].

3.2 Simulation Results of the GA Controller

The best individual of one the runs is characterized by the following values. N1 = 195.96; D1 = 0.1744; D2 = 7.7851; Kp = 273.78; Ki = 999.21; Kd = 16.988; Kf = 3.2039; IntLim = 4.7595; DerFac = 1613.8; p1 = 978.97; p2 = 1543.9. The values are located in figure 1 with the exception of IntLim that is embedded in the integral block.

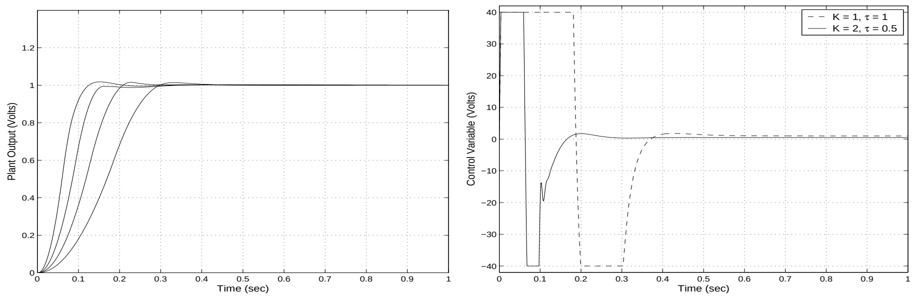

Fig. 3. Plant outputs (left) and control variables (right) for the GA controller. In the left graph, the fastest response corresponds to $K = 2$,$\tau = 0.5$, the slowest to $K = 1$,$\tau = 1$

Different runs proved that the algorithm was able, in a longer time, to reach equivalent solutions without seeding the initial population. During the computation some initial parameters were changed 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 four system responses get faster as the system gets more reactive. The results are shown in figure 3. 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 limit in order to obtain the fastest plant response. Comparing figure 2 and figure 3, it is evident the considerable improvement provided by the GA controller. In particular, the ITAE recorded by the GA controller ($K = 2$, $\tau = 0.5$) is 2.8 $mVolts \cdot sec^2$ versus 19.9 $mVolts \cdot sec^2$ of the GP controller.

In a second run, a disturbance to the feedback signal was applied. The algorithm showed the capability to adapt the controller by introducing a low-pass filter and providing a clean control variable after approximately 30 generations.

4 Discussion

The use of saturated control, not specified as a constraint in [3, 4], allowed the GP controller to 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 was done considering a reduction of the set of control problems to the one where saturated control is applicable.

From the computational point of view, the GP approach is very expensive: in fact the GP controller was synthesized by a parallel computer architecture of sixty six 533MHz elements that took 44.5 hours [3, 4]. Contrary, the design of the PID requires few manual calculations as explained in [1, pages 697-700].

The GA approach proved to be less computational expensive and able to reach considerably better performance. In particular, nonlinearities were used the reach the best performance and a completed bang-bang control was achieved. The flexibility of the algorithm to work with different plant parameters setting or presence of noise suggests a possible future experiment in the direction of adaptive control. The effectiveness of the search method could be tested online to give the current control problem the intrinsic adptive nature of EA.

References

1. R. C. Dorf, R. H. Bishop: *Modern Control Systems*, Ninth Edition, Upper Saddle River. N.J.: Prentice Hall (2001)
2. P. J. Fleming, R. C. Purshouse: *Evolutionary algorithms in control system engineering: a survey.* Control Engineering Practice, Vol 10. 2002, (2002) Pages 1223-1241
3. 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.
4. 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
5. G.P.Liu, J.B.Yang, J.F. Whidborne: *Multiobjective Optimisation and Control* Research Studies Press LTD, Baldock, Hearfordshire, England (2003)
6. Katsuhito Ogata: *Modern Control Engineering*, Third Edition, Upper Saddle River. N.J.:Prentice Hall (1997)
7. Andrea Soltoggio: *A Comparison of Genetic Programming and Genetic Algorithms in the Design of a Robust, Saturated Control System*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), June 2004, Seattle, WA, USA.
8. 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
9. K. J. Åström, T. Hägglund: *The future of PID control.* Control Engineering Practice, Vol 9, 2001. Pages 1163-1175
10. K. J. Åström, T. Hägglund: *PID Controllers: Theory, Design, and Tuning* Second Edition. Research Triangle Park, N.C (1995)