Determination of the PID Controller Parameters by Modified Genetic Algorithm for Improved Performance

AYTEKIN BAGIS
Department of Electrical Electronic Engineering
Erciyes University
38039 Kayseri, Turkey

This paper presents an efficient and fast tuning method based on a modified genetic algorithm (MGA) structure to find the optimal parameters of the proportional-integral-derivative (PID) controller so that the desired system specifications are satisfied. To achieve more effective search, the optimization algorithm is based on the integration of classical genetic algorithm structure and a systematic neighbourhood structure. To demonstrate the effectiveness of presented method, the step responses of closed loop system were compared with that of the existing methods in the literature. Simulation results indicate that the performance of the PID controlled system can be significantly improved by the MGA-based method.

Keywords: PID control, PID tuning, parameter optimization, genetic algorithm, artificial intelligence

1. INTRODUCTION

Despite the enormous interest that modern control techniques have sparked among academics during the last several decades, the PID controllers are still widely used in many industrial control systems. The simplicity and transparency of PID control mechanism, the availability of a large number of highly efficient, reliable, and cost-effective commercial PID control modules, and their acceptance from the operators are among the reasons for their popularity.

The implement a PID controller, three parameters (the proportional gain, Kp; the integral gain, Ki; the derivative gain, Kd) must be determined carefully. Many approaches have been developed to determine PID controller parameters for single input single output (SISO) systems. Among the well-known approaches are the Ziegler-Nichols (Z-N) method [1], the Cohen-Coon method [2], integral of squared time weighted error rule (ISE) [3], integral of absolute error rule (IAE) [4], internal-model-control (IMC) based method [5], gain-phase margin method [6]. Ziegler and Nichols proposed rules for tuning PID controllers based on the transient response characteristics of a given plant [1]. Zhuang and Atherton present the tuning rules based on minimizing an appropriate performance criterion for automatic tuning of optimum PID controllers [3]. Ho *et al*. are derived the simple formulas to tune the PID controller for meet gain and phase margins specifications [6]. Wang *et al*. expose an approach based on the model reduction method for designing PID controllers [7]. A model-free technique for the optimization of the parameters of a controller is given by Lequin *et al*. [8]. For industrial control systems, several PID controller design schemes are presented by Liu and Daley [9].

To enhance the capabilities of traditional PID tuning techniques, several new methods such as genetic algorithms (GAs) [10-15] and fuzzy logic [16-19] have been developed recently to tune the parameters of the PID controllers. Wang and Kwok present a pH neutralization process regulated by a PID controller with its parameters optimized using a GA [10]. The study given in [11] covers the optimization of two decoupled PID controllers for the propulsion and navigation of an oil platform supply ship using a GA. GA based multiobjective PID control approach for a linear brushless dc motor is given by Lin *et al.* [12]. In [13], a methodology for the optimal PID controller design using a modified genetic algorithm is proposed to improve the transient stability of AC-DC transmission systems after faults. Teng *et al.* apply a simple GA method in a real-time experiment on a liquid-level control system for online autotuning PID parameters [15]. Zhao *et al.* propose a rule-based scheme for gain scheduling of PID controllers for process control [16]. The proposed gain scheduling scheme in the work uses fuzzy rules and reasoning to determine the PID controller parameters. A hybrid method for parameter tuning of PID controllers is presented by Wu and Huang [17]. The approach uses different rule base for different plants, and it has the heavy computation burden of the GA.
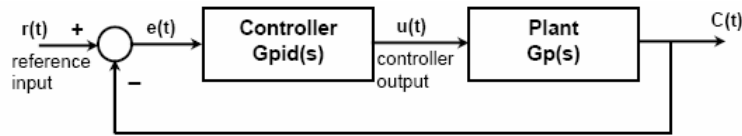

Fig. 1. Control system.

