An Enhanced GA to Improve the Search Process Reliability in Tuning of Control Systems

Andrea Soltoggio
School of Computer Science
The University of Birmingham
Birmingham, B15 2TT, UK
A.Soltoggio@cs.bham.ac.uk

ABSTRACT

Evolutionary Algorithms (EAs) have been largely applied to optimisation and synthesis of controllers. In spite of several successful applications and competitive solutions, the stochastic nature of EAs and the uncertainty of the results have considerably hindered their use in industrial applications. In this paper we propose a Genetic Algorithm (GA) for tuning controllers for classical first and second order plants with actuator nonlinearities. To increase the robustness of the algorithm we introduce two features: 1) genetic operators that perform directional mutations, 2) selection tournaments organized by genome vicinity. The experiment results show that the proposed GA is able to guarantee high performance and low variance in the results from different runs. The increased reliability, compared to the results from a classical GA, seems to favour particularly the application of Evolutionary Computation (EC) in tuning of control systems, where, thanks to this approach, a large search space can be searched repeatedly with high consistency in the solutions.

Categories and Subject Descriptors: I.2.8 [Problem solving, Control Methods and Search] - Control Theory; G.1.6 [Optimization] - Constrained Optimization.

General Terms: Algorithms, Design, Reliability

Keywords: Control Systems, Design and Tuning, Genetic Algorithms, Nonlinear Control

1. INTRODUCTION

Evolutionary computation (EC) techniques have been applied to control system problems of different nature, from classical analogue/digital control, to fuzzy and neural control [7, 9, 10, 11, 15, 16, 21, 22, 19, 20, 5, 8, 13, 12, 4, 14]. A considerable effort has been made trying to identify which evolutionary algorithm better suits a given control problem [6]. Most EA approaches to the synthesis and tuning of control systems focus on the performance of the obtained solutions in comparison to the ones provided by traditional techniques. However, in spite of considerable improvements that EC techniques have shown in particular control problems [15, 16, 21, 22, 8, 14], traditional and well known approaches described in control textbooks [18, 2, 17, 1], often supported by mathematical proofs of stability, are still preferred for most control tasks. This is mainly due to the stochastic nature of EC search processes that do not guarantee a minimum performance in their outcomes. Lack of theoretical justifications and simulation risks also represent an obstacle. These facts have considerably limited the use of EC techniques for the design of online optimisation and safety- or mission-critical systems.

The GA proposed in this paper tackles the issue of providing a reliable algorithm that minimises the effect of EAs' stochastic nature and the high variance of solutions provided by different runs. The problem on which the algorithm is tested consists of tuning feedback controllers for first and second order plants that present actuator nonlinearities. The algorithm allows the synthesis of robust, linear and nonlinear bang-bang control using a PID core, a pre-filter, an anti-windup device and additional filters on the feedback. The tuning is extended to all the controller parameters and is not limited to just three PID values as in [12]. The approach presented in [19, 20], which proposes the tuning of a controller for a second order plant with actuator nonlinearities, was adopted and enhanced to tackle the search process effectiveness and reliability.

To pursue this task, two features were added to a standard GA. The first is the use of a set of genetic operators that perform directional mutation. Individuals store a copy of their parents' genotype in their own genotype. Successful individuals can therefore identify a promising climbing direction and generate their offspring according to that information. A second characteristic involves organising selection tournaments based on the genotype vicinity. Although a speciation mechanism is not implemented, the approach is meant to preserve separate groups of individuals that are climbing different areas of the search space. The population diversity is also better preserved.

The results outline how these features adopted in a standard GA can considerably affect the performance of the search process in terms of computational efficiency but, above all, results quality and reliability.

2. METHODS

In this section we first describe the control problems used as bench marks for the GA. Following, we describe 1) the use of tournaments organised by genome proximity, 2) the use of GA operators that apply directional mutation and the interaction of the two. Finally, we describe the general GA architecture used in the experiments.

2.1 The Control Problems

The plants chosen to run the optimisation on controllers are expressed by the transfer functions

$$G_1(s) = \frac{\mathrm{K}}{(\tau s + 1)} \quad , \tag{1}$$

$$G_2(s) = \frac{\mathrm{K}}{(\tau s + 1)^2} \quad , \tag{2}$$

$$G_3(s) = \frac{\mathrm{K}}{(as + 1)(bs + 1)} \quad . \tag{3}$$

The transfer functions represent, in the frequency domain, the linear systems expressed respectively by the differential equations

$$A_1 \frac{dy(t)}{dt} + y(t) = Ku(t) \quad , \tag{4}$$

$$B_2 \frac{d^2y(t)}{dt^2} + B_1 \frac{dy(t)}{dt} + y(t) = Ku(t) \quad , \tag{5}$$

$$C_2 \frac{d^2y(t)}{dt^2} + C_1 \frac{dy(t)}{dt} + y(t) = Ku(t) \quad , \tag{6}$$

where $A_1 = \tau$, $B_2 = \tau^2$, $B_1 = 2\tau$, $C_2 = ab$, $C_1 = a + b$. Detailed descriptions of physical models represented by the previous linear equations are proposed in control textbooks [2, 18]. In the experiment shown in this paper, mathematical models are used for the simulation without addressing particular physical systems.

