# Tuning PID Controllers Using Artificial Intelligence Techniques Applied To DC-Motor and AVR System

A. Salem[1], M. A. Moustafa Hassan[2], M. E. Ammar[3]

[1] Shams Industry Company (HISENSE)
3rd Industrial Zone, 6 October (Giza, Egypt)
*Ahmedsdcs {at} gmail.com*

[2] Department of Electrical Power and Machines
Faculty of Engineering
Cairo University (Giza, Egypt)
*Mmustafa_98 {at} hotmail.com*

[3]Department of Electrical Power and Machines
Faculty of Engineering
Cairo University (Giza, Egypt)
*mohammed.ammar {at} eng.cu.edu.eg*

---

**ABSTRACT—** ***This paper investigates PID controller tuning using genetic algorithm, modified genetic algorithm and particle swarm optimization techniques. The proposed techniques are compared to PID controllers tuned by the Ziegler-Nichols technique. Closed-loop simulations are conducted using MATLAB and the genetic algorithm toolbox for two applications, a DC-Motor and an Automatic Voltage Regulator (AVR). The overshoots, rise time and settling time with the proposed techniques are shown to be better than those of the conventionally tuned PID controllers.***

**Keywords—** PID-controller, DC-Motor, AVR system, Genetic Algorithm, Particle Swarm Optimization.

---

## 1. INTRODUCTION

The Proportional-Integral-Derivative (PID) controller is one of the earliest control techniques and is still widely used in the industry due to ease of implementation, robust performance and physical principle simplicity. A PID control is a linear control methodology with a very simple control structure. This type of controller operates directly on the error signal, which is the difference between the desired output and the actual output and generates the actuation signal that drives the plant [3]. In order to achieve appropriate closed loop performance, three parameters of the PID controller must be tuned [1]. Tuning methods of PID parameters are classified as traditional and artificial intelligence (AI) methods. Conventional methods such as Ziegler-Nichols method do not provide optimal PID tuning parameters and usually results in closed-loops responses characterized by oscillations and a large overshoot [2]. Artificial intelligence approaches such as genetic algorithm and particle swarm optimization have been applied successfully to solve the optimization problem of tuning PID controller parameters for performance. This article is organized as follows: A brief overview of the PID controller algorithm is given in section (2). The artificial intelligence techniques adopted in this work are discussed in section (3). In section (4), the performance indices for the optimization problems are presented. The AI tuning techniques are validated by MATLAB simulations of a DC motor and an AVR system in section (5).

## 2. PID CONTROLLER

There are several parameters that most process control systems aim to control. These include the rise time (the time required for the controlled parameters to go from 10 to 90% of the final desired values), settling time (the time required for the transient's damped oscillations to reach and stay within ±2% of the steady-state value) and the maximum overshoot (the maximum amount that the controlled variables overshoot the desired values). The PID control signal is given by equation (1) [4]:

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}) \qquad (1)$$

Where $u(t)$ is the control signal and $e(t)$ is the error, which is difference between the desired set point and the measured process variable. The controller parameters consists of the proportional gain ($K_p$), the integral time ($T_i$) and the derivative time ($T_d$) [4, 5].

In the design of PID controller the amount of integral $K_i = K_p/T_i$ is identified to reach to an intended error in steady state. As shown in Figure 1, the PID controller has three basic terms: proportional action, in which the actuation signal is proportional to the error signal, integral action, where the actuation signal is proportional to the time integral of the error signal and derivative action $K_d = K_p T_d$ , where the actuation signal is proportional to time derivative of error signal. The values of the three parameters ($K_p$, $K_i$ and $K_d$) must to be adjusted so that the control input provides acceptable performance.

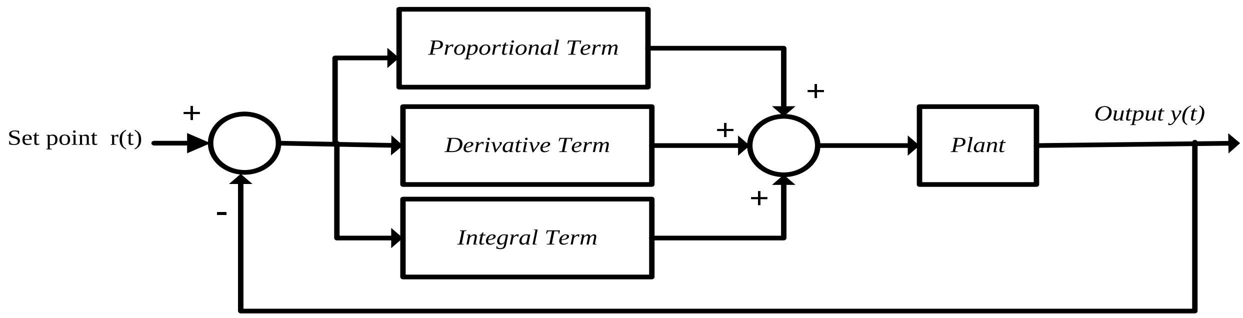


**Figure 1:** Closed loop PID controlled system

Tuning of the PID controller involves choosing the $K_p$, $K_i$ and $K_d$ that provide satisfactory closed loop performance. These parameters must be selected so that the characteristics: response speed, settling time and proper overshoot rate, all of which guarantee the system stability and performance, would be satisfied. The main method for this purpose is based on trial and error, which is time consuming. There are different processes for different composition of proportional, integral and differential gains. The role of control engineering is to adjust the gains to attain the error reduction and dynamic responses simultaneously. The transfer function of PID controller is defined as follows in equation (2) [2]:

