

## An Adaptive Genetic Algorithm for Improving Dynamic Performance of PD Control Systems

Xing shuli, HyangRan Lee and Malrey Lee

\*561-756, Center for Advanced Image and Information Technology, School of Electronics & Information Engineering, Chon Buk National University, 664-14, 1Ga, Deokjin-Dong, Jeonju, Chon Buk, Korea  
[xingshuli600@gmail.com](mailto:xingshuli600@gmail.com), [orange1469@naver.com](mailto:orange1469@naver.com),  
[mrlee@chonbuk.ac.kr](mailto:mrlee@chonbuk.ac.kr) (corresponding author)

### Abstract

The appropriate choice of crossover and mutation rates is critical to the success of genetic algorithms. Earlier researches focused on finding optimal non-dynamic crossover or mutation rates, which vary for different problems, and different stages of the genetic process in a problem. This paper proposed an ameliorate adaptive genetic algorithm where the mutation and crossover rates are adapted dynamically based on the evaluation results of the respective offspring in the next generation. Simultaneously, we combined this new algorithm with PD controllers to improve the stability of control systems. The experimental results are compared with other researchers' approaches in this field. The PD-GA method can also significantly accelerate the system response and induce lower overshoot as well.

**Keywords:** appropriate choice; crossover and mutation rates; earlier researches; non-dynamic; ameliorate adaptive genetic algorithm; adapted dynamically; PD controllers; stability; control systems

### 1. Introduction

PD is using in many control systems, Because of the advantage of simple structure, good stability and high reliability, Proportional Integral Derivative (PID) controllers have been widely used for various industrial control systems for a long time [1]. Among the conventional PID tuning methods, the Ziegler–Nichols method [2] may be the most well-known technique. However, with the development of complexity in control objects, this method sometimes is not able to provide good tuning and tends to produce a big overshoot. Therefore, to enhance the capabilities of traditional PID tuning techniques, several intelligent approaches have been suggested to improve the PID tuning. Md Zain *et al.*, [3] applied Genetic Algorithm (GA) for optimization of PID parameters used to control a single-link flexible manipulator in vertical motion. J.H.Chen *et al.*, [4] adopted Artificial Neural Network (ANN) for online updating of PID controller parameters for non-linear process control applications. A. Visioli [5] proposed a fuzzy logic approach for tuning of PID controllers. IbChiha *et al.*, [6] presented a tuning PID method based on the multi-objective ant colony optimization.

In control engineering practice people often require control system which has faster response speed and a certain degree of damping so the gain of higher-order system often adjusted to ensure that the system has a pair of closed-loop conjugate dominant pole. We can apply the dynamic performance index of second-order system to estimate the higher-order system. Based on this situation, the amelioration of second-order systems plays a vital role. According to classical control theory, the most common way of tuning for second-order systems is not the complete PID control but just PD.

This paper presents a PD-GA method to improve the stability of control systems. This method effectively integrates the merit of each, and at the same time actuates the formulation of the problem more concisely than the PID approaches. In addition, a new dynamic adaptive genetic algorithm is put forward to possess better solution quality and efficiency than others.

The organization of this paper is as follows: Section II demonstrate the new adaptive genetic algorithm and analyzes its working principle and performance. Section III describes the adaptive GA method. Section IV shows the experiment results. Finally, the conclusions are given in Section V.

## 2. Adaptive Genetic Algorithm

Genetic Algorithms (GAs) are a kind of robust searching and optimization techniques which are finding application in a number of practical problems [7-10]. A basic genetic algorithm is composed of two processes. The first process is selection of individuals for the production of the next generation and the second process is manipulation of the selected individuals to form the next generation by crossover and mutation techniques. Therefore, the strategy of selection and the choice of crossover and mutation rates determine the success of genetic algorithms. In this paper, we focus on the operator of crossover and mutation.

### 2.1. Mutation and Crossover

In fact, the power of GAs arises primarily from crossover and mutation. Crossover causes a structure, yet randomized exchange of genetic material between solutions, with the possibility that 'good' solutions can generate 'better' ones. Crossover controls the capability of GAs in exploiting a located hill to reach the local optima. Mutation involves the modification of the value of each 'gene' of a solution. The role of Mutation has been that of restoring lost or unexplored genetic material into the population to prevent the premature convergence of the GAs to suboptimal solutions, it also controls the speed of GAs in exploring a new area. From the interpretation of crossover and mutation above, it can be deduced that the GAs are just a process of exploring and exploiting in a solution space.