The block diagram of a simplified control system is shown in Fig. 1. In practice the output of a PID controller is given by:

$$u(t) = Kp[e(t) + \frac{1}{Ti}\int_0^t e(t)dt + Td\frac{de(t)}{dt}]. \tag{1}$$

The transfer function of a PID controller is

$$Gpid(s) = \frac{U(s)}{E(s)} = Kp + \frac{Ki}{S} + Kd, S = Kp(1 + \frac{1}{Ti\,S} + Td\,S). \tag{2}$$

In these equations, Kp is the proportional tuning constant, Ki is the integral tuning constant, Kd is the derivative tuning constant, Ti is the integral time, Td is the derivative time, and error $e(t)$ is the difference between the setpoint $r(t)$ and the process variable $c(t)$ at time t.

The process of determining the PID controller parameters Kp, Ti, and Td to achieve high and consistent performance specifications is known as controller tuning. In the design of a PID controller, these controller parameters must be optimally selected in such a way that the closed loop system has to give desired response. Moreover, the operation of the parameter determination must be performed as fast as possible for a given process.

In this paper, an efficient and fast tuning method based on a modified GA structure is proposed to find the optimal parameters of the PID controller so that the desired sys-

tem specifications are satisfied. To demonstrate the effectiveness of presented method, the step responses of closed loop system were compared with that of the existing methods in the literature. Simulation examples are given to show the effectiveness of the presented method in handling processes of different characteristics.

2. MODIFIED GENETIC ALGORITHM (MGA)

Genetic algorithms (GAs) are stochastic optimization techniques founded on the concepts of natural selection and genetics [20]. The algorithm starts with a set of solutions called population. Solutions from a population of chromosomes are used to form a new population. Once the initial population is formed, the GA creates the next generation using three main operators: (1) reproduction, (2) crossover and (3) mutation. Reproduction is the process in which the most fits chromosomes in the population receives correspondingly large number of copies in the next generation. This operation increases quality of the chromosomes in the next generation and therefore leads to better solutions of the optimization problem. The crossover operator takes two of the selected parent chromosomes and swaps parts of them at a randomly selected location. This provides a mechanism for the chromosomes to mix and match their desirable qualities in forming offspring. Mutation plays a secondary role in the GA to alter the value of a gene at a random position on the chromosome string, discovering new genetic material or restoring last material. New solutions are selected according to their fitness: the more suitable they are, the more chances they have to reproduce. This produce repeated until some condition is satisfied.

With crossover and mutation taking place, there is a high risk that the optimum solution could be lost as there is no guarantee that these operators will preserve the fittest string. To counteract this, elitism mechanism is often used. In this mechanism, the best individual from a population is saved before any of these operations take place. After the new population is formed and evaluated, it is examined to see if this best structure has been preserved. If not, the saved copy is reinserted back into the population.

Using selection, crossover, and mutation on their own will generate a large amount of different probable solutions. However, some main problems can arise. Depending on the initial population chosen, there may not be enough diversity in the initial solutions to ensure the GA searches the entire problem space. Furthermore, the GA may converge on sub-optimum solutions due to a bad choice of initial population. Moreover, inappropriate operator rates can destroy good solutions and degenerate the GA into a random search. These problems may be overcome by the introduction of an improvement mechanism into the GA.

In this paper, for the new solutions, an optimization algorithm based on the integration of classical genetic algorithm structure and a systematic neighbourhood structure is employed to achieve more effective search. The neighbourhood unit performs two main tasks: (i) receive the best solution found and obtain the neighbour solutions for this best solution, (ii) find a new best solution with more high quality if it is possible. Since this strategy suggests the solution diversity, all of the probable solutions with higher quality can be searched.