Equation 1 is a first order plant with gain K and a single pole in $1/\tau$. Equation 2 represents a second order plant with coincident poles in $1/\tau$ as described in [2] and later used as bench mark for EAs in [10, 11, 19]. Equation 3 is a second order plant with distinct poles in 1/a and 1/b. In order to obtain robust control, as in [19, 10, 11], the simulation of the controlled systems is carried out for the four states corresponding to the four combinations of the values $\mathrm{K} = 1, 2$ and $\tau = 0.5, 1$.

The linear plants as described by the previous equations do not constitute a particular challenge in control engineering. Methods for the synthesis and tuning of such controller are presented in traditional control textbooks [2, 18, 1]. When actuator nonlinearities are introduced though, the control becomes nonlinear and the synthesis and tuning more difficult. In this experiment, we set a limitation of the control variable to ± 10 Volts and a rate limit of 1000Volts/sec for the plant of equation 1; saturation ± 40 Volts and rate limit 10,000 Volts/sec for the plants of equations 2 and 3. Overshoot within 2% is also required.

The control problem of equation (2) is followed, in [2, 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 time domain index ITAE, the integral of the time-weighted absolute error

$$\mathrm{ITAE} = \int_{t_1}^{t_2} |e(t)| \cdot t\mathrm{d}t \quad , \tag{7}$$

where e(t) is the error between the reference signal and the plant output. The ITAE index is used as the main performance indicator in the experiments presented in this paper. In section 2.4.1, a detailed description of the fitness function is given.

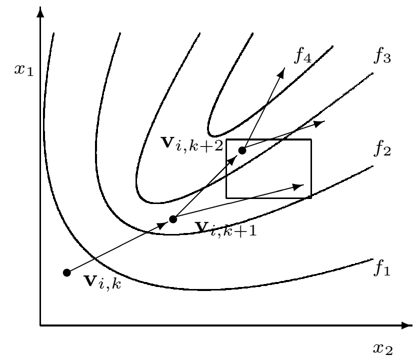

Figure 1: Directional mutation after random mutation in combination with local tournament selection. Representation in a two-dimensional fitness landscape.

2.2 Selection Tournaments Organised by Genome Vicinity

Ill-behaved fitness landscapes are likely to present several hills, different climbing directions and local optima. When the search space is also very large, the initial distribution of the population can greatly affect the final solution. For our control problems, preliminary experiments have shown that the quality of the initial population, randomly generated or seeded with known solutions, can considerably change the quality and characteristic of the outcome. It was observed that during the initial generations, good quality seeds in the population were taking over, destroying all the randomly generated individuals. Although a low selection pressure could work in this case, we attributed the problem to the competition between individuals that are far located in the search space.

To avoid this problem, we implemented a tournament selection that gathers individuals in groups by genotype proximity. Thus, the selection pressure is increased among individuals in the same search space area, and decreased among individuals far located. This way diversity is well preserved. The difference with a speciation mechanism is that the population eventually converges automatically without implementing extinction. Clustering is performed by considering the normalised distances from one individual to the others, gathering the closest k, where k is the size of the tournament, and repeating the operation starting from the $k+1$th individual until the whole population is covered.

2.3 Directional Exploration

Gradient information is used when performing three different special operators that implement directional mutation after random mutation, directional mutation after crossover and global directional mutation as explained as follows.

2.3.1 *Directional Mutation after Random Mutation*

As shown in figure 1, the idea is to use a successful mutation that has brought a fitness increment in the previous generation and to explore the search space in the same direction. In the figure, x_1 and x_2 are two dimensions of a

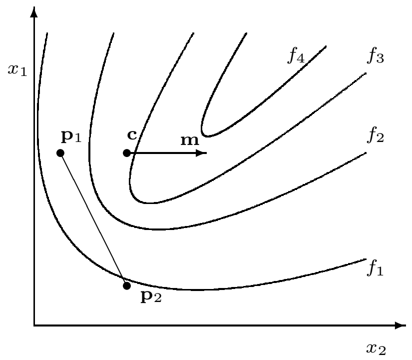

Figure 2: Directional mutation after crossover.

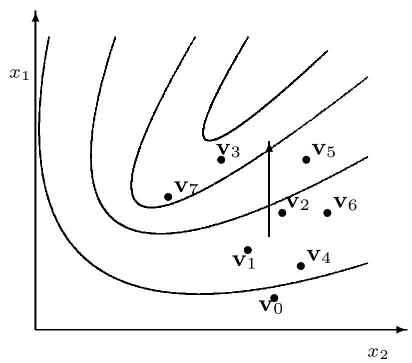

Figure 3: Global directional mutation.

search space, the point $v_{i,k}$ represents the individual i in the population at generation k and $v_{i,k+1}$ a mutation of v_i at generation $k+1$. The new offspring $v_{i,k+2}$ is generated by applying the same mutation that had brought a fitness improvement. To make it more effective, a random perturbation is applied and two offspring are generated. When the selection mechanism exerts selection pressure on individuals with similar genome, as explained previously in section 2.2, it is likely that one of the two siblings will win over the other. In this way, a successful path will likely follow the gradient, driving the search in the steepest direction.