Therefore, how to balance the exploring and exploiting of GAs determines their performance is superior or not, in other words, the values of crossover and mutation direct the quality of GAs. On the basis of this, Hinterding and Michalewicz classified the GAs into static, dynamic deterministic, dynamic adaptive and dynamic self-adaptive [11].

The static approach has a fixed parameter value throughout the evolution. De Jong [12] recommended the population size, the mutation rate and the crossover rate to be 100, 0.001 and 0.6 respectively. Bäck [13] formulated the results from Schaffer [14] as mutation rate =  $1.75 / (\text{population size} \times L^{1/2})$ , where L is the number of bits to represent a chromosome. The dynamic deterministic approach alters the parameter values based on some deterministic rules without using any feedback. Forgarty [15] and Hesser & Männer [16] adapted this approach to alter the mutation rate. In dynamic self-adaptive approach, the parameters are encoded in the chromosomes and these parameters are evolved together with the chromosomes. Hinterding [17] used self-adaptation to change the mutation strength of Gaussian mutation in GAs. The dynamic adaptive approach modifies the parameter values based on feedback from GAs. Davis [18], Julstrom [19] and Tuson [20] assigned the crossover rate according to the productivity of the mutation and crossover process.

## 2.2. Algorithm Scheme

In 1859 Charles Darwin proposed his famous theory: It is the environment decides the direction of the evolution of species. Correspondingly, we believe that the values of the crossover and mutation in GAs should be determined by the performance of the population. In literature this phenotype is discriminate, commonly used are the population diversity and fitness. In this paper, our intuition is to dynamically adjust the probabilities of crossover and mutation according to their productivity. The reason we do this is because through comparing the contribution of crossover and mutation to the next generation it can indirectly reflect what the phenotype of the population should be altered at that moment.

Consider the offspring generated from two parents after only one crossover operation. Employ  $CO_{off}$  be the fitness sum of the two offspring and  $CO_{par}$  represents the fitness sum of the parent. Then, the progress value of crossover  $CO$  can be derived by:

$$CO = CO_{off} - CO_{par} \quad (1)$$

For a generation experiences  $N_{co}$  crossover operations, the average crossover progress value  $\widetilde{CO}$  is:

$$\widetilde{CO} = \frac{1}{N_{co}} \sum CO \quad (2)$$

Similarly, the progress value of just undergoing one mutation operation  $M$  is:

$$M = M_{bef} - M_{aft} \quad (3)$$

And the average mutation progress value  $\widetilde{M}$  can be obtained by:

$$\widetilde{M} = \frac{1}{N_m} \sum M \quad (4)$$

Where  $M_{aft}$  is the fitness of the offspring,  $M_{bef}$  is the fitness of the original individual and  $N_m$  is the frequency of mutation in a generation.

Thus, the  $\widetilde{CO}$  and  $\widetilde{M}$  symbolizes the overall performance of crossover and mutation operator within a generation run. Before each end of generation, the probability of crossover and mutation are adjusted on the basis of these average progress values. The adjustment is executed as follows:

If  $\widetilde{CO} > \widetilde{M}$ :

$$P_c = P_c + \text{Step1}$$

$$P_m = P_m - \text{Step2}$$

If  $\widetilde{CO} < \widetilde{M}$ :

$$P_c = P_c - \text{Step1}$$

$$P_m = P_m + \text{Step2}$$

where  $P_c$ ,  $P_m$ , Step1 and Step2 represent the probability of crossover, the probability of mutation, the amount of adjustment of  $P_c$  and  $P_m$  respectively.

With regard to the values of Step1 and Step2, from the experimental results of [21, 27, 28] (Table 1), we realize that constructing a self-adaptive function of them is more appropriate than the constant values.