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \qquad (2)$$

## 3. ARTIFICIAL INTELLIGENCE OPTIMIZATION TECHNIQUES

### *3.1 GENETIC ALGORITHMS (GA)*

Genetic Algorithms (GA's) are stochastic global search methods that simulate the process of natural evolution resulting in a family of computational models inspired by evolution. The genetic algorithm starts with no knowledge of the correct solution and depends entirely on responses from its environment and evolution operators (i.e. reproduction, crossover and mutation) to arrive at the best solution. By starting at several independent points and searching in parallel, the algorithm avoids local minima and converges to sub-optimal solutions. In this way, GA's have been shown to be capable of locating high performance areas in complex domains without experiencing the difficulties associated with high dimensionality, as may occur with gradient decent techniques or methods that rely on derivative information [6, 7]. Figure (2) shows the steps of creating and implementing the genetic algorithm.

The sequence can be illustrated in steps as follows:

***Step 1***: Initialize a population of individuals where each individual represents a potential solution to the problem at hand.
***Step 2:*** Apply a fitness function to evaluate the quality of each solution.
***Step 3:*** The selection process is applied in iterations to form a new population. The selection process is biased toward the fitter individuals to ensure that they will be part of the new population.
***Step 4:*** Individuals are altered using evolutionary operators. The two most frequently used evolutionary operators are mutation and crossover where:

- *Mutation:* Mutation introduces diversity to the population by introducing new genes into the genetic pool. During mutation individual agents undergo small random changes that lead to the generation of new individuals. This assists in reducing the possibility of agents being trapped within local optima.
- *Crossover (or Recombination):* This process is synonymous to mating. During crossover two individual agents combine to produce an offspring. The main objective of crossover is to explore new areas within the search space.

***Step 5:*** The above-mentioned steps are repeated until the swarm converges to an optimal or sub-optimal solution [8].
The genetic Algorithm advantages are [9]:

- Optimization with continuous or discrete variables.
- Derivative information is not required.
- Simultaneously searching a wide sampling of the cost surface.
- Dealing with a large number of variables.

- Fitting for parallel computers.
- Optimization of the variables with extremely complex cost surfaces (they can jump out of a local minimum).
- Supplying a list of optimum variables, not just a single solution.
- Encoding the variables so that the optimization is done with the encoded variables.
- Working with numerically generated data, experimental data, or analytical functions.

These advantages are exciting and produce better results when traditional optimization approaches fail.