2.3.2 Directional Mutation after Crossover

When the individual is generated from two parents, a simple mechanism can be applied. The vector that identifies the difference between the child and one parent is taken as mutation vector and applied again to the child. In the experiment we have chosen to consider the parent that is closer to the child. A more complex mechanism that considered both parents and a linear combination of the respective distances was also experimented with. However, the results did not justify the implementation of such a complex operator and the first solution was maintained. The situation is pictured in figure 2.

2.3.3 Global Directional Mutation

When the population is distributed on one side of a hill, or the landscape presents a slope, it is possible to identify a direction along which there is a steady fitness gradient. The direction can be identified by summing all the vectors representing the difference between two individuals where the first has a better fitness than the second. Different methods can be used to compute an average fitness gradient. Here we simply sort the individuals from the best to the worst and sum the *n-1* vectors obtained by subtracting two successive individuals:

$$\mathbf{m} = \sum_{i=1}^{n-1} (\mathbf{v}_i - \mathbf{v}_{i+1}), \tag{8}$$

where n is the population size.

It is important to notice that if the population does not present a fitness gradient, for example because it is distributed symmetrically around an optimum, the summation should provide a small, near to zero vector. In that case, the application of this vector as mutation has little effect on the individuals. Hence, the average directional mutation becomes active only when the population is placed as in figure 3, while it remains inactive when the population is distribute around a maximum or on an ill-behaved landscape. The global directional mutation, as just explained, can be also computed for sub-populations with the aim of providing more local information to groups of individuals localised in certain areas of the landscape. For simplicity, in this experiment, the global directional mutation was computed for the overall population only.

2.4 The GAs Used for the Experiment

Two different GAs were used to tune the controllers for the plants previously presented. A Standard GA (SGA) and an Enhanced GA (EGA) were tested both on performance of the solutions and their consistency in several runs. EGA implements directional mutation and localised tournaments as described above in section 2.2 and 2.3 whereas SGA does not.

Both algorithm ran on a population of 400 individuals randomly initialised. The selection mechanism was based on a tournament selection of size 10. When applied, mutation acted on each gene with probability 0.5 and applied a uniformly distributed random mutation in the range $\pm 4\%$ of the gene value. Mutation and crossover were applied to 20% and 70% of the population. The tournament winners (10%) were kept unmodified in the population as the elite.

In the enhanced GA, mutated individuals formed only 10% of the population. Crossover was applied to 60% and the special operators for directional mutation to 20% of the population. The new operators were applied as follows: directional mutation after random mutation or after crossover to 15% of the population and global mutation to 5%. Note that since it is not known a priori whether the individual chosen for reproduction comes from one parent, two parents, or elitism, it is not always possible to apply the desired operator for directional mutation. For instance, directional mutation after random mutation can be applied if mutation itself was successful to a certain degree during the previous generation. If no individuals generated by mutation are selected, directional mutation cannot be applied. In this case, random mutation or crossover are applied. In this way, the mechanism favours the application of operators that had better results in the previous generation.

The GAs make use of Matlab and Simulink for running both the evolutionary computation and simulating the system respectively.

Figure 4 shows the Simulink structure used. The genotype is an array composed of 10 values, where the position of each can be identified in figure 4. By means of 10 parameters, the