**Table 1. Best Fitness Obtained by Progress Rate Genetic Algorithm (PRGA) using Different Constant Step Sizes of  $\theta_1$  and  $\theta_2$  in 0/1 Knapsack Problem**

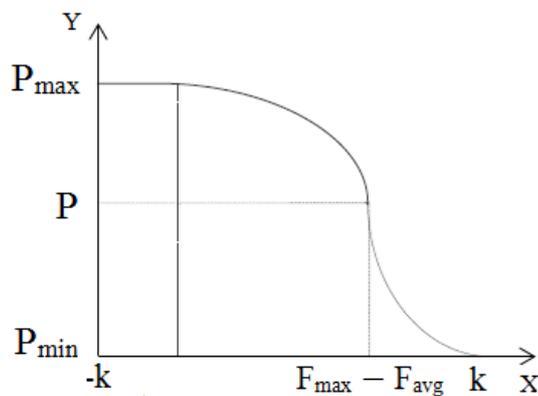
	$\theta_1= 0.01$ $\theta_2= 0.01$	$\theta_1= 0.01$ $\theta_2= 0.001$	$\theta_1= 0.001$ $\theta_2= 0.01$	$\theta_1= 0.001$ $\theta_2= 0.001$
Uncorrelated, $C = 2v$	114	89	87	78
Uncorrelated, $C = 0.5\sum W_i$	1078	983	1089	1069
Weakly, $C = 2v$	54	54	58	54
Weakly, $C = 0.5\sum W_i$	1063	962	1079	947
Strongly, $C = 2v$	60	65	65	65
Strongly, $C = 0.5\sum W_i$	1470	1435	1500	1460

Inspired by [22] and [23], the self-adapting function (refer to Figure 1) is expressed as follows:

Step1=Step2=

$$(P_{\max} - P_{\min}) \times \left\{ \frac{1}{2} + \frac{1}{\pi} \arctan \left[ k \frac{1 - 2(F_{\max} - F_{\text{avg}})}{F_{\max} - F_{\min}} \right] \right\} + P_{\min} \quad (5)$$

Where  $P_{\max}$  and  $P_{\min}$  denote the maximum and minimum step size of crossover (or mutation),  $F_{\max}$ ,  $F_{\min}$  and  $F_{\text{avg}}$  represent the largest, least and average fitness value of the population respectively.



**Figure 1. Curve of Self-Adaptive Function**

It can be seen that the graph of the function (5) has relatively smooth camber at both ends and the middle section is very close to the linear relationship. These properties of the curve on the one hand maintain the abundant diversity and converging precision in early and later period, on the other hand improve the convergence rate in mid-time [24]. In this paper,  $P_{\max}$ ,  $P_{\min}$  and  $k$  are assigned the values of 0.05, 0.005 and 10 respectively.

This kind of interaction between crossover and mutation in adapting the operator rates guarantees a better performance than that of previous schemes [22].

### 3. Adaptive GA Method

In Table 2, the dynamic performance of different higher-order systems is compared.

**Table 2. The Comparison of Dynamic Performance of Higher-Order Systems**

System Number	Transfer Function	Rise Time (s)	Settling Time (s)	Overshoot (%)
①	$1.05 / (0.125s + 1)(0.5s + 1)(s^2 + s + 1)$	1.89	4.42	8.51
②	$1.05(0.4762s + 1) / (0.125s + 1)(0.5s + 1)(s^2 + s + 1)$	1.68	3.75	8.20
③	$1.05(s + 1) / (0.125s + 1)(0.5s + 1)(s^2 + s + 1)$	1.26	3.20	8.10
④	$1.05(0.4762s + 1) / (0.25s + 1)(0.5s + 1)(s^2 + s + 1)$	1.73	4.09	8.36
⑤	$1.05(0.4762s + 1) / (0.5s + 1)(s^2 + s + 1)$	1.66	3.64	8.08
⑥	$1.05 / (s^2 + s + 1)$	1.64	3.64	8.08

The comparison results show that in higher-order systems the closed-loop zeros contribute to accelerate the system response but also increase the overshoot (which concluded from ①, ② and ③) while the closed-loop non-dominant poles provide to reduce the system exceeding, on the other hand slowdown the reaction rate (which concluded from ④ and ⑤). However, if the closed-loop poles and zeros are near each other, their effect on system will be weakened mutually (which concluded from ⑤ and ⑥).

Consequently, in the practical design of higher-order systems, we often utilize a pair of complex conjugate dominant pole to select the system parameters.

