Communications in Computational and Applied Mathematics, Vol. 1 No. 2 (2019) p. 10-15

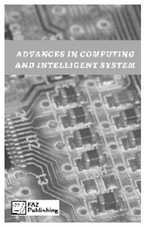
CCAM

FAZ Publishing

Communications in Computational and Applied Mathematics

Journal homepage : www.fazpublishing.com/ccam

e-ISSN : 2682-7468



Applying Genetic Algorithm to Optimize the PID controller Parameters for an Effective Automatic Voltage Regulator

Taha Aljaifi[1], **Abdallah Abdellatif**[1], **Abdullah Bawazir**[1], **Ong Pauline**[1,*], **Low Cheng Yee**[1], **Haslina Abdullah**[1]

[1]Faculty of Mechanical and Manufacturing Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author

Received 04 August 2019; Accepted 17 September 2019; Available online 16 October 2019

Abstract: Appropriate tuning of PID controller is crucial to minimize the system's steady state error and gain overshoot. This is mainly due to the control system is usually nonlinear, time-variant and sensitive to the changes in environment in practice. In this work, the genetic algorithm (GA) was applied to optimize the PID controller parameters for an effective automatic voltage regulator. The obtained results from the experimental simulation of tuning the parameters of PID controller using the GA was compared with African buffalo optimization, particle swarm optimization and linear quadratic regulator. The comparative analysis demonstrated that the GA was a good technique in solving PID controller tuning problem.

Keywords: Genetic algorithm; PID controller; automatic voltage regulator system.

1. Introduction

The automatic voltage regulator (AVR) is important to decrease the difference of system voltage below abnormal situations such as fault conditions [1]. To achieve this, the Proportional-Integral-Derivative (PID) controller is one of the widely used methods to ensure that the efficiency of the AVR system is at the optimum level. In particular, the PID controller is applied to enhance the system's adaptability to the dynamic operating environment in addition to reduce or eliminate the constant state error.

The PID controller consists of three components: the proportional component, integral component and derivative component. The proportional component is related to the recent error; the integral component calculates the total of the recent errors, while the derivative component provides the changing rate of the errors [2]. The tuning of the parameters of the PID controller has to be performed judiciously, since the control system is usually nonlinear, time-variant and sensitive to the changes in environment in practice. Hence, obtaining a set of PID controller parameters that meets the specification of a control system is of great importance.

The Ziegler-Nichols method has been widely used to fine-tune the PID controller. However, the applicability to the process that is open loop unstable is still questionable. Such limitation necessitates the need of a more effective PID controller tuning method, where the use of the metaheuristic approaches, for instance, the genetic algorithm (GA) [3], particle swarm optimization (PSO) [4], sine cosine algorithm (SCA) [5] and ant colony algorithm [6] has seen an upsurge in popularity recently.

Several investigations in the area of optimizing the PID controller of an AVR using the metaheuristic approaches have been conducted. In [5], the obtained results showed that the SCA-PID controller was more efficient than the Ziegler-Nichols, differential evolution (DE), artificial bee colony (ABC) and bio-geography-based optimization (BBO) fine-tuning method. To improve the step response of an AVR, the water wave algorithm (WWA) has been proposed in [7]. The performance comparison with PSO, SCA, bat algorithm (BA) and crow search algorithm (CSA) showed that the accuracy of the WWA was better than that of other competing algorithms. In [8], an improved kidney-inspired algorithm (IKIA) with the integration of the chaotic map has been proposed and applied to fine-tune the PID controller of an AVR. Comparative analysis with PSO, DE, ABC and BBO approaches showed that the IKIA-PID controller has a better transient response in terms of maximum overshoot percentage, settling time, rise time and peak time. The comparison between the optimizing the PID control system of an AVR using harmony search algorithm (HSA), ABC, PSO, DE, and teaching learned based optimization (TLBO) has been done in [9], where the results showed that the HSA and TLBO have higher stability and faster response than the rest. An application of symbiotic organisms search has been made in [10] to optimize the AVR system. A comparison with the ABC and BBO showed that the AVR system performance has been improved. From the literature, other metaheuristic approaches, for instance, fractal search

**Corresponding author: ongp@uthm.edu.my*

algorithm [11], ant lion optimizer [12], African buffalo optimization (ABO) [2] and cuckoo search algorithm [13] are also been applied to determine the optimal PID gains.