The outline of the MGA used in this paper is given below:

```
initialize population;
while predetermined termination condition not satisfied;
{
evaluate chromosomes in population;
keep the best solution;
obtain the neighbour solutions for the best solution;
compare the best solution and best neighbour solution;
keep the new best solution;
reproduction;
apply crossover and mutation to chromosomes in population;
new population = population + new best solution;
}
```

This mechanism searches the all neighbouring solution space for the existing best solution and provides to increase of genetic diversity leads to effectiveness of the algorithm. Because of the problem type and the use of neighbourhood mechanism in the presented algorithm structure, binary coded genetic algorithm was preferred. At each generation of GA search, there are a lot of neighbour chromosomes of the chromosome with high quality. When the all neighbour solutions of the existing chromosome are evaluated by the algorithm, faster convergence with finer tuning could be achieved, and a considerable improvement in the solution quality could be obtained. To overcome the local optimality problem, the use of the crossover and mutation operators is important. It is noted that the crossover and mutation operators are applied to the new population stated by population and new best solution. Thus, the probability of the search trapped in a local optimum can be considerably eliminated.

best solution:	real-coded		binary-coded (8 bits)								decoding
best solution:	2.391	→ (encoding)	0	1	1	1	0	1	1	1	
neighbour solutions			1	1	1	1	0	1	1	1	→ 4.846
			0	0	1	1	0	1	1	1	→ 1.157
			0	1	0	1	0	1	1	1	→ 1.772
			0	1	1	0	0	1	1	1	→ 2.079
			0	1	1	1	1	1	1	1	→ 2.540
			0	1	1	1	0	0	1	1	→ 2.310
			0	1	1	1	0	1	0	1	→ 2.348
			0	1	1	1	0	1	1	0	→ 2.368

Fig. 2. Neighbourhood structure for the best solution.

The neighbourhood of a solution is the set of all formations that can be arrived at by a move. The move is a process that transforms the search from the current solution to its neighbouring solution. If, after making a move, a solution is found which is better than all solutions found in prior moves than the new solution is saved as the new best solution. This neighbourhood procedure is applied to best only one solution of the each generation. Considering a chromosome with n bits, the number of the neighbour solutions is n, as depicted in Fig. 2. To obtain the neighbour solutions for the best solution, encoding and decoding operations are applied. In this work, different bit sizes are tried to achieve a

good balance between the algorithm performance and the computational cost, and we finally adopt a bit size of 8. According to simulation results, this bit size is suitable for the main process in the algorithm described in this paper.

3. PID TUNING BY USING MGA

The proposed PID tuning based on a MGA is schematically shown in Fig. 3. The major objective of the MGA program is to determine the optimal values of the PID controller parameters to improve the transient response of the system at time. To achieve this aim, the algorithm modified by a neighbourhood structure considers the minimization of an objective function. This objective function provides a means for evaluating the performance of the PID controller with the determined gain parameters in the process of the search, so that an optimized controller would be developed by the best individual.
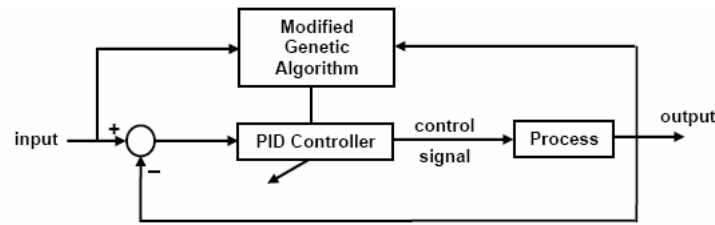

Fig. 3. PID tuning scheme with MGA.

During the optimization process, the reference input and closed loop response of the process is used by the MGA. Using the changed closed loop control performance according to the adjusted controller parameters at the each generation, the tuning algorithm searches the optimal parameters for the PID controller to satisfy the desired system specifications. To illustrate the effectiveness of the presented method, we compared the closed loop response to a step change of a number of simulated systems [16]. For PID controller problem, three different processes with different order are considered as the following:

$$G_1(s) = \frac{e^{-0.5s}}{(s+1)^2}, \ G_2(s) = \frac{4.228}{(s+0.5)(s^2+1.64s+8.456)}, \quad\quad (3a, b)$$

$$G_3(s) = \frac{27}{(s+1)(s+3)^3}. \quad\quad (3c)$$

In control system applications, the chosen performance criterion is often a weighted combination of various performance characteristics such as rise time, settling time, overshoot, and integral of the square of the error. The desired system response should have minimal settling time with a small or no overshoot in the step response of the closed loop system. Therefore, the objective function O is defined using the performance indices integral of the square of the error (ISE), the response overshoot (OS) and the 5 percent settling time ts.