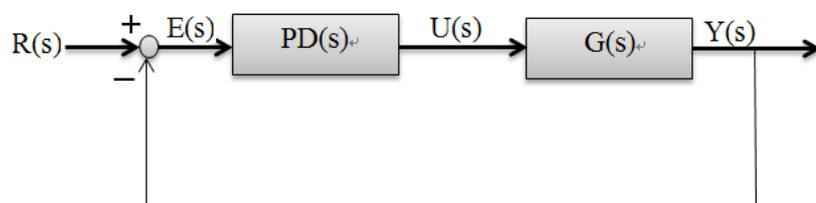
This paper adopted a typical second-order system as the adjustment object. The method is depicted as follows:

Firstly, considering the PD regulation for second-order systems (refer to Figure 2).

$$G(s) = (\omega_n^2) / (s(s + 2\varepsilon\omega_n)) \quad (6)$$

$$PD(s) = K_p + K_d s \quad (7)$$

where  $\varepsilon$  and  $\omega_n$  represent the natural frequency and damping ratio,  $K_p$  and  $K_d$  are the proportional and derivative gain of PD controller.



**Figure 2. The Adjustment of PD for Second-Order Systems**

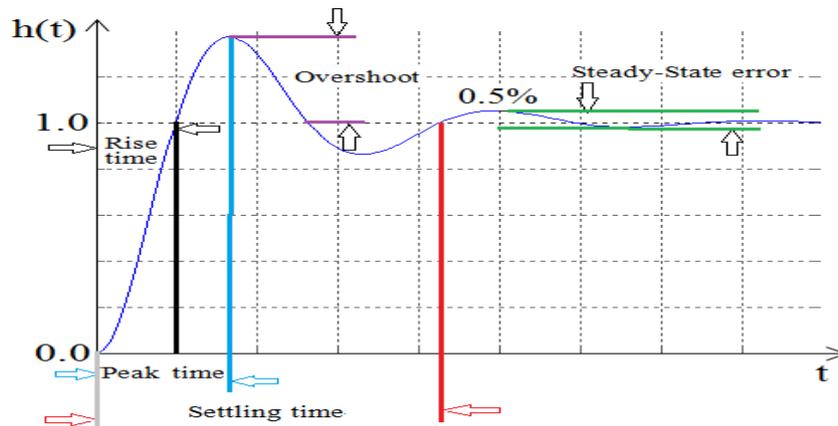
Then, the closed loop transfer function  $\Phi(s)$  can be deduced by:

$$\begin{aligned} \Phi(s) &= (PD(s)G(s)) / (1 + PD(s)G(s)) \\ &= (K_p\omega_n^2(1 + \tau s)) / (s^2 + (2\varepsilon\omega_n + \tau K_p\omega_n^2)s + K_p\omega_n^2) \end{aligned} \quad (8)$$

$$\tau = K_d / K_p \quad (9)$$

Secondly, we will use our GA to define the values of  $K_p$  and  $K_d$ . In traditional industrial control, rise time and settling time symbolizes the response speed while the overshoot

reflects the degree of damping (refer to Figure 3). Thus, the better dynamic performance of second-order systems, the less response time and overshoot is yielded.



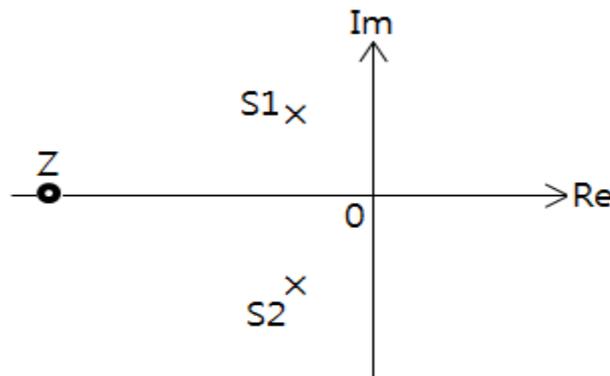
**Figure 3. The Unit Step Response of System**

As for  $\phi(s)$ , we make its poles ( $s_1, s_2$ ) and zeros ( $z$ ) (refer to Figure 4) satisfy the following conditions:

$$|z| \gg |\operatorname{Re}(s_1, s_2)| \quad (10)$$

That is:

$$\varepsilon \omega_n + (\tau K_p \omega_n^2) / 2 \ll 1 / \tau \quad (11)$$



**Figure 4. The Distribution of Poles and Zeros**

Then, the dynamic properties of  $\phi(s)$  can be estimated by the parameters of  $s_1$  and  $s_2$ . We calculate the rise time  $T_r$ , settling time  $T_s$  and overshoot  $\sigma\%$  as follows:

$$T_r = \min_{t \in (0, +\infty)} h(t) = 1 \quad (12)$$

$$T_s = \max_{t \in (0, +\infty)} h(t) = 0.005 \quad (13)$$

$$\sigma\% = (h(T_p) - 1) / 1 \times 100\% \quad (14)$$

where  $T_p$  represents the peak time.