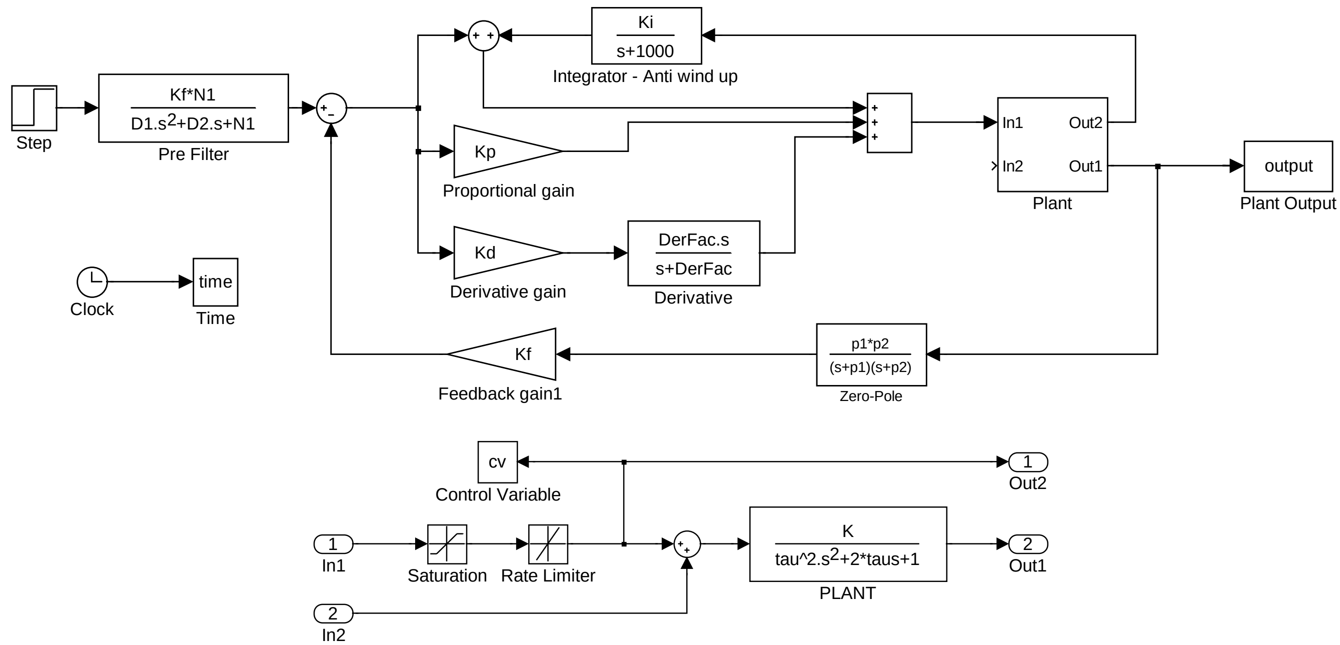

Figure 4: Simulink model used by the genetic algorithm. The plant reported in the picture is the second order plant with coincident poles. For the simulation of each plant, the block called PLANT was modified according to the required transfer function.

GAs tune a feed-forward block (or prefilter), the three PID parameters, a first order filter on the derivative, a feedback gain and a two pole low-pass filter on the feedback signal. The anti-windup system, as shown in the block diagram, was realised using a standard method as described in [3, pages 423-427].

The search space was defined by the following range of values for the alleles of each gene

N1 = [0,1000];	D1 = [0,10];	D2 = [0,200];
Kp = [0,2000];	Ki = [0,3000];	Kd = [0,1000];
Kf = [0.1,10];	DerFac = [0,3000];	
p1 = [0,3000];	p2 = [0,3000].	

2.4.1 Fitness Function

The fitness was evaluated for the system undergoing a step reference signal from 0 to 1 Volt. The main performance index considered was the ITAE between the reference signal and the plant output as expressed by equation 7. Since the plants were simulated in four different parameter configurations, four ITAE values were added. In the ITAE measurement, the simulation time plays an important role: the error between the reference signal and the plant output is weighted on time which means that a long simulation time will amplify a non perfect alignment of reference and output signal. In most cases, errors smaller than 0.1% are irrelevant to the correct behaviour of the controlled system, however, a long simulation will cause the ITAE index to grow indefinitely. For this reason, the ITAE indices reported in tables are relative to the simulation time indicated for each plant.

The behaviour of the output variable alone, as described by the ITAE index, is however not descriptive of the overall system dynamics. The control variable has to be monitored as well to avoid oscillations and instability. To avoid oscillations, a penalty was added when the number of peaks in the control variable exceeded a maximum value of 6. To favour a smooth response, a penalty was added when, after the first peak, the output variable was moving $\pm$ 2% away from the reference value.

3. RESULTS

3.1 First Order Plant

A SGA and EGA were run on the plant of equation 1 for 50 generations. The final simulation time was set to 0.5 sec.

The outcome of the evolutionary process is the best individual of the last generation characterised by its genotype and the ITAE performance index. The performance of solutions obtained from 10 distinct runs are reported in table 1 (left columns). To test the algorithms' robustness, 20 additional runs were performed for both EGA and SGA. The average performance, together with the standard deviation, are reported at the bottom of table 1. Two solutions (best individuals at the end of a run) are shown in table 2. Figure 5 shows the output of the plant and the saturated control variable for K = 1, $\tau = 1$ and K = 2, $\tau = 0.5$. The nonlinear saturated control causes the plant to react faster when the gain is high (K = 2) and the time constant low ($\tau = 0.5$).

3.2 Second Order Plant with Linear Control

The plant of equation 2 was simulated to tune a control system that emulates the linear control as proposed by the traditional method of synthesis in [2, pages 597-600]. The traditional method gives optimal linear control with a maximum measured value of the control variable of 24.6Volts and a maximum derivative of the control variable

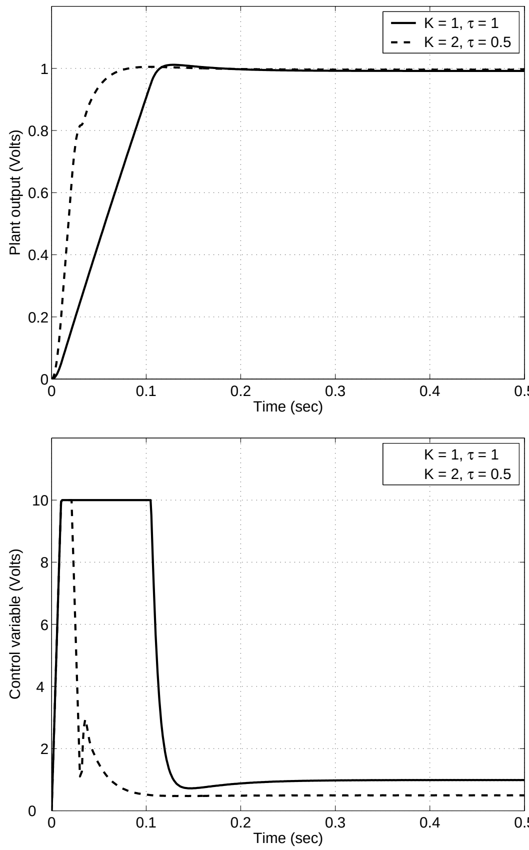

Figure 5: Plant output and control variable of the controller for the first order plant.

