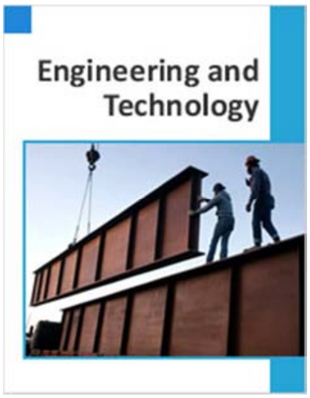
Engineering and Technology
2017; 4(3): 31-42
http://www.aascit.org/journal/et
ISSN: 2381-1072 (Print); ISSN: 2381-1080 (Online)

American Association for Science and Technology



An imperialist Competitive Algorithm (ICA) for Optimal Design of PID Controller in AVR System

Mojtaba Ghasemi

Department of Electronics and Electrical Engineering, Shiraz University of Technology, Shiraz, Iran

Email address
mojtaba.ghasemi1365@yahoo.com

Citation
Mojtaba Ghasemi. An imperialist Competitive Algorithm (ICA) for Optimal Design of PID Controller in AVR System. *Engineering and Technology*. Vol. 4, No. 3, 2017, pp. 34-42.

Keywords
Imperialist Competitive Algorithm,
Automatic Voltage Regulator,
Optimal Controller Design,
PID Controller

Received: April 11, 2017
Accepted: May 18, 2017
Published: August 21, 2017

Abstract
In this paper, the imperialist competitive algorithm (ICA) is used for determining optimal proportional-integral-derivative (PID) controller parameters of an automatic voltage regulator (AVR) system. In this study, four performance criteria that mostly used in evaluating performance of the PID controller are employed to the proposed AVR system. The optimal PID controller is designed in such a way that it minimizes the set of settling time, rise time, maximum overshoot and integral absolute error of the AVR step response. The used approach had superior features, stable convergence characteristic and including easy implementation respect to classic and other evolution algorithm. Simulation results demonstrate that the designed controller adapt themselves to varying loads and hence provide better performance when compared with based controllers such as particle swarm optimization (PSO) algorithm, genetic algorithm (GA) and the shuffled frog leaping algorithm (SFLA).

1. Introduction

An automatic voltage regulator (AVR) is used to hold terminal voltage magnitude of a synchronous generator at a specified level. The AVR must track the generator terminal voltage at all times and under any load conditions, to keep the voltage magnitude within pre-established limits [1].

In recent years large advances have been made in process control techniques. For linear systems, proportional-integral-derivative (PID) controller has been widely used in industrial control processes because of its simple structure, robust performance and being easily tunable in wide range of operating conditions. Despite the fact that control theory has been significantly developed, the PID controllers are still used in many industrial applications such as process controls, motor drivers, flight control, instrumentation, etc [2].

Many approaches have been documented in literatures for determining the PID parameters which was first found by Ziegler Nichols tuning. Genetic algorithm, neural network, fuzzy based approach, particle swarm optimization and shuffled frog leaping algorithm (SFLA) techniques are just a few examples of these numerous works [3, 4].

Genetic algorithms were formally introduced in the United States in the 1970s by John Holland at University of Michigan. The continuing price/performance improvements of computational systems made them attractive for some types of optimization. In particular, genetic algorithms work very well on mixed (continuous and discrete) combinatorial problems. They are less susceptible to getting 'stuck' at local optima than gradient search methods. But they tend to be computationally expensive. Although the GA approach has many advantages but its defects also have been found recently. This degradation in efficiency is more apparent in applications with the highly correlated parameters which are to be optimized [5].

Particle swarm optimization (PSO) is a population-based stochastic optimization technique, and was developed by Kennedy and Eberhart in 1995. PSO simulates the social behavior of organisms, such as bird flocking and fish schooling, to describe an automatically evolving system. In PSO, each single candidate solution is "an individual bird of the flock", that is, a particle in the search space. Each particle makes use of its individual memory and knowledge gained by the swarm as a whole to find the best solution. All particles have fitness values, which are evaluated by fitness function in order to be optimized, and velocities which direct the movement of particles. During movement, each particle adjusts its position according to its own experience, as well as according to the experience of a neighboring particle, and makes use of the best position encountered by itself and its neighbor. The particles move through the problem space by following a current of optimum particles [6].