Therefore, the fitness function (this definition method is similar like [25]) of the ameliorate adaptive genetic algorithm can be derivate by:

$$F = \min_{t \in (0, +\infty)} \omega_1(T_r + T_s) + \omega_2(\sigma \times 1s) \quad (15)$$

Exactly, in distinct industrial environments, the values of  $\omega_1$  and  $\omega_2$  are different. When we require the system has a relatively faster response speed, the value of  $\omega_1$  is often larger than  $\omega_2$ . However, if we expect the system has a smallest tolerance of error, the value of  $\omega_2$  should be larger than  $\omega_1$ . In this paper we set  $\omega_1 = \omega_2 = 1.0$ .

#### 4. Experimental Results

Suppose a second-order system whose transfer function is (Figure 5 shows the unit step of it):

$$G'(s) = 400/(s^2 + 12s + 400) \quad (16)$$

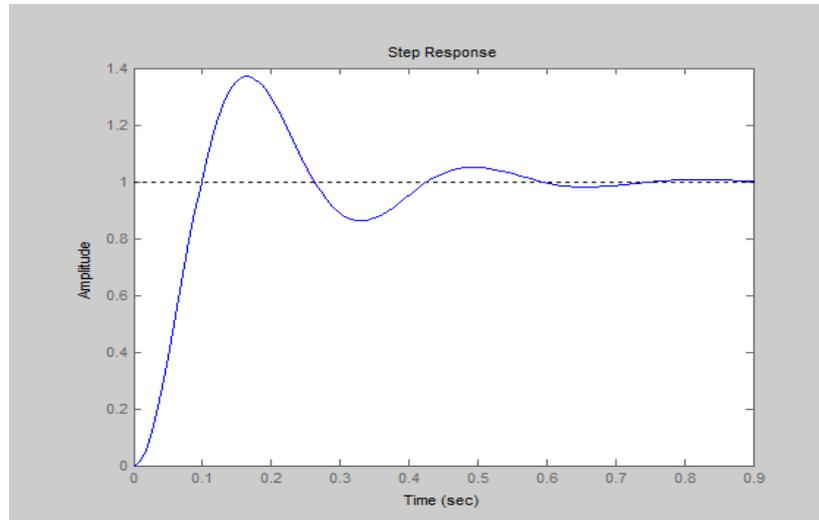


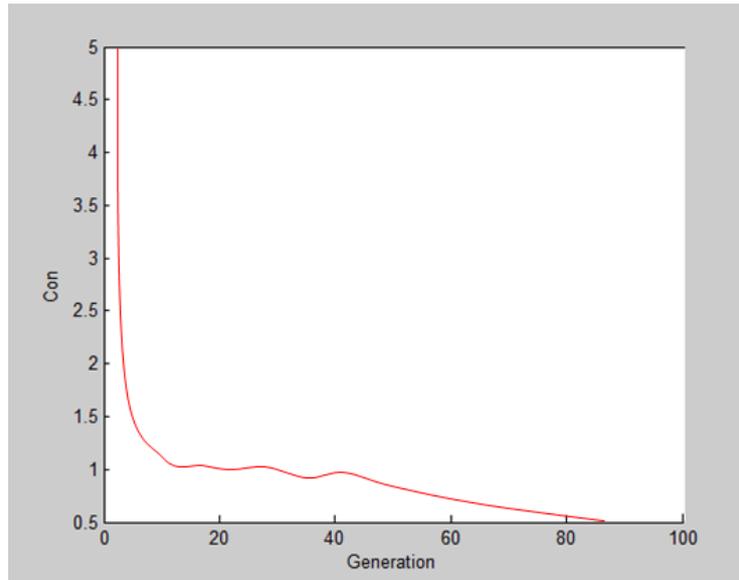
Figure 5. The Unit Step of  $G'(s)$

The employed parameters of our GA are shown in Table 3. In order to guarantee the diversity of population in the prior period, the initial crossover and mutation rate are all assigned to 0.5. Meanwhile, to discriminate the convergence of GA, a variable Con (17) is defined. When Con is equal to 0 it means the end of GA optimization process (refer to Figure 6).