$$O = \alpha(ISE) + \beta(ts) + \delta(OS) \tag{4}$$

In this function, the variables of α, β and δ are the improvement factors. By adjusting these factors, the most appropriate PID controller parameters to achieve the desirable closed loop characteristics of the system to be considered can be obtained. And, the performance of the PID controller can be significantly improved for the predetermined control objectives. For different values of the improvement factors, the main parameters of the PID controlled systems optimized by the algorithm are given in Table 1. The results in this table were found by using 20 generations. PID parameters were randomly initialized in the interval of [0.1, 5]. From this table, it can be clearly seen that to efficiently guide the search by adjusting these factors is possible. Although the number of the generation is low, these characteristics of PID controlled systems optimized by MGA are satisfactory, and are important indications of power and effectiveness of the proposed approach. In this paper, the value of ISE was chosen as the primary performance indicator. In order to clearly demonstrate the effects of the improvement factors, in a simulation study given in Table 1, the values of these factors were selected as 10, 1, and 1, respectively. On the other hand, in the final study given in Table 2, these factors were intuitively selected as 10, 3, and 1, respectively.

Table 1. Parameters of the PID controlled systems for different values of the improvement factors ($O = \alpha(ISE) + \beta(ts) + \delta(OS)$).

plant	parameter	$\alpha=10$, $\beta=1$, $\delta=1$	$\beta=10$, $\alpha=1$, $\delta=1$	$\delta=10$, $\alpha=1$, $\beta=1$
$G_1(S)$	Kp	2.483	2.060	1.714
	Ki	1.157	1.041	0.849
	Kd	1.945	1.195	0.888
	ISE	0.683	0.788	0.894
	$ts(s)$	3.715	2.024	2.833
	$OS(\%)$	0.736	0.777	0.0070
	O	11.281	21.80	3.794
$G_2(S)$	Kp	2.790	2.636	2.598
	Ki	1.791	1.330	0.868
	Kd	2.732	2.656	2.040
	ISE	0.362	0.393	0.463
	$ts(s)$	5.195	3.303	5.101
	$OS(\%)$	5.530	2.675	0.0016
	O	14.347	36.099	5.579
$G_3(S)$	Kp	2.406	2.348	2.444
	Ki	1.464	1.676	1.022
	Kd	2.694	1.560	1.329
	ISE	0.442	0.504	0.552
	$ts(s)$	2.562	1.039	3.502
	$OS(\%)$	4.313	3.927	0.0090
	O	11.295	14.827	4.144

Table 2. Simulation results.

Process	Ziegler-Nichols PID Controller		Kitamori's PID[♦] Controller [16]	
$G_1(s)$	$Kp = 2.808$ $Ti = 1.64$ $Td = 0.41$	$OS = 32\%$ $ts = 4.16$ $IAE = 1.37$ $ISE = 0.871$	$Kp = 2.212$ $Ti = 2.039$ $Td = 0.519$	$OS = 6.8\%$ $ts = 2.37$ $IAE = 1.04$ $ISE = 0.805$
$G_2(s)$	$Kp = 2.19$ $Ti = 1.03$ $Td = 0.258$	$OS = 17\%$ $ts = 5.45$ $IAE = 0.99$ $ISE = 0.526$	– – – –	– – – –
$G_3(s)$	$Kp = 3.072$ $Ti = 1.352$ $Td = 0.338$	$OS = 32.8\%$ $ts = 3.722$ $IAE = 1.13$ $ISE = 0.628$	$Kp = 2.357$ $Ti = 1.649$ $Td = 0.414$	$OS = 10.9\%$ $ts = 2.3$ $IAE = 0.833$ $ISE = 0.596$
Process	Fuzzy PID Controller [16]		MGA based PID Controller	
$G_1(s)$	$OS = 6.0\%$ $ts = 3.09$ $IAE = 1.18$ $ISE = 0.772$		$Kp = 2.391$ $Ti = 2.23$ $Td = 0.61$	$OS = 0.05\%$ $ts = 1.34$ $IAE = 0.935$ $ISE = 0.732$
$G_2(s)$	$OS = 6.1\%$ $ts = 5.01$ $IAE = 1.01$ $ISE = 0.533$		$Kp = 1.637$ $Ti = 1.697$ $Td = 0.237$	$OS = 3.4\%$ $ts = 2.89$ $IAE = 1.189$ $ISE = 0.664$
$G_3(s)$	$OS = 1.9\%$ $ts = 2.632$ $IAE = 0.811$ $ISE = 0.537$		$Kp = 1.772$ $Ti = 1.670$ $Td = 0.436$	$OS = 0.16\%$ $ts = 1.65$ $IAE = 0.95$ $ISE = 0.68$