Particle swarm optimization approach is a metaheuristic as it makes small or no assumptions about the problem being optimized and can search wide space of candidate solutions. It must be noted that metaheuristic methods such as particle swarm optimization do not assure an optimal solution. On the other hand, particle swarm optimization does not use the gradient of the problem being optimized, which means PSO does not need that the optimization problem to be differentiable as it is required by classic optimization methods such as gradient descent and quasi-newton methods. PSO can therefore be used on optimization problems that are partially irregular, noisy or change over time too. PSO can converge quickly in early stage, but easily plunges into local minimum in the late optimization.

SFLA method is a heuristic search algorithm presented for the first time by Eusuff and Lansey in 2003. The major purpose of this algorithm was achieving a method to solve complicated optimization problems without any use of traditional mathematical optimization tools. In fact, the SFLA algorithm is combination of "meme-based genetic algorithm or memetic algorithm" and "particle swarm optimization (PSO)". This algorithm has been inspired from memetic evolution of a group of frogs when seeking for food [7].

Recently, a new algorithm has been proposed by Atashpaz-Gargari and lucas [8] in 2007. Inspired from a socio-human phenomenon, this evolutionary optimization algorithm has been successfully utilized in many engineering applications such as for the non-smooth optimal power flow (OPF) problems [9-12], and for the optimal reactive power dispatch (ORPD) problems [13-14] in recent years and has shown great performance in both convergence rate and achieving global optimal.

ICA is an evolutionary algorithm for global optimization, whose motivation is to simulate the process of dominating the colonies by powerful imperialistic countries through competition during the colonialism. Therefore, the ICA algorithm is also called colonial competitive algorithm (CCA). Although this algorithm occurred lately, it has been quite successful in a wide range of applications such as adaptive antenna arrays, intelligent recommender systems, optimal controller for industrial and chemical possesses [14].

In this study, besides demonstrating how to employ the ICA algorithm to obtain the optimal PID controller parameters of an AVR system, many performance estimation schemes are performed to examine whether the proposed method has better performance than the real-value PSO algorithm in solving the optimal PID controller parameters.

2. Optimization Problem

2.1. Problem Formulation

Imperialist competitive algorithm is a novel global search heuristic that is inspired by the socio-political competition. Figure 1 shows the pseudo code for this algorithm. Like other evolutionary ones, this algorithm starts with an initial population. In this algorithm any individual of the population is called a country. Some of the best countries in the population are selected to be the imperialist states and all the other countries form the colonies of these imperialists. All the colonies of initial population are divided among the mentioned imperialists based on their power which are inversely proportional to their cost.

```
1) Select some random points on the function
and initialize the empires.

2) Move the colonies toward their relevant
imperialist (Assimilating).

3) Randomly change the position of some
colonies (Revolution).

4) If there is a colony in an empire which
has lower cost than that of imperialist,
exchange the positions of that colony and
the imperialist.

5) Compute the total cost of all empires
(Related to the power of both imperialist
and its colonies).

6) Pick the weakest colony (Colonies) from
the weakest empire and give it (Them) to the
empire that has the most likelihood to
possess it (Imperialistic competition).

7) Eliminate the powerless empires.

8) If stop conditions satisfied, stop, if
not go to 2.
```

Figure 1. *Pseudo code for the ICA algorithm.*

After dividing all colonies among imperialists and creating the initial empires, these colonies start moving toward their relevant imperialist country. This movement is a simple model of assimilation policy that was perused by some imperialist states. Figure 2 shows the movement of a colony towards the imperialist. In this movement, θ and x are random numbers with uniform distribution and d is the distance between colony and the imperialist [15].

