Military Technical College
Kobry El-Kobbah,
Cairo, Egypt

9th International Conference
on Electrical Engineering
ICEENG 2014

Tuning PID Controllers Using Artificial Intelligence Techniques

By

A. A. Salem* Prof. Dr. M. Mustafa** Dr. M. E. Ammar**

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).

Keywords:

PID-controller, DC-Motor, AVR system, Genetic Algorithm, Modified Genetic Algorithm, Particle Swarm Optimization.

1. Introduction:

The Proportional-Integral-Derivative (PID) controller is one of the most successful industrial controllers due to ease of implementation, simplicity and robust performance. A PID control is a linear control methodology with a very simple control structure. 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 methods. Conventional methods such as Ziegler-Nichols method do not provide optimal PID tuning parameters and usually produce oscillations and overshoots [2]. A technique for multivariable PID design using bilinear matrix inequalities is presented in [3]. A PID tuning method (PID) for singe loop control of turbine speed control system by approximating the feedback form of an IMC controller with elimination of higher order terms in the controller form to classical PID form is proposed in [4]. Tuning PID controllers for robust performance based on internal model control techniques is proposed in [5]. Artificial intelligence approaches have been applied successfully to solve the optimization problem of tuning PID

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

controller parameters for performance. In [6], a comparison between different methods, based on fuzzy logic, for the tuning of PID controllers is presented. The performance of genetic algorithm-tuned fuzzy logic like PID, genetic algorithm-tuned ANN like PID and genetic algorithm-tuned fuzzy like PID using adaptive neuron fuzzy inference system (ANFIS) were investigated in [7]. Adaptive genetic algorithms (AGA) are proposed in [8] as a method for PID optimization.

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, 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 be adjusted to result in a satisfactory closed-loop performance. The transfer function of PID controller is defined as follows :

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \quad (1)$$

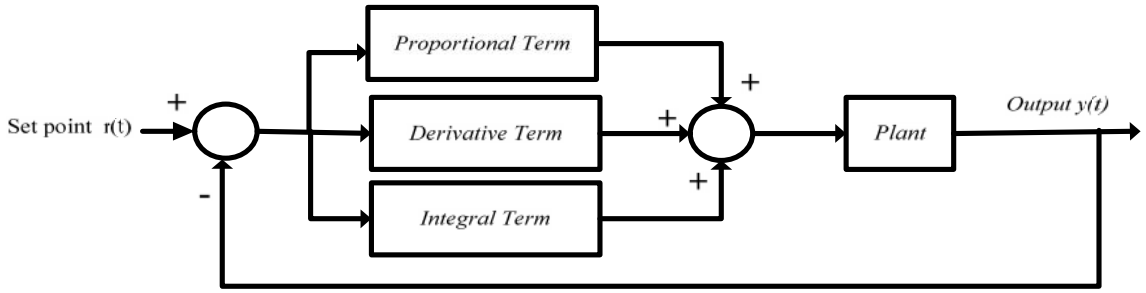

Figure (1): *Closed loop PID controlled system*

2. Artificial Intelligence Optimization Techniques

2.1 Genetic Algorithms:

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

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

techniques or methods that rely on derivative information [9]. Figure (2) shows the steps of creating and implementing the genetic algorithm.