[♦] The PID parameters of the Kitamori's controller are not available for the process $G_2(s)$. OS is the percent maximum overshoot , ts is the 5 percent settling time, and IAE, ISE are the integral of the absolute error and the integral of the squared error, respectively.

We use a population size of 20, and terminate the process after 100 generations. In the algorithm, roulette wheel selection operator and single point crossover process was employed. After several trials, the most appropriate crossover and mutation factors found was 0.8 and 0.2, respectively. Unfortunately, there is no systematic procedure for choosing the most appropriate genetic operators. Therefore, trial-and-error method was employed for the selection of the genetic operators. In order to avoid favoring a particular solution, the initial populations were generated at random within the range $0.1 \leq Kp, Ki$, $Kd \leq 5$. The binary GA search was constrained to a bit length of 8 for each variable. Since three main PID parameters (Kp, Ti, Td) are described by 24 bits, the number of neighbour solutions for the best solution at each generation is 24. The representative simulation results of the second, third, and fourth-order processes are shown in Table 2.

The values of the overshoot and the settling time for different PID controllers are obviously shown in Fig. 4. The time responses for the Ziegler-Nichols and MGA based PID controllers are also plotted in Figs. 5-7. In Table 1 and Figs. 5-7, the parameters of
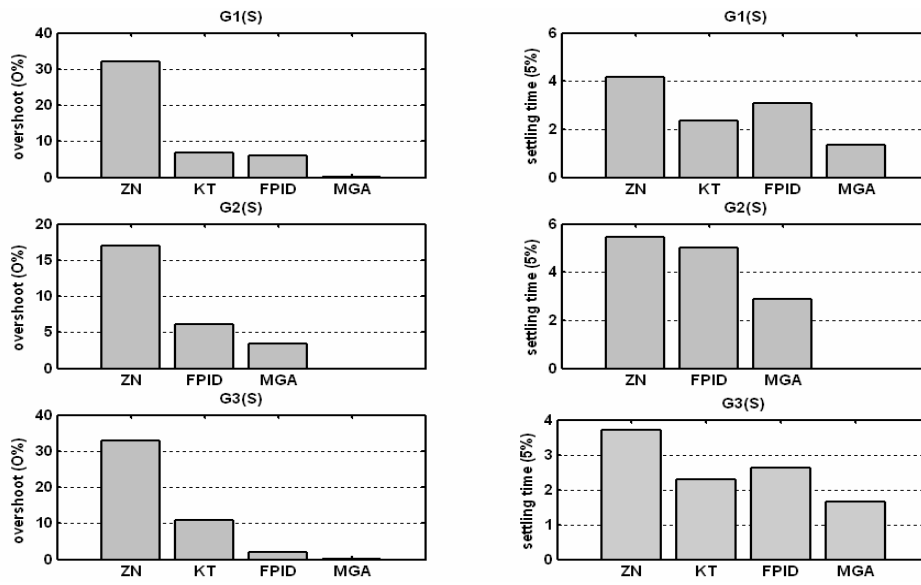
Fig. 4. Values of the overshoot and the settling time for different PID controllers.
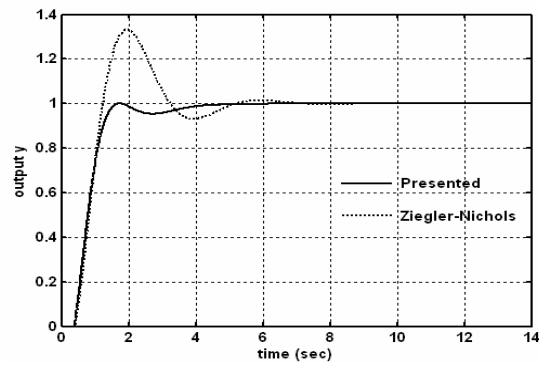