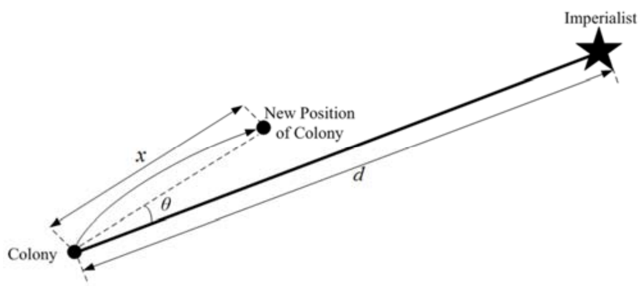


Figure 2. *Motion of colonies toward their relevant imperialist.*

$$X \sim U(0, \beta * d)$$
$$\theta \sim U(-\gamma, \gamma) \quad (1)$$

β and γ are arbitrary numbers that modify the area colonies randomly search around the imperialist. In this paper implementation β and γ are 2 and 0.5 (rad) respectively.

In each generation certain numbers of countries go through a sudden change which is called “revolution”. This process is similar to mutation process in GA which helps the optimization process in escaping local optimal traps [11]. Figure 3 shows the revolution in culture-language axis.

The total power of an empire depends on both the power of the imperialist country and the power of its colonies. In this algorithm, this fact is modeled by defining the total power of an empire as the power of imperialist state plus a percentage of the mean power of its colonies. Any empire that is not able to succeed in imperialist competition and cannot increase its power (or at least prevent decreasing its power) will be eliminated. The imperialistic competition will gradually results in an increase in the power of great empires and a decrease in the power of weaker ones. Weak empires will lose their power gradually and ultimately they will collapse. Basically the competition can be continued until there would be only one imperialist in the search space, However, different conditions may be selected as termination criteria including reaching a maximum number of iterations or having negligible improvement in objective function. As long as the convergence criterion is not satisfied, the algorithm continues (Figure 1) [11, 15].

2.2. AVR System

One of the most important solutions for improvement of power system stability and keeping the quality of generated electrical power at a desired level is Synchronous generator excitation control. The goal of an AVR system is to maintain the system voltage between defined limits by adjusting the excitation of the synchronous generator. An AVR system operates by measuring the difference between a rectified voltage derived from the stator voltage and a reference voltage. This error signal is then amplified and fed to the excitation circuit. The change of excitation maintains the volt amp reactive (VAR) balance in the network. This method is also referred to as megawatt volt amp reactive (MVAR) control or reactive-voltage (QV) control [16].

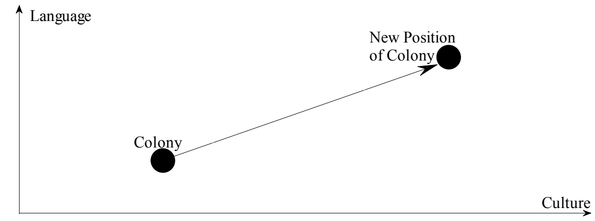


Figure 3. *Revolution; a sudden change in socio-political characteristics of a country.*

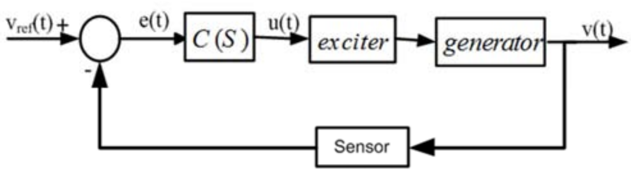


Figure 4. *Block diagram of voltage control in a generator.*

2.2.1. Proportional-Integral-Derivative (PID) Controller

Block diagram of the AVR control system is shown in Figure 4. In this figure, *C(s)* is the controller. The goal is to design a controller so that the output of the AVR system has some defined characteristics.

The design a PID controller to control the AVR system. For a PID controller, *C(s)* is defined as:

$$C(s) = K_P + \frac{K_i}{S} + K_d S \quad (2)$$

where K_p, K_i, and K_d are the proportional, integral and derivative gains respectively. The output of the AVR system will be:

$$u(t) = K_p e(t) + K_i \int_0^\infty e(t)dt + K_d \dot{e}(t) \quad (3)$$

For a given AVR system, the problem of designing a PID controller is to adjust the parameters K_p, K_i, and K_d for getting a desired performance of the considered AVR system.