$$Con = F_{\max} - F_{\text{avg}} \quad (17)$$

Table 3. The Employed Parameters in Ameliorate Adaptive GA

Parameter	Value/Type
Encoded Variables	Kp, Kd
Encoding Method	10-bit binary
Population Size	50
Maximum Generation	150
Type of Selection	Roulette Wheel Selection
Type of Crossover	Single-point Crossover
Type of Mutation	Single-point Mutation
Termination Method	Con is equal to 0



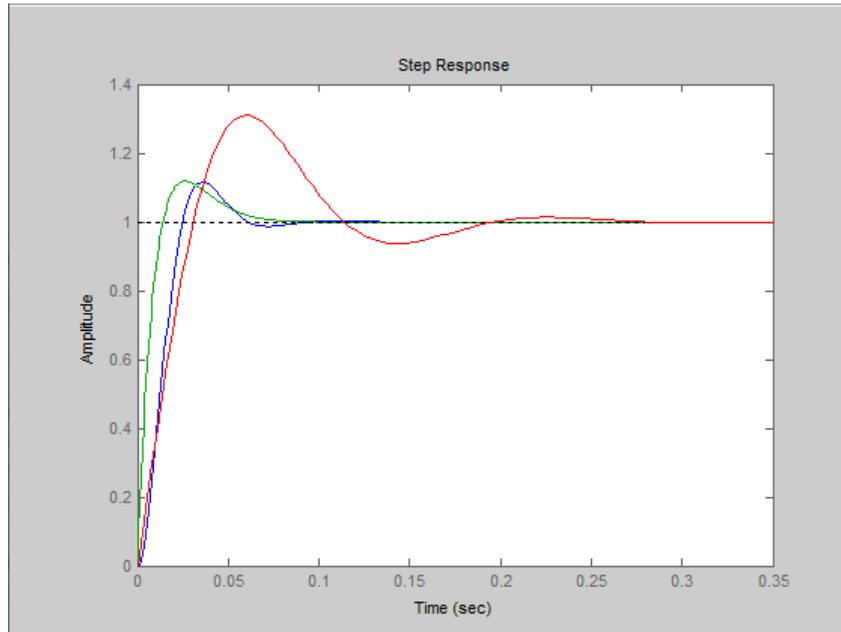
**Figure 6. The Optimization Process of GA**

Here, we also compare the simulation results of our method with that of Ziegler–Nichols tuning method and SOGACM (self-organization genetic algorithm with cyclic mutation) [26]. The Ziegler–Nichols tuning method is a heuristic method of tuning a PID controller. It was developed by John G. Ziegler and Nathaniel B. Nichols [2]. It is performed by setting the I (integral) and D (derivative) gains to zero. The SOGACM is also an adaptive GA which is proposed by introducing the dominant selection operator and the cyclic mutation operator based on standard GA. In table 4 the final optimized values of  $K_p$ ,  $K_I$  and  $K_d$  are shown.

**Table 4. The Final Optimized Values of  $K_p$ ,  $K_I$  and  $K_d$  in the Three Mentioned Methods**

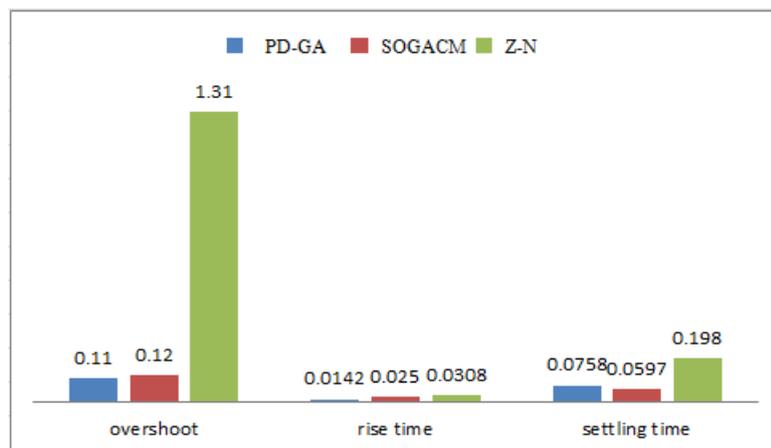
	$K_p$	$K_d$	$K_I$
Ziegler–Nichols	30.128	1.533	2.371
SOGACM	18.010	0.050	0.351
Ameliorate Adaptive GA	28.0075	0.2087	–

In Figure 7, the red curve is the step response curve of Ziegler–Nichols tuning method (Z-N). The blue curve is the step response of PID controller tuning based on the SOGACM method and the green curve is the step response curve of our new PD-GA approach.