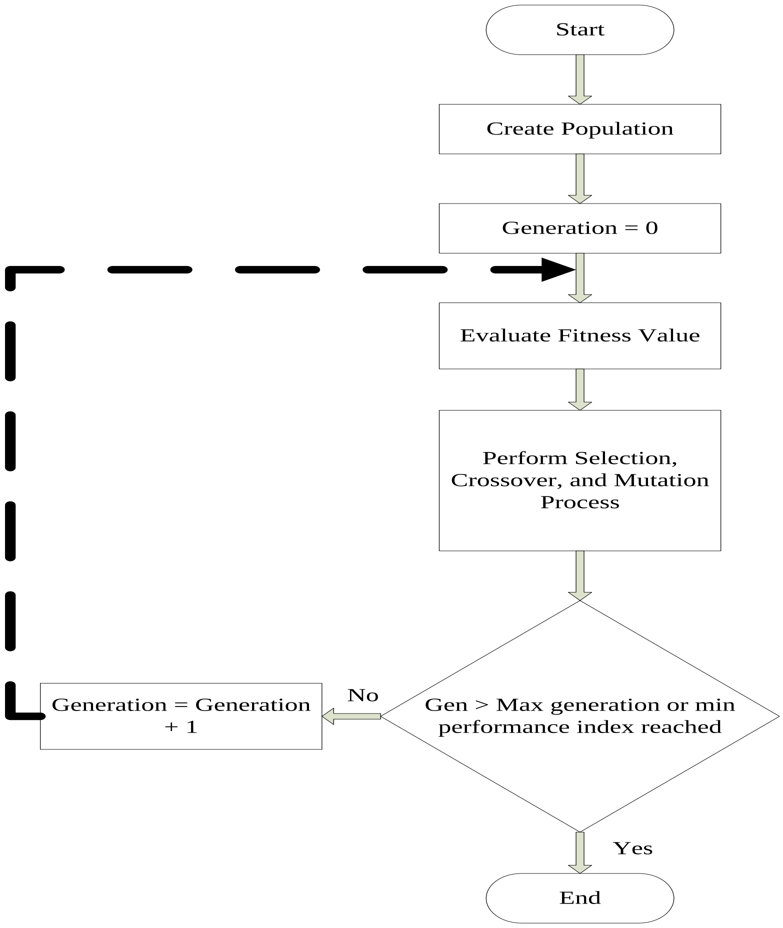


**Figure 2:** Genetic Algorithm Flowchart

### *3.2 PARTICLE SWARM OPTIMIZATION (PSO)*

Particle Swarm Optimization (PSO) was inspired from the social behavior of bird flocking. Computer algorithms simulate the complicated flocking behavior of birds [8]. In a PSO system, a swarm of individuals (called particles or intelligent agents) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its entire population [10]. The best position obtained is referred to as the global best particle. The performance of each particle is measured using a fitness function that varies depending on the optimization problem. Modification of the particles position is realized by the position and velocity information according (3) and (4) respectively [11, 12]:

$$v_i^{k+1} = v_i^k + c_1 rand_1 \times \left(pbest_i - x_i^k\right) + c_2 rand_2 \times \left(gbest - x_i^k\right) \quad (3)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (4)$$

Where:

$v_i^k$: current velocity of particle $i$ at iteration $k$,

$v_i^{k+1}$: new velocity of particle $i$ next at iteration $k$ +1,

$c_1$: cognitive acceleration constant (self-confidence),

$c_2$: social acceleration constant (swarm confidence),
$x_i^k$: current position of particle $i$ at iteration $k$,
$x_i^{k+1}$: new position of particle $i$ at next iteration $k$+1 ,
$pbest_i$: personal best of particle $i$ ,
$gbest$: Global best of the population.

Figure (3) illustrates the general flowchart for the PSO technique. The sequence can be described as follows [13]:

***Step 1:*** Generation of initial conditions of each particle. Initial searching points ($x_i^0$) and the velocities ($v_i^0$) of each particle are usually generated randomly within the allowable range. The current searching point is set to $pbest$ for each particle. The best evaluated value of $pbest$ is set to $gbest$ and the particle number with the best value is stored.

***Step 2:*** Evaluation of searching point of each particle. The objective function is calculated for each particle. If the value is better than the current $pbest$ value of the particle, then $pbest$ is replaced by the current value. If the best value of $pbest$ is better than the current $gbest$, the $gbest$ value is replaced by the best value and the particle number with the best value is stored.

***Step 3:*** Modification of each searching point.
The Particle Swarm Optimization (PSO) differs from The Genetic Algorithm (GA) in the following:

- PSO is generally faster, more robust and performs better than GA's especially when the dimension of the problem increases.
- PSO performance is insensitive to the population size (however, the population size should not be too small). PSO with smaller swarm sizes perform comparably better than GA's having larger populations [8].