The output of the AVR system has some important characteristics. We use four most important ones to design our controller. These characteristics are:

2.2.2. Rise Time

The rise time is defined as the time required for the step response to rise from 10 to 90 percent of its final value.

2.2.3. Settling Time

The settling time, is defined as the time required for the step response to stay within δ percentage of its final value. Where the value of δ is in the range 2 to 5. In our simulation, δ was set to 5.

2.2.4. Max Overshoot

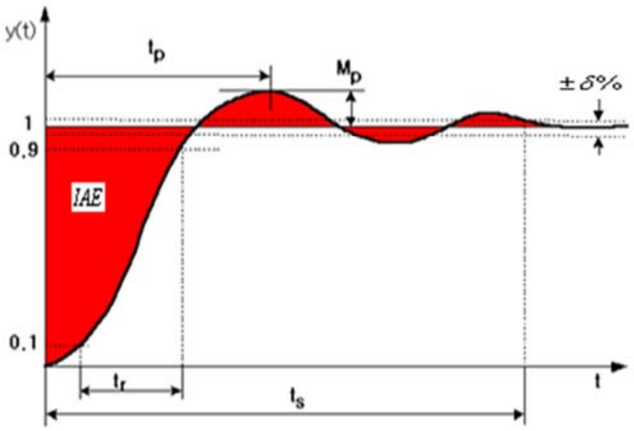


Figure 5. *Rise time, Settling time, Max over shoot and Integral absolute error.*

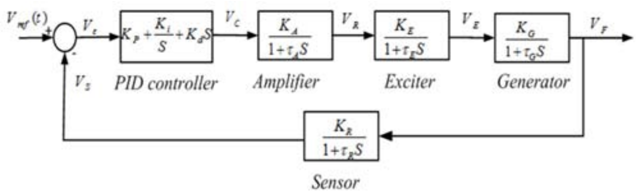


Figure 6. *Block diagram of an AVR system with a PID controller.*

If max y denotes the maximum value of y and y_{ss}, represent the steady-state value of it, the maximum overshoot will be defined as:

$$MaxOvershoot = y_{\max} - y_{ss} \tag{4}$$

2.2.5. Integral Absolute Error (IAE)

The integral of the absolute error (IAE) is defined as:

$$IAE = Ki\int_0^{\infty} e(t)\,dt \tag{5}$$

In our implementation we calculate the integral up to (4* settling time) which is an acceptable approximation of the real value of IAE.

All the above characteristics are shown in Figure 5.

2.3. Linearized Model of an AVR System

The basic objective of the AVR system is to control the terminal voltage by adjusting the generator exciter voltage. A simple AVR system comprises four main components, namely amplifier, exciter, generator, and sensor. For mathematical modeling and determining transfer functions of the four components, these components must be linearized, which takes into account the major time constant and ignores the saturation or other nonlinearities. The approximate transfer functions of these components are represented respectively, as follows [17].

2.3.1. Amplifier Model

The amplifier model is represented by a gain K_A and a time constant τ_A. The transfer function is as follows:

$$\frac{V_R(s)}{V_C(s)} = \frac{K_A}{1+\tau_A S} \tag{6}$$

where the value of K_A is in the range of 10 to 400 and the value of the amplifier time constant τ_A is in the range of 0.02s to 0.1s. In our simulation, K_A was set to 10 and τ_A was set to 0.1s.

2.3.2. Exciter Model

After linearization, the transfer function of a modern exciter can be represented by a gain K_E and a single time constant τ_E.

$$\frac{V_E(s)}{V_R(s)} = \frac{K_E}{1+\tau_E S} \tag{7}$$

where the value of K_E ranges from 1 to 200 and the value of the amplifier time constant τ_E is in the range of 0.5s to 1s. In this simulation, K_E was set to 1 and τ_E was set to 0.4s.