The objective of this study is to design and implement GA-PID controller to search for the optimal parameter for effective AVR system. The remaining of this study is organized as follows. In Section 2, the AVR is briefly introduced, and followed by the fundamental principles of GA in Section 3. Section 4 focuses on the application of GA-PID controller in tuning the AVR system, while Section 5 presents the simulation results. Lastly, conclusions are presented in Section 6.

2. Model Automatic Voltage Regulator System

An AVR is a system made to modify, control or keep the system's terminal voltage level under different load conditions. In order to calculate the dynamic efficiency and limits of stability for the parameters of the AVR system based on the various control regulations, a controller is needed such that the steady-state error of the system can be reduced.

In general, an AVR system consists of four components: generator, exciter, sensor and amplifier. The block diagram which shows the AVR transfer function without PID controller is presented in Fig. 1. The mathematical model of the transfer function of each component in linear form is discussed further in the following subsections.

2.1 Amplifier Model

The transfer function of the amplifier component, TF_A, in terms of time constant in the s domain, T_a and gain of the amplifier system, K_a is given by:

$$TF_A = \frac{K_a}{1 + sT_a} \tag{1}$$

2.2 Exciter Model

The transfer function of an exciter model, TF_E, in the representation of a time constant, T_e in the s domain, and a gain K_e is mathematically modelled as:

$$TF_E = \frac{K_e}{1 + sT_e} \tag{2}$$

2.3 Generator Model

The transfer function of a generator model, TF_G, is given by:

$$TF_G = \frac{K_g}{1 + sT_g} \tag{3}$$

where T_g is the time constant in s domain and K_g is the gain of the generator model.

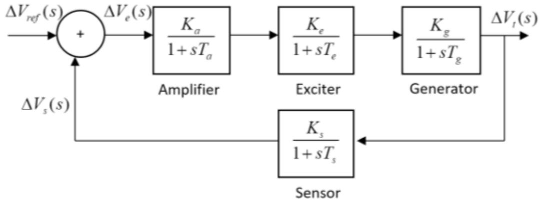


Fig. 1 – The block diagram of an AVR system without PID controller

2.4 Sensor Model

Table 1 – The parameter of all transfer functions

Model	Transfer Function	Range of Parameter
Amplifier	$TF_A = \frac{K_a}{1 + sT_a}$	$0.02 < T_a < 0.1$ $10 < K_a < 40$
Exciter	$TF_E = \frac{K_e}{1 + sT_e}$	$0.4 < T_e < 1$ $1 < K_e < 10$
Generator	$TF_G = \frac{K_g}{1 + sT_g}$	$1 < T_g < 2$ $0.7 < K_g < 1$
Sensor	$TF_S = \frac{K_s}{1 + sT_s}$	$0.001 < T_s < 0.06$ $K_s = 1$

The transfer function of a sensor model represented by a single time constant, T_s and coupled with a gain K_s, which is given by:

$$TF_S = \frac{K_s}{1 + sT_s} \tag{4}$$

For the transfer functions in Equations (1)-(4), the appropriate ranges of parameters that increases the efficiency of the components are summarized in Table 1 [2].

3. Genetic Algorithm

The fundamental principles of GA were first introduced by John Holland in 1960. The method was inspired by the biological process in which stronger individual is probably to be the winners in a competitive environment [14]. GA works on the direct analogy of such organic development to solve extremely complex problems. It considers that the potential solution of the underlying problem is an individual and could be represented by a couple of parameters [15]. These parameters are considered as genes of a chromosome and could be organized by a sequence of concatenated values. The shape of parameters illustration is described by the encoding scheme. The parameters may be displayed by binary, real numbers or other forms, depending on the program data. Its range and the search space are generally identified by the problem [16].

An illustrative flowchart of the GA implementation is as shown in Fig. 2. At first, a population of potential solutions of the underlying problem, *i.e.* the chromosome, is randomly generated. Then, the fitness value of each chromosome is evaluated in accordance with the objective function in decoded