**Figure 7. The Response Curves of System When Parameters Changed**

We notice that these three measures are all able to improve the dynamic performance of original second-order system (refer to Figure 8). However, compared with the other two methods our new approach produces the smallest overshoot and the least rise time. While the PID controllers tuning based on SOGACM possesses the shortest usage of settling time. The Ziegler-Nichols tuning method generates the worst adjusting effect among them; this phenomenon once again proves the superiority of intelligent algorithm in this field.



**Figure 8. The Dynamic Performance of the Second-Order System after Used the Three Mentioned Methods**

## 5. Conclusions

Talking about the prospect of the application, regulators that based on the intelligent algorithms will become more and more dominant in this field. The reason is not only because of their improving effectiveness is better than other approaches (such as the Ziegler-Nichols rules and Nonlinear PID controller) but also due to the convenience of setting constraints and optimization objective. In this paper, an ameliorate adaptive genetic

algorithm where the mutation and crossover rates are adapted dynamically based on the evaluation results of the respective offspring in the next generation is put forward. We combine this new algorithm with PD controllers to improve the stability of control systems. Moreover, the experimental result shows that the PD-GA method can significantly accelerate the system response and induce lower overshoot as compared to other researchers approach in this field.

With the development of the control technology, the demands of stability and accuracy on equipment have become more and more stringent. Therefore, eliminating the disturbances and noises of system will be our future work.

## Acknowledgement

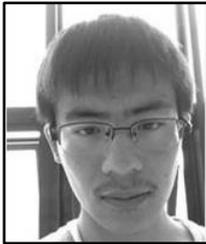
This work (Grants No:1401001175 ) was supported by Business for Academic-industrial Cooperative establishments funded Korea Small and Medium Business Administration in 2014.

## References

- [1] K. J. Åström and T. Hägglund, "Advanced PID Control, ISA-The Instrumentation, Systems, and Automation Society, (2006).
- [2] P. Vedika, D. N. Sonawane and I. Deepak, "Design and Implementation of Discrete Augmented Ziegler-Nichols PID Control", CCPE 2012, Springer-Verlag, (2012), pp. 435-441.
- [3] B. A. M. Zain, M.O. Tokhi and S.F. Toha, "PID based Control of a Single-Link Flexible Manipulator in Vertical Motion with Genetic Optimisation", in Third UKSim European Symposium on Computer Modeling and Simulation, (2009); Athens, Greece.
- [4] J. Chen and T.-C. Huang, "Applying neural networks to on-line updated PID controllers for nonlinear process control", Journal of Process Control, vol. 14, no. 2, (2004), pp. 211-230.
- [5] A. Visioli, "Tuning of PID controllers with fuzzy logic," IEE Proceedings: Control Theory and Applications, (2001).
- [6] I. Chiha, N. Liouane and P. Borne, "Multi-Objective Ant Colony Optimization to tuning PID controller", International Journal of Engineering, vol. 3, no. 2, (2009), pp. 11-16.
- [7] L. Davis (Ed.), Genetic Algorithms and Simulated Annealing, London: Pitman, (1987).
- [8] K. A. DeJong, "Genetic algorithms: A 10 year perspective", in Proceedings of an International Conference of Genetic Algorithms and their Applications, (J Greffenstette, editor), Pittsburgh, (1985).
- [9] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning. Reading", MA: Addison Wesley, (1989).
- [10] J. H. Holland, "Adapation in Natural and ArtiJicial Systems, Ann Arbor:Univ", Michigan Press, (1975).
- [11] R. Hinterding, Z. Michalewicz and A. E. Eiben, "Adaptation in Evolutionary computation: A Survey", Proceedings of the 41th IEEE International Conference on Evolutionary Computation. IEEE Press, (1997).
- [12] K. De Jong, "An Analysis of the behavior of a class of genetic adaptive systems", PhD. Thesis, University of Michigan, (1975).
- [13] T. Back, "Self-adaptation in genetic algorithms", In Proceedings of the First European Conference on Artificial Lqe, Cambridge, (1992).
- [14] J. D. Schaffer, R. Caruana, L. J. Eshelman and R. Das, "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization", J. D. Schaffer, editor, Proceedings of the 3rd International Conference on Genetic Algorithms.
- [15] T. Fogarty, "Varying the probability of mutation in the genetic algorithm", J.D. Schaffer, editor, Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann, (1989).
- [16] J. Hesser and R. Manner, "Towards an optimal mutation probability for genetic algorithms", H.-P. Schwefel and R. Manner, editors, Proceedings of the 1<sup>st</sup> Conference on Parallel Problem Solving from Nature, number 496 in Lecture Notes in Computer Science, Springer-Verlag, (1990).
- [17] R. Hinterding, "Gaussian mutation and self-adaptation in numeric genetic algorithms", IEEE International Conference on Evolutionary Computation, IEEE Press, (1995).
- [18] L. Davis, "Adapting operator probabilities in genetic algorithms", In Proceedings of the 3rd International Conference on Genetic Algorithms. Morgan Kaufmann.
- [19] B. A. Julstrom, "What have you done for me lately? adapting operator probabilities in a steady-state genetic algorithm", In Proceedings of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, (1995).
- [20] A. Tuson and P. Ross, "Cost based operator rate adaptation: An investigation", In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P Schwefel, editors, Proceedings of the 4th Conference on Parallel Problem Solving from Nature, number 1141 in Lecture Notes in Computer Science, Springer, Berlin, (1996).