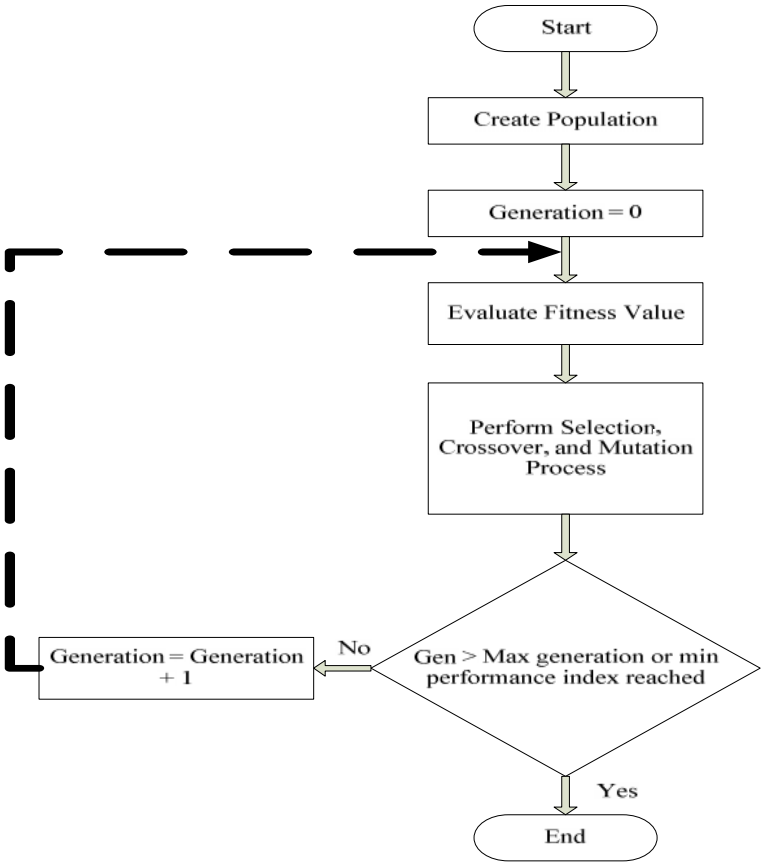

Figure (2): *Genetic Algorithm Flowchart*

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.
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 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.

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

- Crossover (or Recombination) is synonymous to mating. During crossover two individual agents are combined to produce an offspring. The main objective of crossover is to explore new areas within the search space.

Step 5: The aforementioned steps are repeated until the swarm converges to an optimal or sub-optimal solution [10].

The genetic Algorithm advantages are:

- 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.
- 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.

2.2 Modified Genetic Algorithm:

Genetic algorithms (GA) depend on the initial population chosen. There might 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. 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. For the new solutions, an optimization algorithm based on the integration of classical genetic algorithm structure and a systematic neighborhood structure is employed to achieve more effective search. The neighborhood unit performs two main tasks [11]:

(i) Obtain the neighbor solutions for the best solution found.
(ii) Find a new best solution with higher quality if possible.

Since this strategy suggests the solution diversity, all of the probable solutions with higher quality can be searched.

2.3 Particle Swarm Optimization:

Particle swarm optimization (PSO) was inspired by the social behavior of bird flocking where computer algorithms simulate the complicated flocking behavior of birds [10]. 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 from its own experience and the position of the best particle in its entire population. The best position

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

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 to equations (2) and (3) respectively:

$$\mathbf{v_i^{k+1} = v_i^k + c_1 rand_1 \times (pbest_i - x_i^k) + c_2 rand_2 \times (gbest - x_i^k)} \quad (2)$$

$$\mathbf{x_i^{k+1} = x_i^k + v_i^{k+1}} \quad (3)$$

Where:

$\mathbf{v_i^k}$: current velocity of particle i at iteration k

$\mathbf{v_i^{k+1}}$: new velocity of particle i next at iteration k +1
$\mathbf{c_1}$: cognitive acceleration constant (self-confidence)
$\mathbf{c_2}$: social acceleration constant (swarm confidence)
$\mathbf{x_i^k}$: current position of particle i at iteration k
$\mathbf{x_i^{k+1}}$: new position of particle i at next iteration k+1
$\mathbf{pbest_i}$: personal best of particle i
$\mathbf{gbest}$: Global best of the population

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

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.

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

The particle swarm optimization (PSO) differs from genetic algorithms (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 [10].

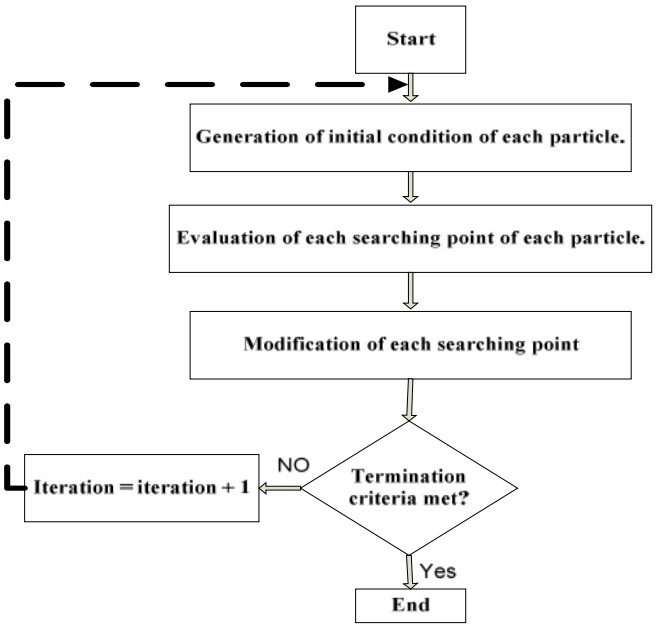

Figure (3): *The Flowchart of Particle Swarm Optimization.*

3. Applications:

This paper applies the artificial intelligence tuning methods to control a DC motor and an automatic voltage regulator (AVR) system. The performance of the closed-loops controlled by a PID tuned using the proposed techniques are compared to loops tuned through the Ziegler-Nichols method. The closed-loop simulations were run on MATLAB 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 tuning of PID controllers for the two simulations.