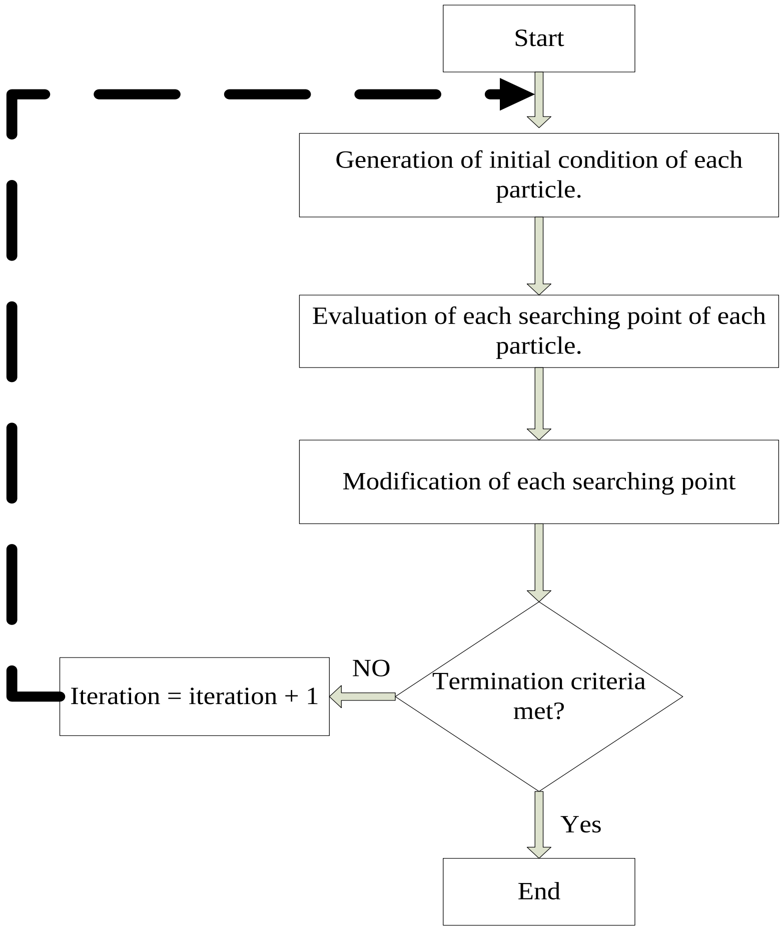


**Figure 3:** The Flowchart of Particle Swarm Optimization [14].

## 4. PERFORMANCE INDICES

The performance index (objective function) is used to evaluate the system's performance, whether the aim is to improve the design of a system or to design a control system.

*a) Integral of the Square of the Error (ISE):*

ISE is determined by equation (5); the squared error is mathematically convenient for analytical and computational purposes.

$$ISE = \int_0^T e^2(t)dt \quad (5)$$

*a) Integral of the Absolute Magnitude of the Error (IAE):*

IAE gets the absolute value of the error to remove negative error components. IAE is particularly useful for computer simulations studies, calculated by equation (6).

$$IAE = \int_0^T |e(t)|\, dt \quad (6)$$

a) *Integral of the Time Absolute Magnitude of the Error (ITAE):*

ITAE weighs the error with time and hence emphasizes the error values later on in the response rather than the initial large errors as shown in equation (7) [15, 16].

$$ITAE = \int_0^T t|e(t)|\, dt \quad (7)$$

In Modified Genetic Algorithm (MGA) the objective function is illustrated as in equation (8).

$$O = \propto (Performance\ Index) + \beta(t_s) + \delta(os) \quad (8)$$

The variables α, β and δ are the improvement factors. By adjusting these factors, the PID controller parameters to achieve the desirable closed loop characteristics of the system can be obtained. The performance of the PID controller can be significantly improved for the predetermined control objectives [17].

## 5. APPLICATIONS

This article applies the artificial intelligence tuning methods to two models; DC motor and an Automatic Voltage Regulator (AVR). The performance of the closed-loops controlled by a PID tuned using the proposed techniques are compared to loops tuned with Ziegler-Nichols method. The closed-loop simulations were run on MATLAB (*MATLAB version 8.0.0.783*) where the Genetic Algorithm (GA), Modified Generic Algorithm (MGA) and Particle Swarm Optimization (PSO) were implemented using the genetic algorithm toolbox. Figure (4) illustrates the block diagram of tuning PID-controller using the Genetic Algorithm (GA), the Modified Genetic Algorithm (MGA) and the Particle Swarm Optimization (PSO) for both the DC-Motor and the Automatic Voltage Regulator (AVR) in closed-loop.

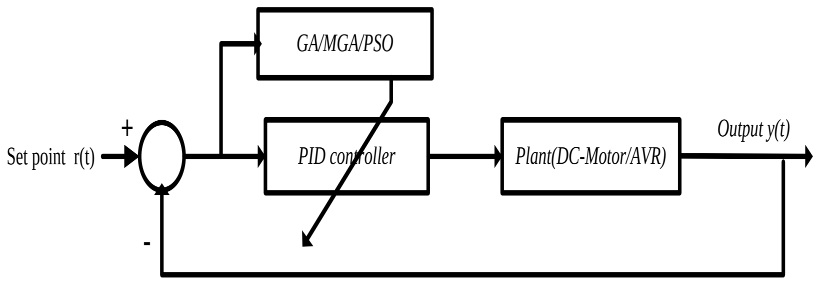


**Figure 4:** Tuning PID Controller Based on GA, MGA and PSO.

### *5.1 POSITION CONTROL OF DC-MOTOR*

A simulation is run with the DC-Motor that has the following specifications:

2hp, 230 v, 8.5 amperes, 1500 rpm
$R_a(Armature\ Resistance) = 2.45\ \Omega$,
$L_a(Armature\ Inductance) = 0.035\ H$,

$K_b(Back\ EMF) = 1.2\ Vs/rad$,
$J_m(Moment\ of\ Inertia) = 0.022\ kgm^2$,
$B_m(Frictional\ Constant) = 0.5 \times 10^{-3}$ $(N_m S/rad)$.

The transfer function of DC-Motor is given by [18, 19]:

$$\frac{\theta(s)}{V_a(s)} = \frac{1.2}{0.00077s^3 + 0.0539s^2 + 1.441s} \quad (9)$$

The step responses of DC-Motor controlled by a PI-Controller and PID-Controller for the proposed artificial intelligence tuning methods versus the Ziegler-Nichols tuning are shown in figures (5) and (6) respectively. Tables (1) and (2) present the controller parameters, maximum overshoot, rise time and settling time for the tuning techniques investigated in this work for the PI controller and PID controller cases respectively.