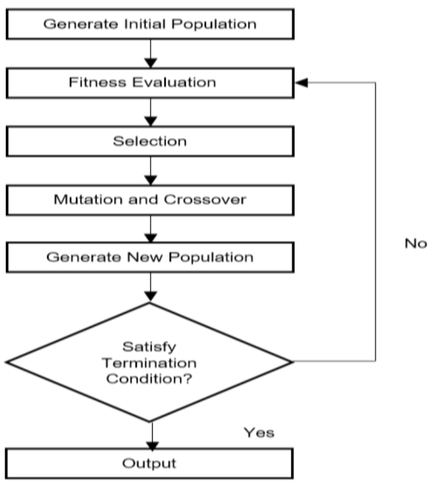


Fig. 2 – The flowchart of GA

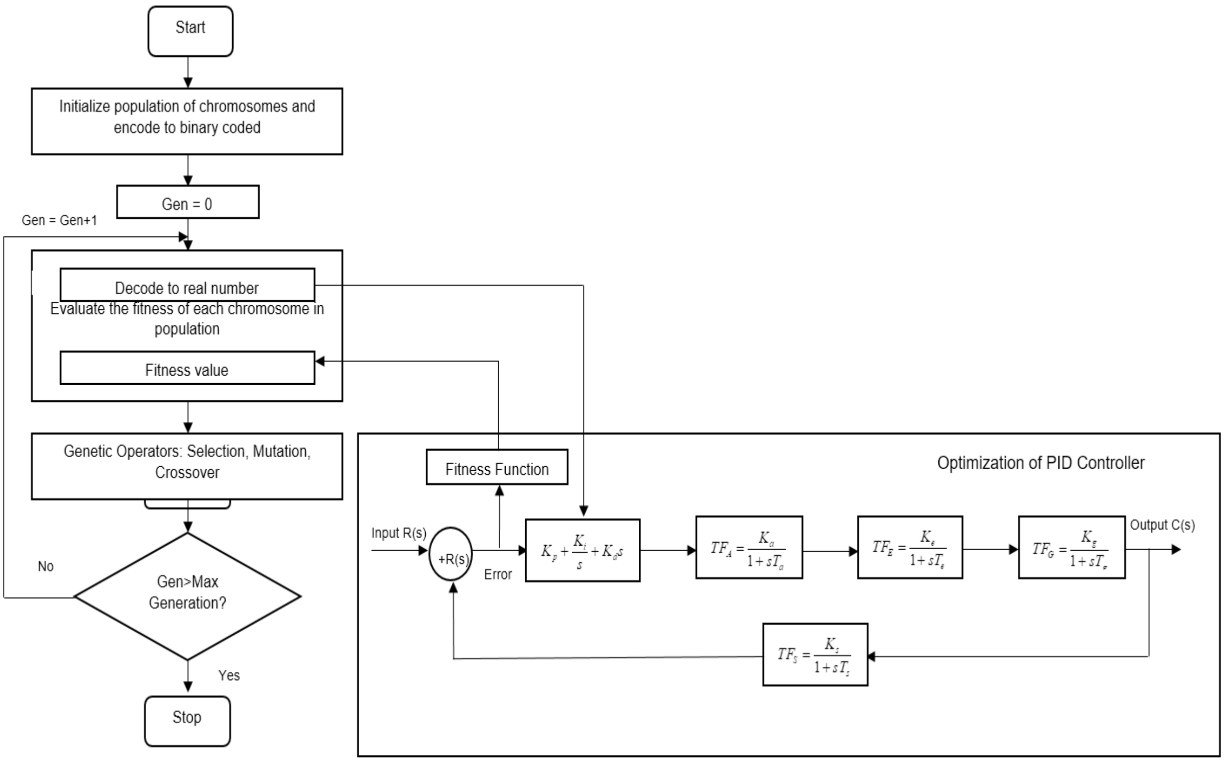


Fig. 3 – The implementation of the GA-PID controller in tuning the AVR system

form. A group of chromosomes is selected subsequently to the mating pool, and the genetic operators of mutation and crossover are used to generate the new population. The process is repeated until the population converges to the global optimum or other specified stopping conditions are met.

4. Genetic Algorithm for Tuning PID Controller of Automatic Voltage Regulator System

The dynamic feedback of an AVR can be stabilized with the utilization of the PID controller, leading to the reduction of the steady-state error. The PID controller, in general, consists of three components, namely, the proportional component (K_p), which reduces the rise time of a power system but with the limitation in eliminating the steady-state error [17]; the integral component (K_i), which compensates the limitation of K_p in removing the steady-state error for a step input, and suitable in reducing the transient response of the power system; and the derivative component (K_d), which increases the system stability by decreasing the overshoot, and hence, improving the transient resistance of the system.