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

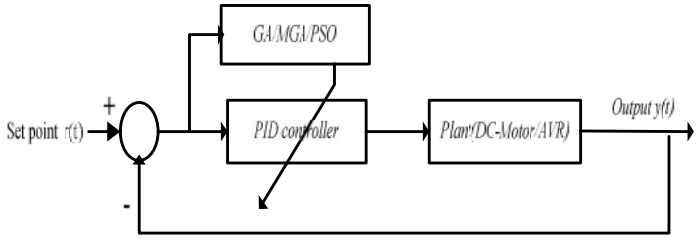

Figure (4): *Tuning PID controller based on GA, MGA and PSO.*

3.1 Position Control Of Dc-Motor:

A simulation is run with a DC motor that has the following specifications:
2hp, 230 v, 8.5 amperes, 1500 rpm

$\mathbf{R_a(Armature\ Resistance) = 2.45\ \Omega}$,

$\mathbf{L_a(Armature\ Inductance) = 0.035\ H}$,

$\mathbf{K_b(Back\ EMF) = 1.2\frac{Vs}{rad}}$,

$\mathbf{J_m(Moment\ of\ Inertia) = 0.022\ kgm^2}$,

$\mathbf{B_m(Frictional\ Constant) = 0.5 \times 10^{-3}}$ *(*$\mathbf{N_m}$*S/rad).*

The transfer function of DC motor is [12]:

$$\frac{\theta(s)}{V_a(s)} = 1.\frac{2}{0.00077s^3 + 0.0539s^2 + 1.441s} \quad (8)$$

The step responses of a DC motor controlled by a PI-Controller and PID-Controller tuned through the proposed artificial intelligence techniques 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 with 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

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

***TABLE (2):** DC motor in closed-loop with 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

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

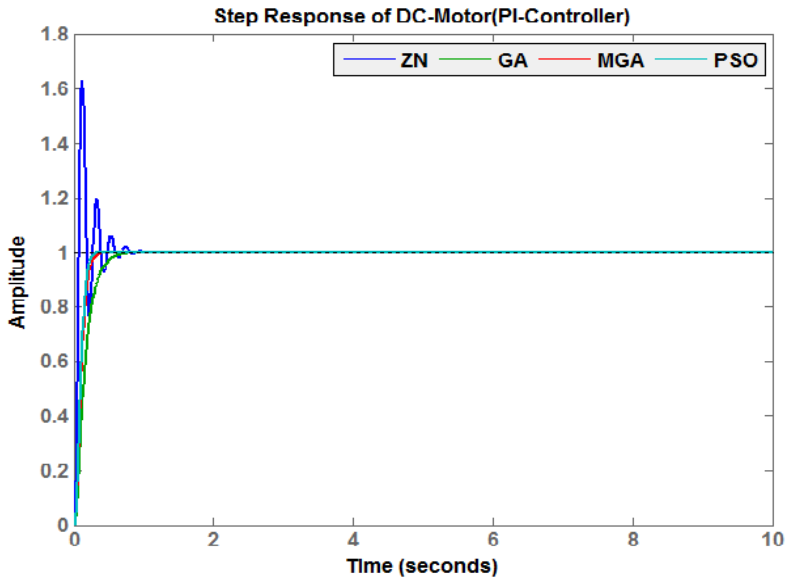

Figure (5): *Step response of the DC motor with PI-Controller using ZN, GA, MGA and PSO.*