**TABLE 1:** DC-Motor in Closed-Loop Controlled Using a PI-Controller

| Tuning Method | ZN | GA | MGA | PSO |
|---|---|---|---|---|
| $K_p$ | 37.8 | 6.94 | 10.589 | 11.4274 |
| $K_i$ | 151.2 | 0.011 | 0.012 | 0.002 |
| $Max.OS$ | 63 | 0.0273 | 0.0128 | 0.00184 |
| $t_r$(sec) | 0.0423 | 0.291 | 0.157 | 0.14 |
| $t_s$(sec) | 0.752 | 0.547 | 0.288 | 0.246 |

The results given in table (1) show that the maximum overshoot using proposed Particle Swarm Optimization (PSO) is reduced by 85% from the closest best result in Modified Genetic Algorithm (MGA). The rise time is reduced by 10% while the settling time is reduced by 14%. This indicates that the performance of DC-Motor is improved significantly. The results in table (2) show that the performance of the DC-Motor closed-loop with the PID controller is improved using the proposed Particle Swarm Optimization (PSO).

**TABLE 2:** DC-Motor in Closed-Loop Controlled Using a PID-Controller

| Tuning Method | ZN | GA | MGA | PSO |
|---|---|---|---|---|
| $K_p$ | 49.41 | 14.859 | 19.873 | 32.6051 |
| $K_i$ | 329.4 | 0.01 | 0.009 | 0.0125 |
| $K_d$ | 1.852 | 3.589 | 3.221 | 1.0441 |
| $Max.OS$ | 16.9 | 0.00542 | 0.00273 | 0.00141 |
| $t_r$(sec) | 0.0298 | 0.0258 | 0.184 | 0.08 |
| $t_s$(sec) | 0.355 | 0.771 | 0.541 | 0.0981 |

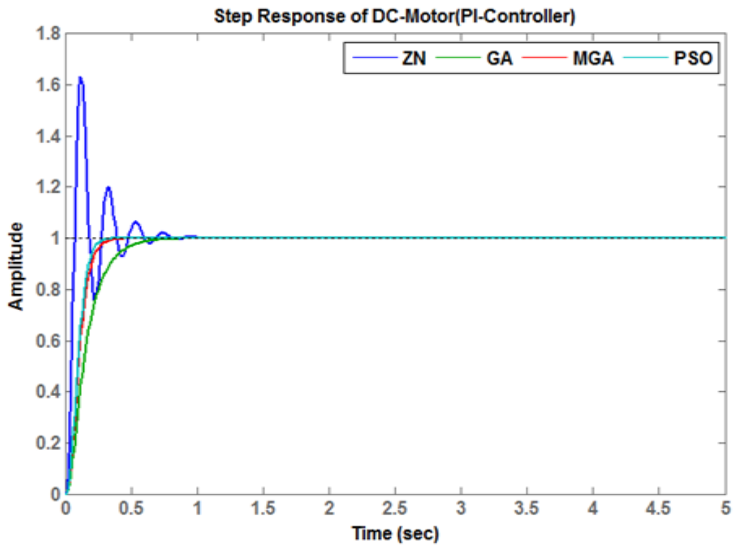


**Figure 5:** Step Response of DC-Motor with PI-Controller using ZN, GA, MGA and PSO.

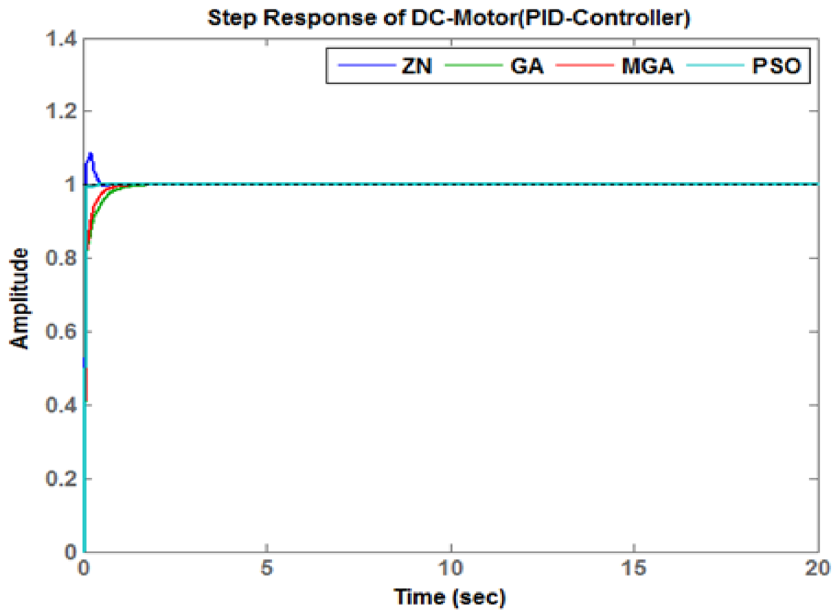


**Figure 6:** Step Response of DC-Motor with PID-Controller using ZN, GA, MGA and PSO.

### 5.2 AUTOMATIC *VOLTAGE REGULATOR (AVR)*

The role of an AVR is to keep the terminal voltage magnitude of a synchronous generator at a specified level. A simple AVR system comprises of four main components, namely amplifier, exciter, generator, and sensor [20, 21]. The transfer function of these components may be represented by equations (10), (11), (12) and (13) respectively [22]. The block diagram of the AVR system with a PID-controller [21, 23] is shown in figure (7).