	1^{st} order plant		2^{nd} order plant	
run	EGA	SGA	EGA	SGA
1	4.36	14.31	191.3	192.6
2	4.85	174.50	195.6	255.8
3	5.21	101.23	194.8	195.7
4	4.63	188.87	194.0	199.0
5	5.95	13.49	189.0	288.5
6	4.18	5.80	191.2	717.8
7	4.77	13.05	196.0	255.8
8	4.53	9.08	192.9	201.1
9	4.26	1038.41	205.4	197.5
10	5.03	30.3	192.8	303.3
Average of 30 runs including the previous 10 runs				
	4.46	114.860	194.39	259.07
Standard deviation of ITAE indices				
	0.75	272.42	4.76	142.75

Table 1: Sum of the four ITAE values (in mVolts $\cdot$ sec^2) for the first order saturated controlled system and second order linear controlled system evolved by the enhanced GA (EGA) and standard GA (SGA)

	Controller 1	Controller 2
N1	1000	1000
D1	0.93	0.294
D2	65.91	28.19
Kp	138.21	902.69
Ki	117.02	142.7
Kd	6.26	971.07
Kf	0.76	0.21
DerFac	1589	0.007
p1	20.37	1636
p2	1681	1743
Performance indices		
ITAE (mVolts·sec^2)	2.42/0.75[a]	2.23/0.81
Settling time (msec)	94/63	101/70

[a]The values are reported here as x_1/x_2, where x_1 refers to the plant with K=1,$\tau = 1$ and x_2 to the plant with K=2,$\tau = 0.5$

Table 2: Two saturated controllers for the first order plant

of 495Volts/sec. The sum of the four ITAE values given for the four combinations of K and τ was measured as 193.7 mVolts·sec^2.

To test both the GAs, we set a maximum allowable control variable to 25 Volts and its maximum derivative to 500 Volts/sec: i.e. slightly higher than what was measured for the traditional controller. Thus, we expected the GAs to find solutions with performance around the optimal provided in [2]. Table 1 (right columns) shows the ITAE values recorded by the enhanced GA and standard GA in 10 runs. Simulation time was set to 1.4 sec.

Figure 6 shows the plant output and the control variable of the traditional controller and an EGA controller tuned in one of the run. The ITAE index, as shown in table 1, rise time and settling time of the optimal traditional controller are repeatedly reached by the GA tuned controllers. However, the behavioural similarity among different EGA tuned controllers do not correspond to a genotypical similarity: different runs produce very different controllers. The traditional optimal controller proposed in [2] can be simulated with the Simulink model of figure 4 given the genotype: [42.67 1 11.38 136.36 512 12 1 2000 2000 2000]. Two near optimal solutions obtained by the EGA have the following genotypes: [352.81 7.67 86.85 693.11 363.76 61.238 0.10745 740.25 614.1 256.04] and [287.83 6.0574 70.089 1116.7 3.8536 99.299 0.1 461.5 999.79 236.16], where the order of the parameters is as in table 2.

3.3 Second Order Plant with Actuator Nonlinearities

The plant of equation 2 was simulated to obtain saturated control as well and reproduce the results presented in [19]. In the experiments here, however, we focus on the reliability of the algorithm to discover repeatedly near optimal solutions. In table 3 (left columns), the performance of solutions obtained by the EGA and SGA algorithms are reported for 10 runs. The termination criterion was set to 50 generations. The controllers provided by the 30 runs recorded an average settling time of 302ms for K = 1, τ = 1, and 131ms for K = 2,τ = 0.5. The average performance reproduced the best

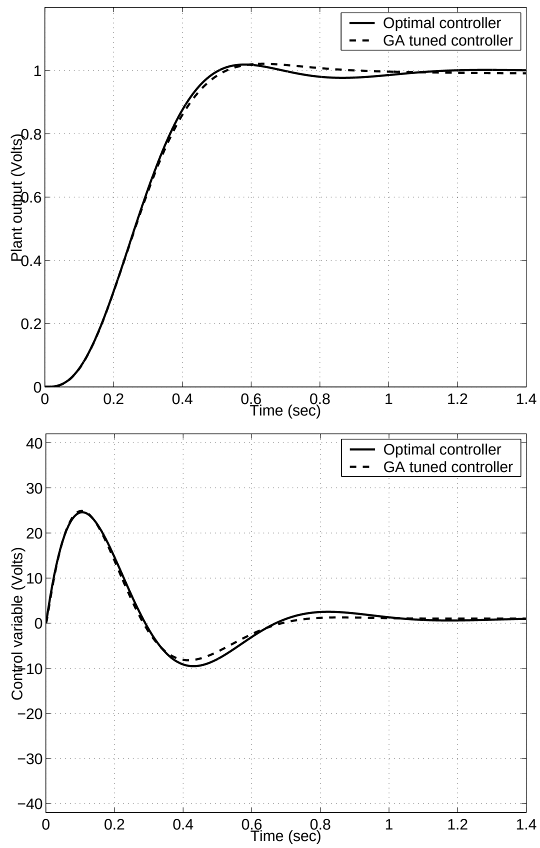

Figure 6: Plant output and Control variables of the optimal and GA controllers for the second order plant.