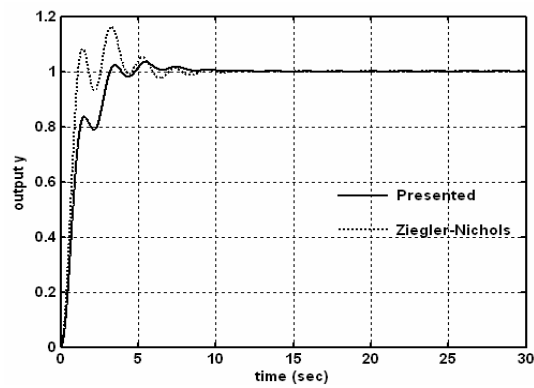Fig. 5. Comparison of step responses of the controlled second-order process.

Fig. 6. Comparison of step responses of the controlled third-order process.
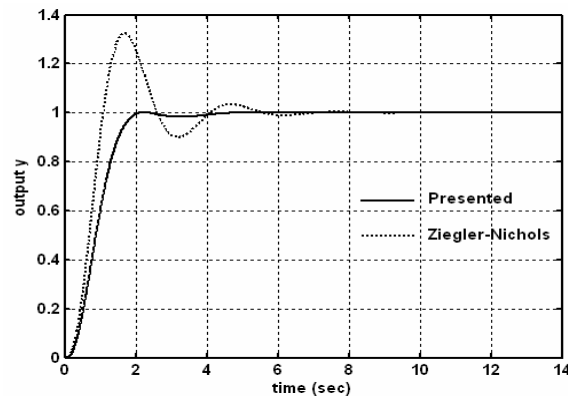
Fig. 7. Comparison of step responses of the controlled fourth-order process.

Ziegler-Nichols PID controller are determined as $Kp = 0.6\ Ku$, $Ti = 0.5\ Tu$, and $Td = 0.125\ Tu$. Where, Ku and Tu are, respectively, the gain and the period of oscillation at the stability limit under P-control.

As shown in results of Table 2, the performance of the PID controller optimized by MGA is, indeed, impressive. Performance improvements for such a problem are demonstrated by minimization of the settling time of the system's step response, and eliminate of the undesirable overshoots (Table 2 and Fig. 4). Especially, for the processes of $G_1(s)$ and $G_3(s)$, the value of the maximum overshoot is quite small, nearly 0% as shown in Fig. 4. Again, the settling time is smaller than the value of 2s for these processes.

To observe the performance of real-coded GA and binary-coded GA (MGA), the performance comparison of the GAs with different coded is given in Table 3. Simulation results show that the binary-coded GA gives better results than the real-coded GA. For $G_1(s)$, $G_2(s)$, and $G_3(s)$ processes, when the binary-coded GA is used, the values of the objective function O are 11.39, 18.71, and 11.91, respectively. On the other hand, if real-coded GA is used in the tuning procedure, these values are obtained as 13.90, 19.58, and 15.55, respectively. Especially, for $G_1(s)$ and $G_3(s)$ processes, in the step response analysis of PID controlled system based on the binary-coded GA, there is a remarkable reduction in the parameters of ts and OS.

Table 3. Performance comparison for real-coded GA and binary-coded GA.