$$\frac{V_R(s)}{V_e(s)} = \frac{K_A}{1+\tau_A s} \quad , K_A = 10, \tau_A = 0.1s \tag{10}$$

$$\frac{V_F(s)}{V_R(s)} = \frac{K_E}{1+\tau_E s} \quad ,K_E = 1, \tau_E = 0.4s \tag{11}$$

$$\frac{V_t(s)}{V_F(s)} = \frac{K_G}{1+\tau_G s} \quad , K_G = 0.7, \tau_G = 1s \tag{12}$$

$$\frac{V_S(s)}{V_t(s)} = \frac{K_R}{1+\tau_R s} \quad , K_R = 1, \tau_R = 0.01s \tag{13}$$

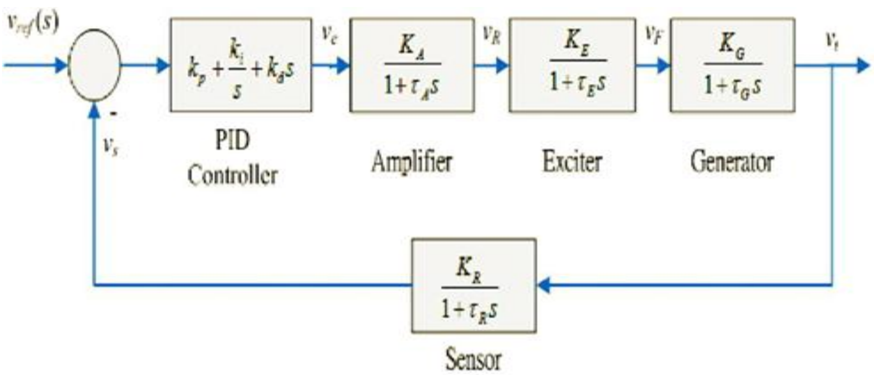


**Figure 7:** Block diagram of the AVR System with PID controller

Figures (8) and (9) show the step responses of the AVR system when controlled by PI-Controller and PID-Controller for the tuning methods investigated.