	Coincident poles		Distinct poles	
run	EGA	SGA	EGA	SGA
1	38.2	40.0	12.98	12.54
2	36.4	61.3	12.00	13.43
3	36.4	46.4	12.24	18.12
4	35.9	65.8	11.16	13.93
5	33.7	57.3	11.16	25.26
6	35.2	165.5	12.16	18.67
7	37.3	40.8	12.06	15.78
8	36.2	75.4	11.25	12.17
9	33.0	46.1	11.19	15.38
10	37.6	61.3	11.83	12.54
Average of 30 runs including the previous 10 runs				
	36.1	62.3	11.84	17.49
Standard deviation of ITAE indices				
	1.93	30.77	1.21	10.33

Table 3: Sum of the four ITAE values (in mVolts·sec^2) for the second order plants with coincident and distinct poles and with actuator nonlinearities

result proposed in [20] and confirmed the improvement with respect to the GP controller for the same plant proposed in [10, 11].

3.4 Second Order Plant with Distinct Poles and Actuator Nonlinearities

For the plant of equation 3, the controller was tuned to achieve nonlinear control. Two distinct poles were chosen for a=0.5 and b=1. Ten runs of both the EGA and SGA gave the ITAE values reported in table 3 (right columns). Thirty runs gave the average and standard deviation reported at the bottom of the table. The final simulation time was 0.5 sec. Figure 7 shows the plant outputs and the control variables for the best individual of a run. Note that the control variable reaches both extremes of saturation and shows a complete bang-bang control (the actuator usage is maximised and the maximum power is delivered to the plant to achieve time optimality) in both the cases where K = 1 and K = 2.

3.5 Performance of Directional Mutation and GA Process Analysis

In a further experiment, we wanted to evaluate the effect of having directional mutation in the search process. Therefore, we added localised tournaments to the SGA and we compared the solutions with those obtained with the EGA. We simulated the second order plant with coincident poles and nonlinear actuator. We verified the capability of both algorithms for delivering a minimum quality solution in a limited time of 50 generations. The termination criterion was to reach an ITAE value of 40 mVolts·sec^2 with a simulation time of 1 sec. We performed 20 runs. The EGA reached the target ITAE in all 20 runs in an average of 27 generations. The SGA with localised tournaments reached the target ITAE in 14 runs in an average of 35 generations and failed in 6 runs.

Directional mutation therefore seems to benefit the search process. To have a better insight on the effect of directional mutation we investigated the effectiveness of the new operators by considering an efficiency index. Given a class of individuals characterised by being generated by means of a certain operator, we define the efficiency index of that operator as the ratio of the percentage of selected individuals belonging to the considered class over the percentage in the whole population. If x% of a population was generated by applying the operator a, we expect a similar percentage (efficiency around 1) with that origin to be part of the selected individuals (parents for the next generation). If none or few individuals generated by the operator a are selected, we infer that the operator a did not perform well in that generation. If, on the other hand, the percentage of selected individual that were generated by the operator a is higher than the percentage of the same individuals in the previous generations, we conclude that operator a performed better than other operators and gave high quality offspring.

Figures 8 and 9 plot the efficiency of directional mutation, global directional mutation and crossover for a single run. The efficiency of operators tends to decrease at the end of the computation since elitism becomes very efficient when the offspring do not bring any improvement. Finally, we looked at the dynamic of the fitness improvement during a run for the EGA and SGA with localised tournaments. Figure 10 shows that the EGA starts slower than the SGA but eventually proceeds with a steep uphill climb whereas the SGA decreases its climbing speed.

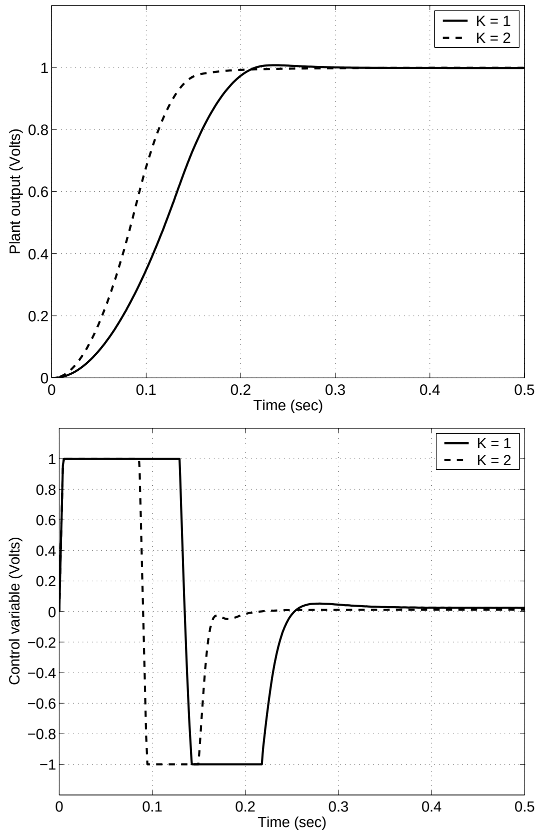

Figure 7: Plant outputs and control variables of the controller for the second order plant with distinct poles and actuator nonlinearities.