2.3.3. Generator Model

In the linearized model of AVR system, the transfer function relating the generator terminal voltage to its field voltage can be represented by a gain K_G and a time constant τ_G.

$$\frac{V_F(s)}{V_E(s)} = \frac{K_G}{1+\tau_G S} \tag{8}$$

Here the constants depend on the load; the value of K_G varies between 0.7 to 1, and generator time constant τG in the range of 1s to 2s from full load to no load. In our simulation, K_G was set to 1 and τ_G was set to 1s.

2.3.4. Sensor Model

The sensor is modeled by a simple first order transfer function, given by:

$$\frac{V_S(s)}{V_F(s)} = \frac{K_R}{1+\tau_R S} \tag{9}$$

where the gain K_R is usually kept at 1 and the time constant τ_R is very small, ranging from 0.01s to 0.06s. In our simulation, K_R was set to 1 and τ_R was set to 0.01s.

2.4. AVR System with PID Controller

The above models provide an AVR system compensated with a PID controller block diagram, which is shown in Figure 6.

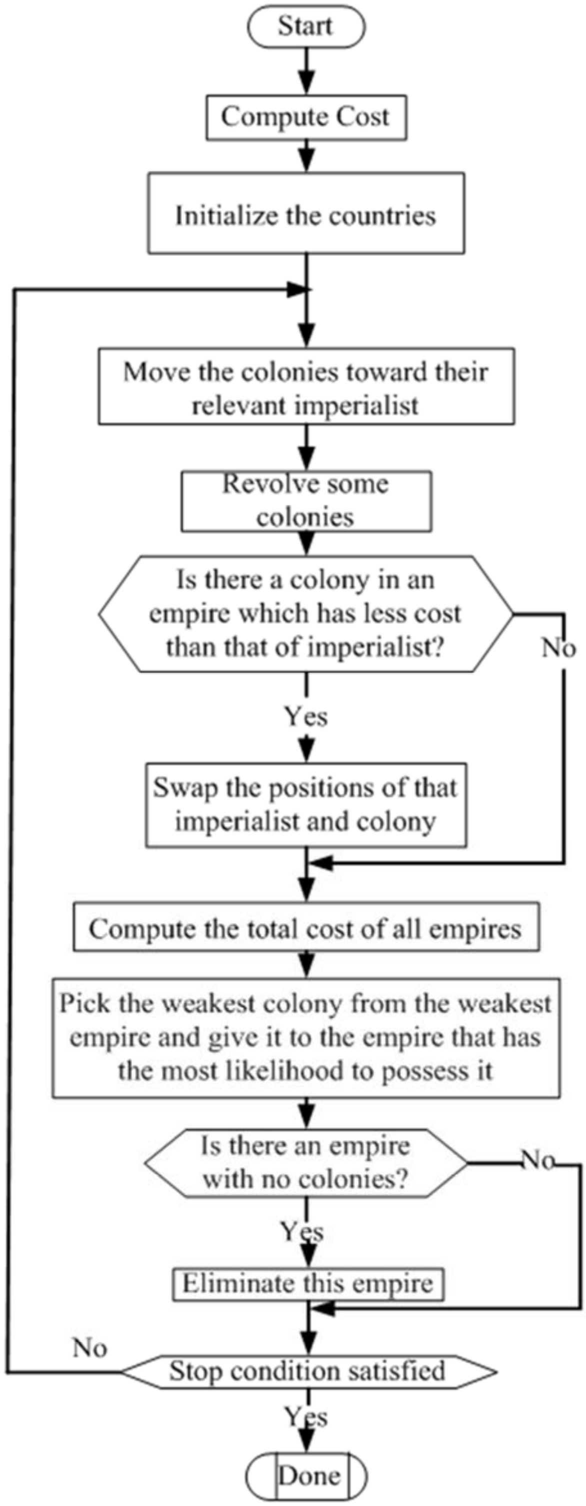


Figure 7. *Flowchart of the ICA applied to the problem of designing a PID controller.*

Designing Optimal PID Controller Using ICA

In this study, a PID controller using the ICA algorithm was developed to improve the transient step response of an AVR system. So it is also called the ICA-PID controller. The ICA algorithm was mainly utilized to determine three optimal controller parameters K_p, K_i and K_d, such that the controlled AVR system could obtain a good step response output.