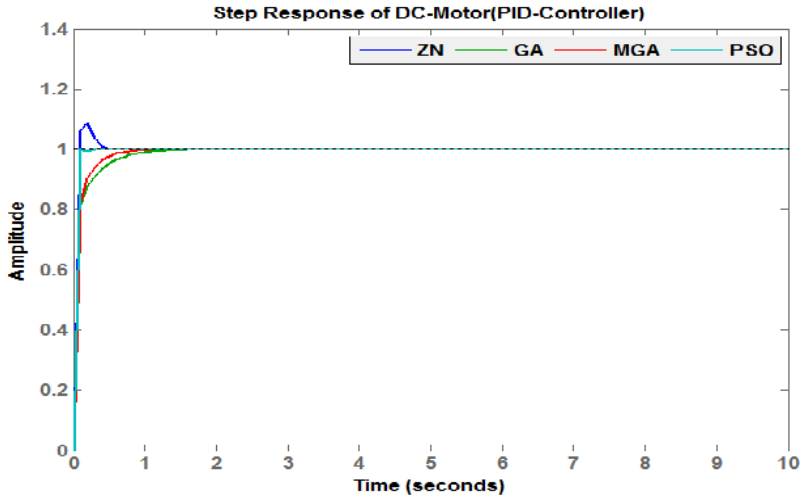

Figure (6): *Step response of the DC-Motor with PID-Controller using ZN, GA, MGA and PSO.*

3.2 Automatic Voltage Regulator:

The role of an AVR is to keep the terminal voltage magnitude of a synchronous generator at a specified level. A simple AVR system is comprised of four main components, namely amplifier, exciter, generator, and sensor [13] and [14]. The transfer function of these components may be represented, respectively, as shown in equations (9), (10), (11) and (12). The block diagram of the AVR system with a PID-controller [15] is shown in figure (7).

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

$$\frac{V_R(s)}{V_e(s)} = \frac{K_A}{1+\tau_A s} \quad , K_A = 10, \tau_A = 0.1s \tag{9}$$

$$\frac{V_F(s)}{V_R(s)} = \frac{K_E}{1+\tau_E s} \quad , K_E = 1, \tau_E = 0.4s \tag{10}$$

$$\frac{V_t(s)}{V_F(s)} = \frac{K_G}{1+\tau_G s} \quad , K_G = 0.7, \tau_G = 1s \tag{11}$$

$$\frac{V_s(s)}{V_t(s)} = \frac{K_R}{1+\tau_R s} \quad , K_R = 1, \tau_R = 0.01s \tag{12}$$

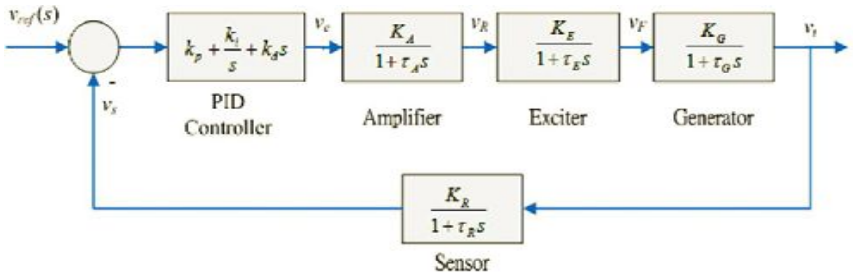

Figure (7): *Block diagram of the AVR System with PID controller*

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.

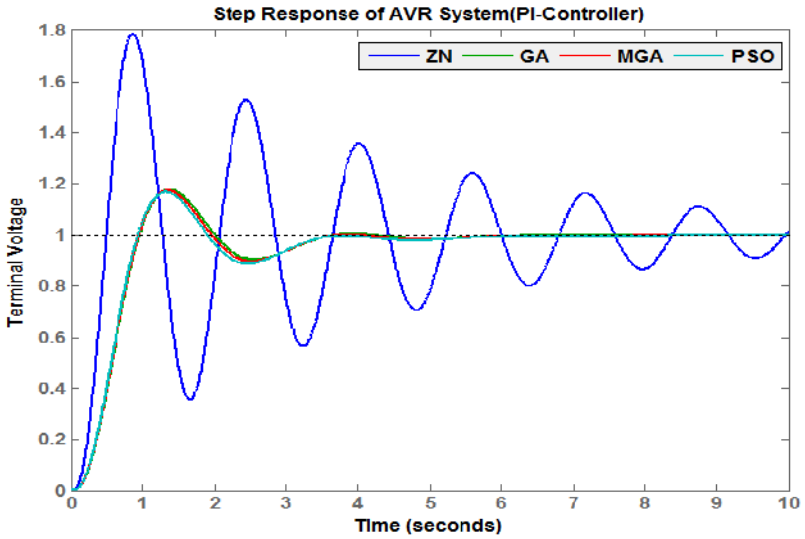

Figure (8): *Step response of the AVR System with PI-Controller using ZN, GA, MGA and PSO.*