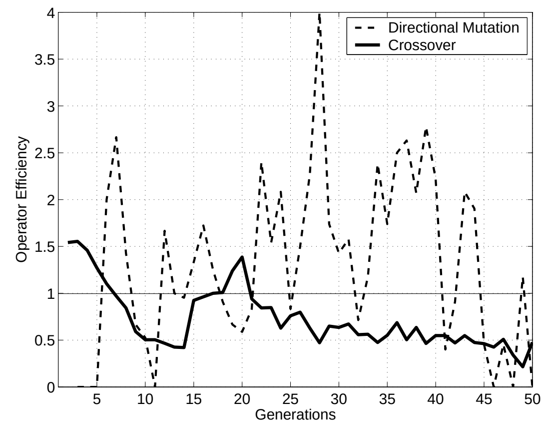

Figure 8: Efficiency of operators crossover and directional mutation after random mutation.

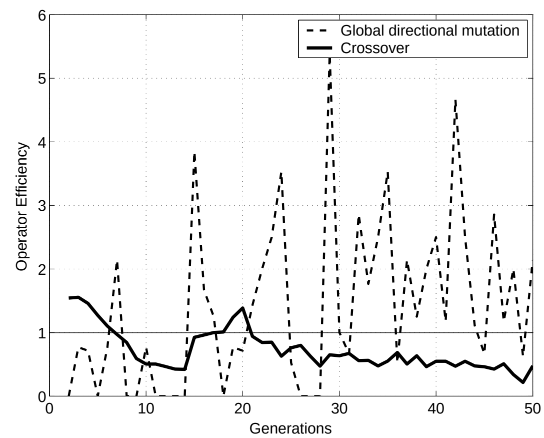

Figure 9: Efficiency of operators crossover and global directional mutation.

4. DISCUSSION AND CONCLUSION

The experiment results outlined that both the SGA and EGA obtained good solutions for the chosen control problems. Linear and nonlinear control are achieved equally without modification in the application or in the search space. Tables 1 and 3 show that the EGA and the SGA reach solutions of similar qualities in certain runs. However, the EGA has shown to steadily obtain high quality solutions with a very low variance. The SGA on the other hand, does not seem to guarantee a minimum quality outcome.

For the first order plant, the SGA recorded a best performance of ITAE = 5.80 mVolts·sec^2, very close to the best solution of the EGA (4.18 mVolts·sec^2). On the other hand, the worst performance of EGA is 5.95 mVolts·sec^2 versus 1083.41 mVolts·sec^2 for the SGA, indicating the nature of the variability in the outcomes as also highlighted by the standard deviation measurements.

The experiment for the second order linear plant was meant to provide a verification of the GA potential when trying to approximate an optimal solution from a control textbook [2]. The SGA reached optimality in some of the runs (1, 3 and 9 from table 1) while EGA reached high quality solutions in all the runs. The same consideration can be made for the nonlinear second order plants with coincident and distinct poles. The 10 runs data presented in tables 1 and 3 are a sample of a set of 30 runs executed to give statistical significance to the results. The diversity in the genotypes, even maintaining near optimal performance (see table 1 left columns), indicates that the optimal solution given by the traditional method is only one out of many possible. This suggests that a GA could be employed to explore the feasibility of controllers designed within a stricter search space that might satisfy some technical or environmental requirements.

Section 3.5 tries to give an insight into the GA process and explains how EGA obtains better results. From figures 8 and 9 we can infer that directional mutation operates in an unsteady way by performing well in some generations and very poorly in others. In particular, from figures 8 and 9 it is evident that directional mutation has efficiency 0 during certain generations (i.e. none of the offspring generated by global directional mutation are selected to become parents). In other generations, however, the operator shows to have efficiency much greater than 1. Being highly efficient in some generations, directional mutation is seen to have a very high speculative ability in driving the population towards a fruitful part of the search space.