In this section, a new performance criterion in the time domain is proposed for evaluating the PID controller. A set of good control parameters, and can yield a good step response that will result in performance criteria minimization in the time domain. These performance criteria in the time domain include the overshoot *Mp*, rise time *tr*, settling time *ts*, and integral absolute error IAE. Therefore, a new performance criterion *Cost (K)* is defined as follows:

$$Cost(K) = (1-\alpha)(MP + IAE) + \alpha(ts + tr) \qquad (10)$$

where *K* is [K_p, K_i and K_d], the range of the parameter values as shown in Table 1 and α is the weighting factor. The performance criterion *Cost (K)* can satisfy the designer' requirements using the weighting factor α value. In this paper, α is set in the range of $\frac{1}{6}$ to $\frac{3}{4}$. Figure 7 depicts the flowchart of the ICA used to design the optimal controller.

3. Numerical Examples and Results

To verify the efficiency of the ICA-PID controller, a practical high-order AVR system was tested. The block diagram of the AVR system with a PID controller is shown in Figure 8.

Constant parameters for proper operation of ICA used in this work are shown in Table 2.

3.1. Performance of AVR System Without PID Controller

AVR system shows a very weak performance in absence of PID controller. Figure 9 shows the terminal voltage step response of the AVR system without using PID controller. We found the output parameters of system as follows: *Mp* =0.5047, IAE =4.34011, *tr* =0.2627*s*, *ts* = 6.9874*s*.

3.2. Performance of the ICA-PID Controller

In this section, we represent the performance of the proposed ICA-PID controller in terms of overshoot (*Mp*), integral absolute error (IAE), rise time (*tr*) and settling time (*ts*). The ICA-PID controller shown in Figure 8 then replaced the PID controller. In order to have a better view of competition process, initial parameter values and data are represented as the following: The initial population of 200 countries is shown in Figure 10. Initial number of countries is 200 which 12 of the best ones are chosen to be the imperialists and control others. The imperialists are shown by star mark in different colors and the colonies of each imperialist are shown by • in the same color as the imperialist. Figures 10, 11 and 12 illustrate the initial empires, empires at iterations 100 and 200, respectively. The simulation results that showed the best solution were summarized in Table 3. Figures 13–16 showed convergence characteristics of the ICA-PID controller and terminal voltage step response of the AVR system at different simulation conditions, respectively.

As we can see from the convergence characteristics of best *Cost* value, ICA algorithm reaches to a satisfactory result within 40 iterations. Also the smooth convergence characteristic of the algorithm indicates that the proposed ICA-PID controller is able to improve terminal voltage step response of the AVR system without much fluctuation.

3.3. Comparison of Four Proposed Controllers

In order to emphasize on advantages of the proposed ICA-PID controller, we also implemented the PSO-PID, GA-PID and the SFLA-PID controllers derived from the real-value PSO [18], GA [19], SFLA [20-22] and ICA [23-29] methods.

3.3.1. Terminal Voltage Step Response

There were four simulation examples to evaluate the performance of every four ICA-PID, the PSO-PID, the GA-PID and the SFLA-PID controllers. In each simulation example, the weighting factor ⍰ in the performance criterion and the number of iterations were shown in Table 4.