Process	GA	Kp	Ti	Td	ts(s)	OS(%)	ISE	IAE	O
$G_1(s)$	Real-coded	2.64	3.34	0.93	5.47	5.60	0.712	1.289	13.90
	Binary-coded	2.39	2.23	0.61	1.34	0.05	0.732	0.935	11.39
$G_2(s)$	Real-coded	1.58	1.95	0.06	3.26	2.01	0.779	1.304	19.58
	Binary-coded	1.64	1.70	0.24	2.89	3.40	0.664	1.189	18.71
$G_3(s)$	Real-coded	2.14	1.78	1.06	3.01	1.68	0.484	0.959	15.55
	Binary-coded	1.77	1.67	0.44	1.65	0.16	0.680	0.950	11.91

It is clearly seen from Figs. 5-7 that the MGA based PID controller can successfully manage the process and provide better transient response under the predetermined control conditions. It would be seen from Figs. 5-7 that the presented method results are better and more desirable than the Z-N tuning method.

Fig. 8 shows the PID parameters determined by the MGA based tuning approach for controlling the fourth-order process ($G_3(s)$). In this figure, there are four main regions exhibited by the MGA, and represented by the A, B, C, and D. The parametric values of the PID controller in these regions for $G_3(S)$ are reported in Table 4. We observe that the presented method produces the good results for all regions. It is especially noted that a satisfactory performance with the good results is obtained by only four iterations. At the end of iteration 4, the value of the percent maximum overshoot and the settling time are 1.35 and 2.68, respectively. On the other hand, the value of the maximum overshoot could be reduced to 50% at first 32 iterations. And finally, in the region D formed by iteration 86, the values of the overshoot and settling time are obtained as 0.16% and 1.65, respectively.
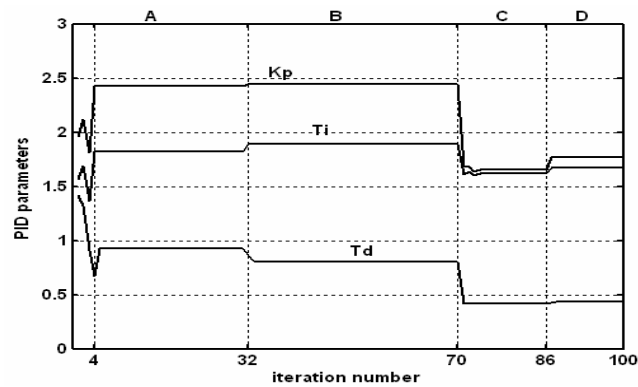

Fig. 8. Variation of the PID parameters for $G_3(s)$.

Table 4. Parametric values of the PID controller in different iteration regions for $G_3(S)$.

region	A	B	C	D
iteration	4	32	70	86
Kp	2.425	2.444	1.656	1.772
Ti	1.824	1.893	1.620	1.670
Td	0.928	0.804	0.420	0.436
OS(%)	1.35	0.69	0.37	0.16
ts(s)	2.68	2.78	1.75	1.65
ISE	0.464	0.479	0.716	0.682
O	14.03	13.82	12.79	11.91

Consequently, when compared with the other tuning methods, the presented tuning approach that is based on the use of a modified genetic algorithm structure performs an important tuning alternative for the PID controllers. Considering different tuning rules or

objective functions such as ISE, IAE, ITAE, this method can be successfully employed for improved of the transient response in a PID controlled system and/or achieved both off-line and on-line controller tuning.

4. CONCLUSION

In this paper, an efficient and effective tuning approach based on a modified genetic algorithm is presented to obtain the optimal PID controller parameters. The use of the modified genetic algorithm, in conjunction with a systematic neighbourhood structure for the tuning of the PID controller parameters, leads to a significant improvement in the dynamical response of the system under control and preserves the good performances in the existent of different process types. The simulation results indicate that the presented approach works effectively, and provides a good relation between the objective function that optimizes the PID controller and dynamic response of the system to be controlled. Hence, the presented method seems to be particularly appropriate to adopted in different process types.