In this study, the GA is used to tune the PID controller of an AVR, where the parameters used for the transfer functions of amplifier, exciter, generator and sensor models are as in Table 1. The variable V_e in Fig. 1 is given by:

$$V_e = V_t(s) - V_{ref}(s) \quad (5)$$

where V_e is the error signal, denoting the difference between the input signal, $V_{ref}(s)$ and the output signal $Vt(s)$. The error signal is propagated to the PID controller, such that the K_p, K_i, and K_d of this error signal can be calculated. The transfer function of a PID controller and the AVR system are given in Equation (6) and Equation (7), respectively.

$$K_p + \frac{K_i}{s} + K_d * s \quad (6)$$

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{(S^2K_d + SK_p + K_i)(K_AK_EK_GK_S)(1+ST_s)}{S(1+S\tau_A)(1+S\tau_E)(1+S\tau_G)(1+S\tau_s)(S^2K_d + SK_p + K_i)(K_AK_EK_GK_S)}$$

(7)

The implementation of the GA-PID controller in tuning the AVR system is summarized in Fig. 3.

5. Results and Discussion

To evaluate the performance of AVR system by applying the GA-PID controller, the uncontrolled response of an AVR is first presented. Subsequently, the GA is applied to find the optimum solution for K_p, K_i, and K_d.

5.1 Uncontrolled AVR Performance

Table 2 presents the uncontrolled AVR system response in terms of the rise time, settling time, overshoot and steady-state error. It is obviously that system performance is less satisfactory for any of the AVR systems in use. The performance of the system is too slow for handling the rapid changes in supply and demand. The overshoot percentage resulted in a 220V operating apparatus, for instance, a 365.34

Table 2 - The time performance of the uncontrolled system

Rise Time	Settling Time (2%)	Overshoot %	Steady State Error
0.259	6.98	65.5	0.089

Published by FAZ Publishing
http://www.fazpublishing.com/ccam

V will cause permanent damage to the system. Therefore, a PID controller is essential to control AVR system and improve its performance.

5.2 Controlled AVR Performance using GA-PID Controller

To improve the system response, the GA has been applied to optimize the parameters of the PID controller (K_p, K_i, and K_d). The parameters of K_p, K_i, and K_d are bounded within the range from 0 to 500. The transfer function as in Equation (7) is used. The parameters of the PID controller are summarized in Table 3. Hence, the transfer function of the AVR system is expressed as [18]:

$$\frac{\Delta V_t(s)}{\Delta V_{ref}(s)} = \frac{0.1s+10}{0.0004s^4+0.045s^3+0.555s^2+1.51s+11} = \frac{250(s+100)}{(s+98.82)(s+12.63)(s^2+1.057s+22.04)} \quad (8)$$

The simulation is repeated several times by varying the number of maximum iterations (50, 75 and 100), mutation rate (0.01, 0.015, 0.02), and crossover values (0.7, 0.75 and 0.8). The population size is assigned as 20. The results are presented in Table 4 to Table 6.

It can be observed in Table 4 and Table 5 that the GA-PID controller has a remarkable performance, with 0% gain overshoot for the experimental setting of (iteration number = 100, mutation = 0.01%, crossover = 0.7) and (iteration number = 50, mutation = 0.015%, crossover = 0.75) (See Fig. 4). Interestingly, although at the same mutation rate of 0.015% and crossover of 0.75, the GA-PID gives the worst result (0.607 sec) in terms of rise time when the iteration number increases to 100. The GA-PID is with the fastest rise time when the mutation rate and crossover are assigned to 0.02% and 0.8, respectively, and at the iteration number of 100.

The settling time of the GA-PID is the fastest (1.1 sec) when the experimental setting of (iteration number = 75, mutation = 0.01%, crossover = 0.7) is applied. On the other hand, although by maintaining the same number of iteration at 75, the system has the slowest settling time (2.43 sec) when the mutation and the crossover are assigned to 0.02% and 0.8, respectively.

It is pertinent to note that the GA-PID at the parameter setting of (iteration number = 100, mutation = 0.01%, crossover = 0.7) although does not have the fastest settling time and rise time, the obtained results are considered satisfactory with the settling time of 1.28 sec and rise time of 0.334. Hence, these results will be used to compare against other optimizers.