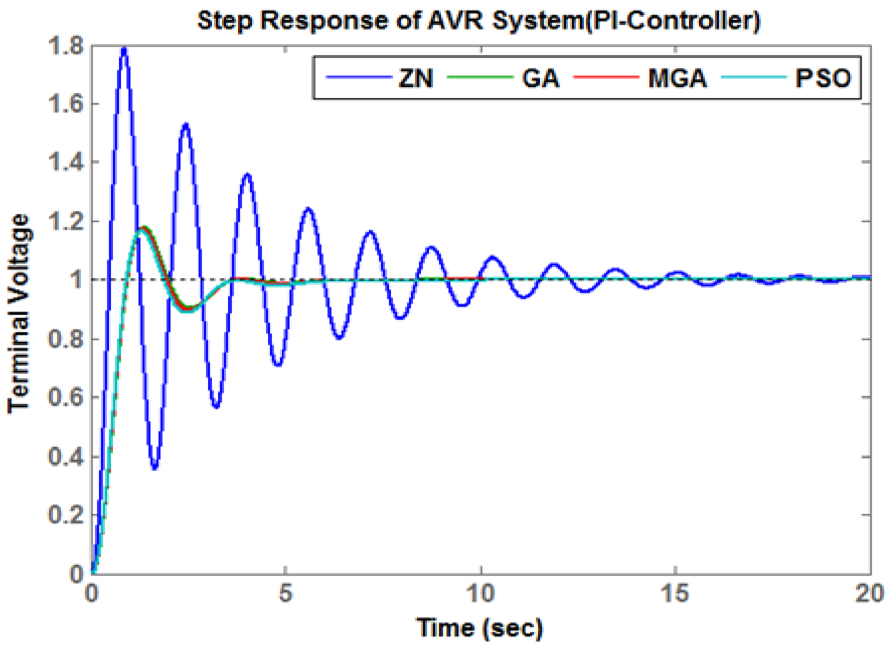


**Figure 8:** Step Response of The AVR System with PI-Controller using ZN, GA, MGA and PSO.

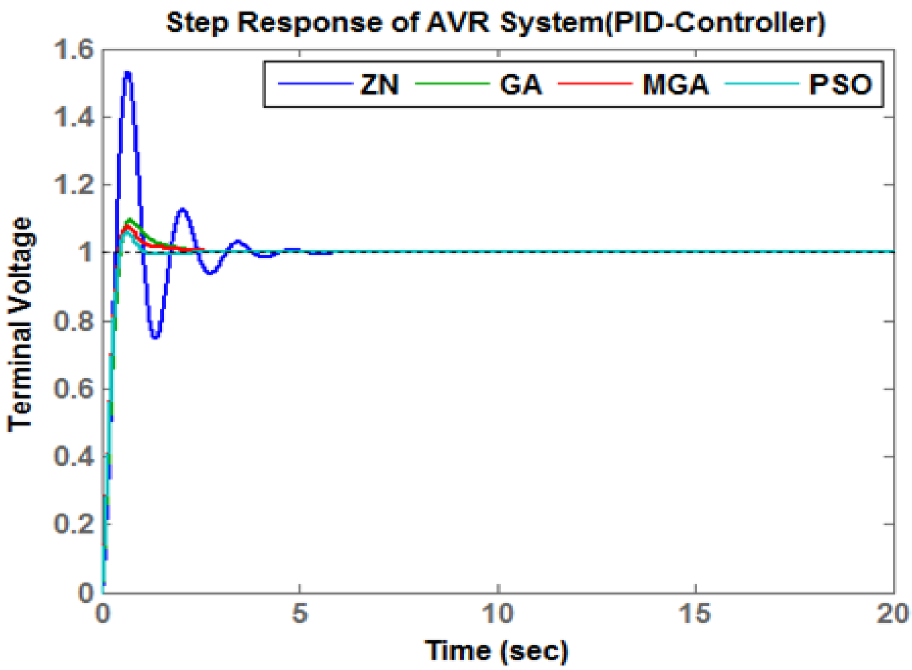


**Figure 9:** Step Response of The AVR System with PID-Controller using ZN, GA, MGA and PSO.

Tables (3) and (4) present the controller parameters, maximum overshoot, rise time and settling time for the tuning techniques investigated in this work for the PI controller and PID controller cases respectively.

**TABLE 3:** AVR in Closed-Loop Controlled Using a PI-Controller

| Tuning Method | ZN | GA | MGA | PSO |
|---|---|---|---|---|
| $K_p$ | 1.093 | 0.413 | 0.424 | 0.4379 |
| $K_i$ | 1.203 | 0.251 | 0.243 | 0.2329 |
| $Max.OS$ | 78.7 | 18 | 17.5 | 16.9 |
| $t_r$(sec) | 0.289 | 0.604 | 0.598 | 0.59 |
| $t_s$(sec) | 15.2 | 3.38 | 3.36 | 3.35 |

The results presented in table (3) show that the maximum overshoot using proposed Particle Swarm Optimization (PSO) is reduced by 24% from the nearest best result in Modified Genetic Algorithm (MGA), rise time reduced by 2% and settling time by 0.3% that indicates the performance of AVR system is improved. The results obtained for the PID controller in table (4) show that the performance of AVR system is improved using the proposed Particle Swarm Optimization (PSO).

**TABLE 4:** AVR in Closed-Loop Controlled Using a PID-Controller

| Tuning Method | ZN | GA | MGA | PSO |
|---|---|---|---|---|
| $K_p$ | 1.457 | 0.893 | 0.96 | 0.9564 |
| $K_i$ | 2.006 | 0.848 | 0.826 | 0.6725 |
| $K_d$ | 0.149 | 0.236 | 0.265 | 0.2613 |
| $Max.OS$ | 53.4 | 9.37 | 7.82 | 5.95 |
| $t_r$(sec) | 0.249 | 0.326 | 0.301 | 0.301 |
| $t_s$(sec) | 3.63 | 1.64 | 1.22 | 0.926 |

## 6. CONCLUSION

The PID controllers tuned using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) exhibited better steady-state response and performance indices than conventional tuning methods. The computation time for solving the optimization is a fraction of a second. The controller tuned using Particle Swarm Optimization (PSO) algorithms resulted in the most satisfactory performance (no overshoot, minimal rise time, steady state error is equal zero). The simulations for PSO tuned PID controllers show that it results in higher quality solution with better computational efficiency. The proposed PSO method is robustly stable and is more efficient than the GA method in solving the tuning problem of PID controllers.