REFERENCES

1. J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Transactions of American Society of Mechanical Engineers*, Vol. 64, 1942, pp. 759-768.
2. G. H. Cohen and G. A. Coon, "Theoretical investigation of retarded control," *Transactions of American Society of Mechanical Engineers*, Vol. 75, 1953, pp. 827-834.
3. M. Zhuang and D. P. Atherton, "Automatic tuning of optimum PID controllers," *IEE Proceedings on Control and Applications*, Vol. 140, 1993, pp. 216-224.
4. D. W. Pessen, "A new look at PID-controller tuning," *Journal of Dynamical Systems Measures and Control*, Vol. 116, 1994, pp. 553-557.
5. M. Morari and E. Zafiriou, *Robust Process Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.
6. W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of PID controllers based on gain and phase margin specifications," *Automatica*, Vol. 31, 1995, pp. 497-502.
7. Q. G. Wang, T. H. Lee, H. W. Fung, Q. Bi, and Y. Zhang, "PID tuning for improved performance," *IEEE Transactions on Control Systems Technology*, Vol. 7, 1999, pp. 457-465.
8. O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, and L. Triest, "Iterative feedback tuning of PID controllers comparison with classical tuning rules," *Control Engineering Practice*, Vol. 11, 2003, pp. 1023-1033.
9. G. P. Liu and S. Daley, "Optimal-tuning PID control for industrial systems," *Control Engineering Practice*, Vol. 9, 2001, pp. 1185-1194.
10. P. Wang and D.P. Kwok, "Optimal design of PID process controllers based on genetic algorithms," *Control Engineering Practice*, Vol. 2, 1994, pp. 641-648.
11. E. Alfaro-Cid, E. W. McGookin, and D. J. Murray-Smith, "GA-optimized PID and pole placement real and simulated performance when controlling the dynamics of a supply ship," *IEE Proceedings on Control Theory Applications*, Vol. 153, 2006, pp.

   228-236.

12. C. Lin, H. Jan, and N. Shieh, "GA-based multiobjective PID control for a linear brushless dc motor," *IEEE/ASME Transactions on Mechatronics*, Vol. 8, 2003, pp. 56-65.

13. Y. P. Wang, N. R. Watson, and H. H. Chong, "Modified genetic algorithm approach to design of an optimal PID controller for AC-DC transmission systems," *Electrical Power and Energy Systems*, Vol. 24, 2002, pp. 59-69.

14. R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejection PID controllers using genetic algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 5, 2001, pp. 78-82.

15. T. K. Teng, J. S. Shieh, and C. S. Chen, "Genetic algorithms applied in on-line auto-tuning PID parameters of a liquid control system," *Transactions of Institute of Measurement and Control*, Vol. 25, 2003, pp. 433-450.

16. Z. Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Transactions on Systems Man and Cybernetics*, Vol. 23, 1993, pp. 1392-1398.

17. C. J. Wu and C. H. Huang, "A hybrid method for parameter tuning of PID controllers," *Journal of the Franklin Institute*, Vol. 334B, 1997, pp. 547-562.

18. T. P. Blanchett, G. C. Kember, and R. Dubay, "PID gain scheduling using fuzzy logic," *ISA Transactions*, Vol. 39, 2000, pp. 317-325.

19. R. Bandyopadhyay, U. K. Chakraborty, and D. Patranabis, "Autotuning a PID controller: a fuzzy-genetic approach," *Journal of Systems Architecture*, Vol. 47, 2001, pp. 663-673.

20. D. T. Pham and D. Karaboga, *Intelligent Optimization Techniques: Genetic Algorithms*, *Tabu Search*, *Simulated Annealing and Neural Networks*, Springer-Verlag, 2000.

Aytekin Bagis was born in Bartin, Turkey, on October 04, 1973. He received the B.S., M.S., and Ph.D. degrees from the Department of Electronic Engineering, Erciyes University, Kayseri, Turkey in 1994, 1996, and 2001, respectively.

He has been an Assistant Professor of Electrical and Electronic Engineering Department, Erciyes University, Kayseri, Turkey, since 2002. His main research interests include intelligent control systems based on fuzzy logic, artificial intelligence optimization algorithms, fuzzy modeling and control.