Table 3 – Parameters used for the PID controller

Parameter	Value
K_A	10
K_E	1.0
K_G	1.0
K_S	1.0
τ_A	0.1
τ_E	0.4
τ_G	1.0
τ_S	0.01

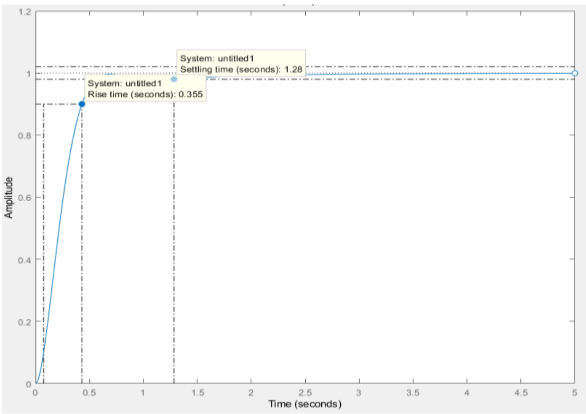


Fig. 4 – GA-PID for iteration number = 50, mutation = 0.015%, crossover = 0.75

Table 4 – GA-PID mutation 0.01%, crossover 0.7

Iteration	K_p	K_i	K_d	Rise Time (sec)	Settling Time (sec)	Over-Shoot (%)
50	0.727	0.6999	0.2933	0.231	1.99	3.46
75	0.5692	0.359	0.179	0.247	1.1	2.35
100	0.5692	0.359	0.179	0.334	1.28	0

Table 5 - GA-PID mutation 0.015 %, crossover 0.75

Iteration	K_p	K_i	K_d	Rise Time (sec)	Settling Time (sec)	Over-Shoot (%)
50	6.125	0.06575	0.1766	0.355	1.28	0
75	0.853	0.92353	0.2576	0.266	2.02	7.48
100	2.942	0.7952	0.4156	0.607	2.24	2.76

Table 6 - GA-PID mutation 0.02%, crossover 0.8

Iteration	K_p	K_i	K_d	Rise Time (sec)	Settling Time (sec)	Over-Shoot (%)
50	0.8004	0.8159	0.3141	0.358	1.78	9.96
75	0.1146	0.9239	0.7881	0.411	2.43	4.31
100	0.8356	0.9881	0.7501	0.246	2.24	2.76

5.3 Performance Comparison with Other Optimizers

In this section, the performance comparison of the GA-PID with other algorithms, specifically, ABO-PID, PSO-PID and linear quadratic regulator (LQR)-PID controllers obtained from [2] is made. The PSO-PID and LQR-PID controllers are chosen for performance comparison due to their popularity, while the ABO-PID controller is selected since it is being one of the recently developed bioinspired optimization algorithm. The parameter settings used for the PSO-PID and ABO-PID controllers are summarized in Table 7.

As shown in Table 8, the GA-PID and ABO-PID are the only controllers which have 0% steady-state error as 0% overshoot gain. Also, the performance of the PSO-PID is acceptable, with the 0% overshoot gain and 0.008% steady-state error, which is considerably small. Among all, the LQR-

Published by FAZ Publishing
http://www.fazpublishing.com/ccam

Table 7 – Parameter settings for PSO-PID, ABO-PID and LQR-PID controllers

Controller Type	Parameter
PSO-PID	Population size = 50, maximum inertia weight = 0.9, minimum inertia weight = 0.4, acceleration constants, C1 = C2 = 1.4, maximum iteration = 50
ABO-PID	Population size = 40, m.k = 1.0, bgmax/bpmax = 0.6, learning parameter = 0.5, w.k = 1.0, maximum iteration = 50

Table 8 - The comparison between different types of PID controller

Controller Type	K_p	K_i	K_d	Rise Time	Settling Time	Over-shoot	Steady State
GA-PID	0.5692	0.359	0.179	0.335	1.28	0	0
PSO-PID	3.3172	0.8993	0.2814	0.4993	10.2	0	0.008
ABO-PID	3.007	1.0724	0.4304	1.77	2.85	0	0
LQR-PID	1.01	0.5	0.1	0.5	2.335	2.44	0.02

PID is the worst in terms of overshoot gain and steady-state error.

In terms of the rise time, the GA-PID is the fastest with less than 0.4 sec, followed by LQR-PID with 0.5 sec. The worst result is given by the ABO-PID, with the rise time of 1.77 sec is required. Again, the GA-PID outperforms the others with respect to the settling time, in which 1.28 sec is recorded. This is followed by the LQR-PID with 2.3355 sec and ABO-PID with 2.85 sec. The PSO-PID is the slowest in reaching the settling time, with 10.2 sec is needed.