## 7. REFERENCES

[1] K.J. Astrom, T. Hagglund, “PID Controllers: Theory, Design and Tuning”, International Society for Measurement and Con., 2nd Ed., pp.200-210, 1995.
[2] A. Jalilvand, A. Kimiyaghalam, A. Ashouri, H. Kord, “Optimal Tuning of PID Controller Parameters on a DC Motor Based on Advanced Particle Swarm Optimization Algorithm”, ISSN 2077-3528,IJTPE Journal, 2011.
[3] Guillermo J. Costa, www.powertransmission.com, “Tuning a PID Controller”, April 2011.
[4] Ali Marzoughi, Hazlina Selamat, Mohd Fua’ad Rahmat and Herlina Abdul Rahim, “Optimized proportional integral derivative (PID) controller for the exhaust temperature control of a gas turbine system using particle swarm optimization”, International Journal of the Physical Sciences Vol. 7(5), pp. 720 - 729, 30 January, 2012.
[5] Katsuhiko Ogata, University of Minnesota, “Modern Control Engineering, Fourth Edition”, Publisher: Aeeizb, 2002.
[6] A. Zilouchian, M. Jamshidi, “Intelligent Control Systems Using Soft Computing Methodologies”, by CRC Press LLC, 2001.
[7] Mohammed Obaid Ali, S. P. Koh, 1K. H. Chong, S.K.Tiong and Zeyad Assi Obaid, “Genetic Algorithm Tuning Based PID Controller for Liquid-Level Tank System”, Proceedings of the International Conference on Man-Machine Systems (ICoMMS)11 – 13 October 2009, Batu Ferringhi, Penang, MALAYSIA, 2009.
[8] N. Pillay, “A Particle Swarm Optimization Approach for Tuning of SISO PID Control Loops”, Durban University of Technology, 2008.
[9] A.A. Javadi, R. Farmani, T.P. Tan, “A hybrid intelligent genetic algorithm”, www.elsevier.com/locate/aei, Advanced Engineering Informatics 19 (2005) 255–262, 2005.
[10] Andries P. Engelbrecht, “Computational intelligence: an introduction-2nd Ed.”, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2007.
[11] K.E. Parsopoulos and M.N. Vrahatis, “Recent approaches to global optimization problems through Particle Swarm Optimization”, Academic Publishers. Natural Computing 1: 235–306, 2002.
[12] J´anos M´adar, J´anos Abonyi and Ferenc Szeifert, “Interactive Particle Swarm Optimization”, Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA’05), 2005.
[13] Dian Palupi Rini, Siti Mariyam Shamsuddin and Siti Sophiyati Yuhaniz, “Particle Swarm Optimization: Technique, System and Challenges”, International Journal of Computer Applications (0975 – 8887) Volume 14– No.1, January 2011.
[14] A. Marzoughi, H. Selamat, M. F. Rahmat, H. Abdul Rahim, “ Optimized Proportional Integral Derivative (PID) Controller for the Exhaust Temperature Control of a Gas Turbine System Using Particle Swarm Optimization”, International Journal of the Physical Sciences Vol. 7(5), pp. 720-729, 2012.
[15] M. A. Tammam, Faculty of Engineering, Cairo University, Giza, Egypt, “Multi Objective Genetic Algorithm Controller’s Tuning for Load Frequency Control in Power Systems”, 2010.

[16] Anant Oonsivilai and Padej Pao-La-Or, "Optimum PID Controller tuning for AVR System using Adaptive Tabu Search", 12th WSEAS International Conference on COMPUTERS, Heraklion, Greece, July 23-25, 2008.
[17] A. Bagis, "Determination of the PID Controller Parameters by Modified Genetic Algorithm for Improved Performance", Journal of Information Science and Engineering 23, 1469-1480 (2007).
[18] N. Thomas, Dr. P. Poongodi, "Position Control of DC Motor Using Genetic Algorithm Based PID Controller", Proceedings of the World Congress on Engineering 2009 Vol II, London, U.K., 2009.
[19] A. Jalilvand, A. Kimiyaghalam, A. Ashouri and H. Kord, "OPTIMAL TUNING OF PID CONTROLLER PARAMETERS ON A DC MOTOR BASED ON ADVANCED PARTICLE SWARM OPTIMIZATION ALGORITHM", International Journal on "Technical and Physical Problems of Engineering" (IJTPE) Published by International Organization on TPE (IOTPE). December 2011.
[20] Zwe-Lee Gaing, "A Particle Swarm Optimization Approach for Optimum Design of PID Controller in AVR System", IEEE Transactions on Energy Conversion, Vol. 19, NO. 2, JUNE 2004.
[21] Ching-Chang Wong, Shih-An Li and Hou-Yi Wang, "Optimal PID Controller Design for AVR System", Tamkang Journal of Science and Engineering, Vol. 12, No. 3, pp. 259_270 (2009).
[22] S. F. Kheder, "Genetic Algorithm Based Controller's Tuning for Linear and Nonlinear Automatic Voltage Regulator in Electrical Power System", M.Sc. thesis, Faculty of Engineering, Cairo University 2013.
[23] S. M. A. Mohammadi, A.A. Gharaveisi, M. Mashinchi, and S.M.R. Rafiei, Senior Member, IEEE, "New Evolutionary Methods for Optimal Design of PID Controllers for AVR System", Paper accepted for presentation at IEEE Bucharest Power Tech Conference, June 28th - July 2nd, Bucharest, Romania, 2009.