Finally, from figure 10, we can see that the EGA is able to steadily climb the fitness landscape throughout all the 50

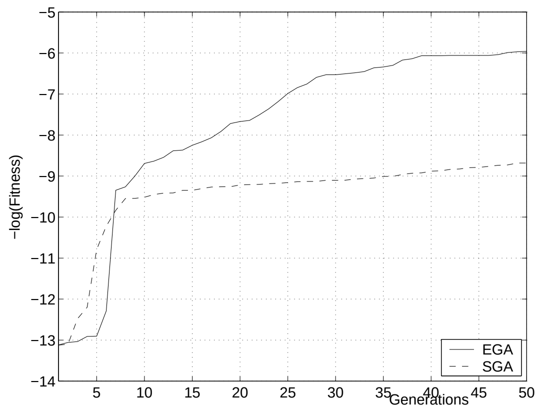

Figure 10: Average fitness for a run of the EGA and SGA.

generations while the SGA seems to settle around a local optimum and slowly improves it. A drop in population diversity seems a major cause for SGA's inability to improve the fitness after a certain number of generations. The EGA, on the other hand, works around this cause and continually improves the fitness, indicating the fact that the idea of having directional mutation is promising.

The characteristics of the results presented in this paper suggest that GAs can be designed to be both effective in the search and consistent in successive runs. Different controllers were tuned for plants with different characteristics and constraints. The proposed EGA is shown to be a promising tool in control system optimisation. The increase in reliability and efficiency of the search (see experiment with target ITAE in section 3.5) opens the possibility of using a GA for online tuning, for systems with minimum requirements in performance and for adaptive control.

5. REFERENCES

[1] K. J. Åström and T. Hägglund. *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, N.C., 1995.

[2] R. H. Bishop and R. C. Dorf. *Modern Control Systems*. Upper Saddle River. N.J.: Prentice Hall, ninth edition, 2001.

[3] P. Bolzern, R. Scattolini, and N. Schiavoni. *Fondamenti di controlli automatici*. McGraw-Hill Libri Italia, 1998.

[4] L. dos Santos Coelho, A. Rodrigues Coelho, and R. Krohling. Parameters tuning of multivariable controllers based on memetic algorithm: fundamentals and application. In *Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on*, pages 752–757, 27-30 Oct 2002.

[5] D. C. Dracopoulos and S. Kent. Genetic programming for prediction and control. *Neural Computing and Application*, 6(4):214–228, 1997.

[6] P. J. Fleming and R. C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11):1223–1241, April 2002.

[7] G. D. Gray, D. J. Murray-Smith, Y. Li, K. C. sharman, and T. Weinbrenner. Nonlinear model structure identification using genetic programming. *Control Engineering Practice*, 6:1341–1352, 1998.

[8] J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions of Systems. Man and Cybernetics*, 16(1):122–128, 1986.

[9] M. Jamshidi, L. dos Santos Coelho, R. A. Krohling, and P. J. Fleming. *Robust Control System with Genetic Algorithm*. CRC Press, London, 2002.

[10] J. R. Koza, M. A. Keane, J. Yu, F. H. Bennet III, and W. Mydlowec. Automatic creation of human-competitive programs and controllers by means of genetic programming. *Genetic Programming and Evolvable Machines*, (1):121–164, 2000.

[11] J. R. Koza, M. A. Keane, J. Yu, F. H. Bennet III, and W. Mydlowec. Method and apparatus for automatic synthesis controllers. Technical report, U.S. Patent. 6,564,194, May 2003.

[12] R. Krohling. Design of PID controller for disturbance rejection: a genetic optimization approach. In *Genetic Algorithms In Engineering Systems:Innovations And Applications, 1997. GALESIA 97. Second International Conference On (Conf. Publ. No. 446)*, pages 498–503, 2-4 Sept 1997.

[13] R. A. Krohling and J. P. Rey. Design of optimal disturbance rejection PID controllers using genetic algorithms. *IEEE Transactions of Evolutionary Computation*, 5(1):78–82, Feb 2001.

[14] W. Lennon and K. Passino. Intelligent control for brake systems. *Control Systems Technology, IEEE Transactions on*, 7(2):188–202, March 1999.

[15] G. P. Liu and S. Daley. Optimal-tuning PID control for industrial systems. *Control Engineering Practice*, 9:1185–1194, 2001.

[16] T. Morimoto and Y. Hashimoto. AI approaches to identification and control of total plant production systems. *Control Engineering Practice*, 8(5):555–567, May 2000.

[17] C. L. Nachtigal. *Instrumentation and Control, Fundamentals and application*. Wiley Interscience Publication, John Wiley and Sons, 1990.

[18] K. Ogata. *Modern Control Engineering*. Prentice Hall, Upper Saddle River, N.J., third edition, 1997.

[19] A. Soltoggio. A comparison of genetic programming and genetic algorithms in the design of a robust, saturated control system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 04)*, volume 3103 of *Lecture Notes in Computer Sciences*, pages 174–185. Springer, 2004.

[20] A. Soltoggio. GP and GA in the design of a constrained control system with disturbance rejection. In *Proceedings of the International Symposium on Intelligent Control, (ISIC 2004), 2-4 September 2004, Taipei, Taiwan*, 2004.

[21] C. Vlachos, D. Williams, and J. B. Gomm. Solution to the shell standard control problem using genetically tuned PID controllers. *Control Engineering Practice*, 10(2):151–163, February 2002.

[22] P. Wang and D. P. Kwok. Optimal design of PID process controllers based on genetic algorithms. *Control Engineering Practice*, 2(4):641–648, August 1994.