6. Conclusion

Optimization of the PID controller using the metaheuristic approaches are found to be more accurate than the manual optimization method. In this study, the GA is applied to optimize the PID controller of an AVR system, and subsequently compared with the PSO-PID, ABO-PID and LQR-PID. The results showed that the GA-PID is one of the techniques that gives optimal solution with 0% steady-state error and gain overshoot. In addition, the GA-PID is the fastest in terms of rise time (0.335 sec) and settling time (1.28 sec), in comparison with others.

For future work, further investigation of applying more recent developed bioinspired optimizers into the tuning of AVR can be conducted, such that the efficiency of AVR tuning can be increased.

References

[1] Gozde, H., M. Taplamacioglu, & I. Kocaarslan (2010). Application of artificial bee colony algorithm in an automatic voltage regulator (AVR) system. International Journal on Technical and Physical Problems of Engineering, 1(3), 88-92.

[2] Odili, J.B., M.N.M. Kahar, & A. Noraziah (2017). Parameters-tuning of PID controller for automatic voltage regulators using the African buffalo optimization. PloS one, 12(4), e0175901.

[3] Perng, J.W. & S.C. Hsieh (2019). Design of Digital PID Control Systems Based on Sensitivity Analysis and Genetic Algorithms. International Journal of Control, Automation and Systems, 17(7), 1838-1846.

[4] Xiang, Z., D. Ji, H. Zhang, H. Wu, & Y. Li (2019). A simple PID-based strategy for particle swarm optimization algorithm. Information Sciences, 502, 558-574.

[5] Hekimoğlu, B. (2019). Sine-cosine algorithm-based optimization for automatic voltage regulator system. Transactions of the Institute of Measurement and Control, 41(6), 1761-1771.

[6] Aabid, M., A. Elakkary, & N. Sefiani (2017). PID parameters optimization using ant-colony algorithm for human heart control. International Review on Modelling and Simulations, 10(2), 94-102.

[7] Zhou, Y., J. Zhang, X. Yang, & Y. Ling (2019). Optimization of PID controller based on water wave optimization for an automatic voltage regulator system. Information Technology and Control, 48(1), 160-171.

[8] Ekinci, S. & B. Hekimoglu (2019). Improved Kidney-Inspired Algorithm Approach for Tuning of PID Controller in AVR System. IEEE Access, 7, 39935-39947.

[9] Çelik, E. & N. Öztürk (2018). A hybrid symbiotic organisms search and simulated annealing technique applied to efficient design of PID controller for automatic voltage regulator. Soft Computing, 22(23), 8011-8024.

[10] Çelik, E. & R. Durgut (2018). Performance enhancement of automatic voltage regulator by modified cost function and symbiotic organisms search algorithm. Engineering Science and Technology, an International Journal, 21(5), 1104-1111.

[11] Çelik, E. (2018). Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system. Neural Computing and Applications, 30(6), 1991-2002.

[12] Pradhan, R., S.K. Majhi, & B.B. Pati (2018). Design of PID controller for automatic voltage regulator system using Ant Lion Optimizer. World Journal of Engineering, 15(3), 373-387.

[13] Bingul, Z. & O. Karahan (2018). A novel performance criterion approach to optimum design of PID controller using cuckoo search algorithm for AVR system. Journal of the Franklin Institute, 355(13), 5534-5559.

[14] Sivanandam, S. & S. Deepa, *Genetic algorithm optimization problems*, in *Introduction to genetic algorithms*. 2008, Springer. p. 165-209.

[15] Bajpai, P. & M. Kumar (2010). Genetic algorithm–an approach to solve global optimization problems. Indian Journal of computer science and engineering, 1(3), 199-206.

[16] Oyama, A., S. Obayashi, & K. Nakahashi (2000). Real-coded adaptive range genetic algorithm and its application to aerodynamic design. JSME International Journal Series A Solid Mechanics and Material Engineering, 43(2), 124-129.

[17] dos Santos Coelho, L. (2009). Tuning of PID controller for an automatic regulator voltage system

using chaotic optimization approach. Chaos, Solitons & Fractals, 39(4), 1504-1514.

[18] Chatterjee, S. & V. Mukherjee (2016). PID controller for automatic voltage regulator using teaching–learning based optimization technique. International Journal of Electrical Power & Energy Systems, 77, 418-429.