- [21] W.-Y. Lin, W.-Y. Lee and T.-P. Hong, "Adapting Crossover and Mutation Rates in Genetic Algorithms (Dept. of Information Management)", I-Shou University, Journal of Information Science and Engineering, (2003).
- [22] T. P. Hong, H. S. Wang, W. Y. Lin and W. Y. Lee, "Evolution of appropriate crossover and mutation operators in a genetic process", Applied Intelligence, vol. 16, (2002), pp. 7-17.
- [23] T. Yu, J. Q. Hu, J. C. Yin and X. X. Huo, "An Adaptive Genetic Algorithm based on Arctangent Function", In Proceedings of the 26<sup>th</sup> Chinese International Conference on Control and Decision Conference, IEEE Press, (2014).
- [24] M. Bessaou and P. Slarry, "A Genetic Algorithm with Real-value Coding to Optimize Multimodal Continuous Functions", Structure Multitask Optimization, vol. 23, no. 1, (2001), pp. 63-74.
- [25] M. Tarokh and X. Zhang, "Real-time motion tracking of robot manipulators using adaptive genetic algorithms", Journal of Intelligent and Robotic Systems: Theory and Applications, (2013), pp. 1-12.
- [26] J. H. Zhang, J. Zhuang and H. F. Du, "PID controller optimization based on the self-organization genetic algorithm with cyclic mutation", Proceedings of the 6th International Conference on Artificial Intelligence, Aguascalientes, Mexico, (2008).
- [27] K.-S. Min, S.-W. Chai and M. Han, "An International Comparative Study on Cyber Security Strategy", International Journal of Security and Its Applications, vol. 9, no. 2, (2015), pp. 13-20.
- [28] Md. S. Islam, "An Algorithm for Electronic Money Transaction Security (Three Layer Security): A New Approach", International Journal of Security and Its Applications, vol. 9, no. 2, (2015), pp. 203-214.

## Authors



**Xing Shu Li**, he is a Master student in the Department of Computer Science Engineering at Chonbuk National University, South Korea. He receives his Bachelor Degree from South-central University for Nationality, China. His research interests include the Genetic Algorithm, Artificial Neural Network, Image Processing and Multi-agent Control.



**Hyangran Lee**, she Lee received a M.S. in Computer Science from the University of Kunsan National. She has been a researcher at the ChonBuk National University in Korea. She has many publications in various areas of Computer Science, concentrating on Artificial Intelligence, Robotics, Medical Healthcare and Software Engineering.



**Malrey Lee**, she received a Ph.D. in Computer Science from the University of Chung-Ang. She has been a Professor at the ChonBuk National University in Korea. She has over one hundred publications in various areas of Computer Science, concentrating on Artificial Intelligence, Robotics, Medical Healthcare and Software Engineering.