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

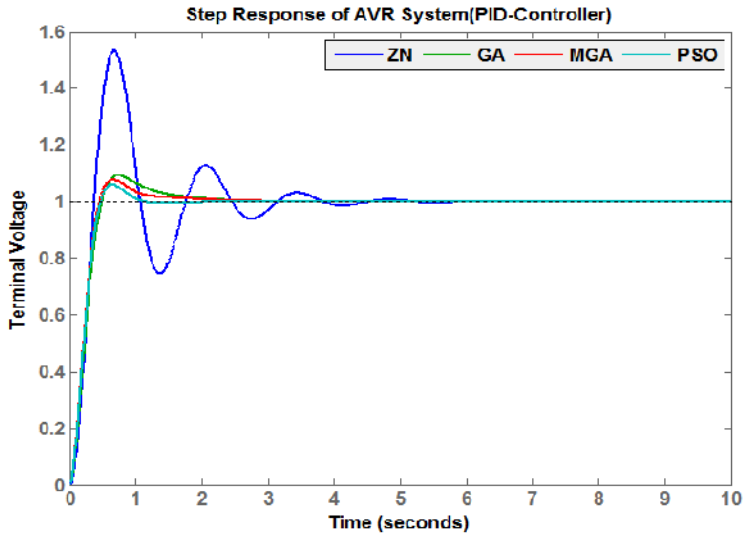

Figure (9): *Step response of the AVR System with PID controller using ZN, GA, MGA and PSO.*

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

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

4. Conclusion:

The PID controllers tuned using Genetic Algorithm (GA) and Particle Swarm

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

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.

5. 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, issue 9, vol. 3, no. 4, pp. 10-17, 2011.

[3] F.D.Bianchi, R.J. Mantz, C.F. Christiansen, "Multivariable PID Control with Set-Point Weighting via BMI Optimization", Automatica, Vol. 44, pp. 472-478, 2008.

[4] M.V. Subramanyam, K. Satya Prasad, and P.V. Gopi Krishna Rao"Robust Control of Steam Turbine System Speed Using Improved IMC Tuned PID Controller". International Conf. on Modelling, Optimization and Computing, Procedia Engineering, vol. 38, pp. 1450–1456, 2012.

[5] P. V. Gopi Krishna Rao, M. V. Subramanyam, and K. Satyaprasad,Model based Tuning of PID Controller", Journal of Control & Instrumentation, Vol. 4, Issue 1.

[6] A. Visioli,"Tuning of PID controllers with fuzzy logic", IEE Proceedings on Control Theory and Applications, vol. 148, issue 1, pp. 1-8, 2001.

[7] Khedr, S.F.M.; Ammar, M.E. and Hassan, M.A.M. "Multi objective genetic algorithm controller's Tuning for non-linear automatic voltage regulator" Proceedings of the 2013 International Conference on Control, Decision and Information Technologies (CoDIT), pp. 857 – 863, May 2013.

[8] Guohan Lin and Guofan Liu, "Tuning PID controller using adaptive genetic algorithms", Proceedings of the 5th International Conference on Computer Science and Education (ICCSE), pp. 519-523 Aug. 2010.

[9] A. Zilouchian, M. Jamshidi, "Intelligent Control Systems Using Soft Computing Methodologies", by CRC Press LLC, 2001.

[10] 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",

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University

International Journal of the Physical Sciences Vol. 7(5), pp. 720-729, 2012.

[11] 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).

[12] 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.

[13] 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.

[14] 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).

[15] 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.

Nomenclatures:

PID	Proportional, Differential and Integral Controller.
K_p	Proportional Gain.
K_i	Integral Gain.
K_d	Differential Gain.
t_r	Rise Time.
OS	Maximum (Percent) Overshoot, Overshoot.
$rand_{1,s}$	Random number between 0 and 1.
K_A	Amplifier Model Gain.
τ_A	Amplifier Model Time Constant.
V_R	Amplifier Model Output.
K_E	Exciter Model Gain.
τ_E	Exciter Model Time Constant.
V_F	Exciter Model Output.
K_G	Generator Model Gain.
τ_G	Generator Model Time Constant.
V_t	Generator Model Output (Terminal Voltage).
K_R	Sensor Model Gain.
τ_R	Sensor Model Time Constant.
V_s	Sensor Model Output.

* Shams Industry Company (*HISENSE EGYPT*) ** Dept. of Electric Power and machines, Cairo University