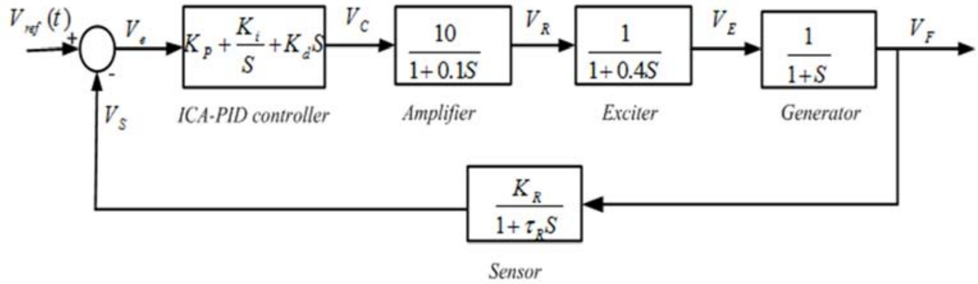


Figure 8. *Block diagram of an AVR system with an ICA-PID controller.*

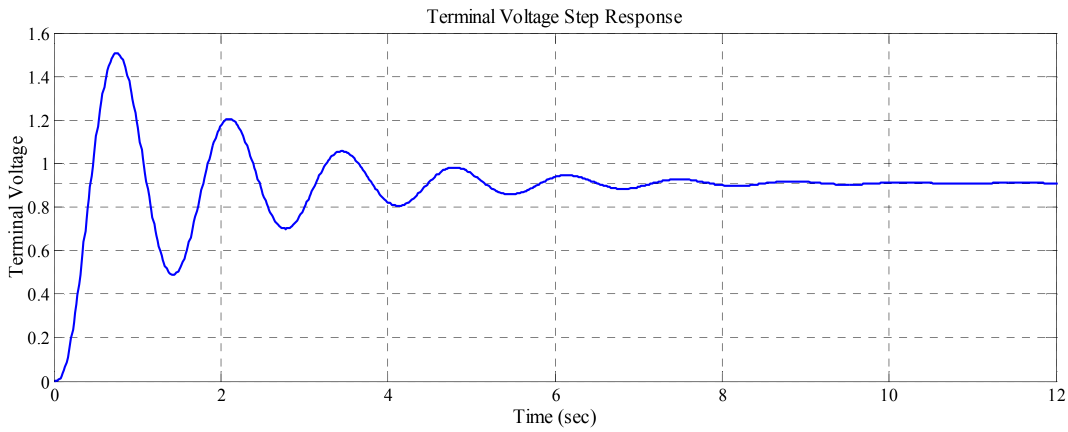


Figure 9. *Terminal voltage step response of an AVR system without PID.*

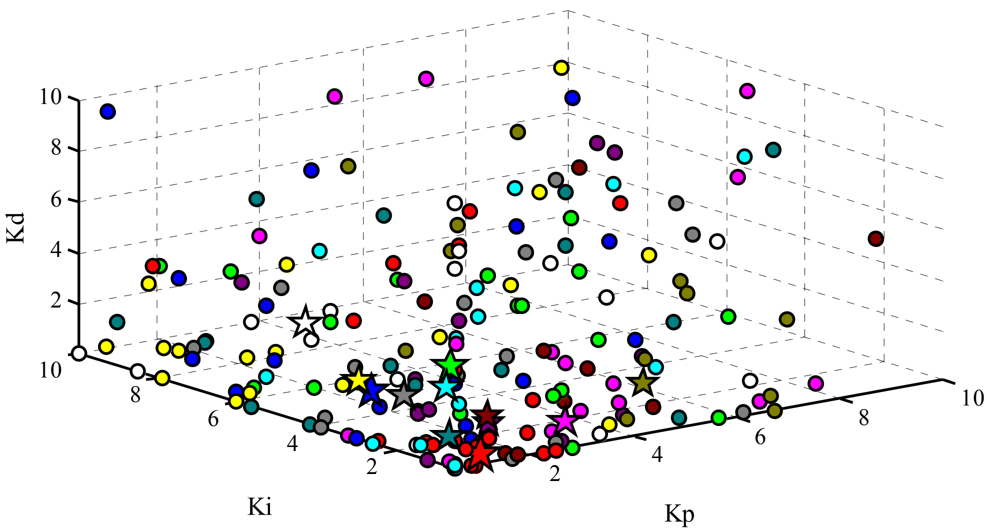


Figure 10. *Initial empires for designing a PID controller (⍰=0.225).*

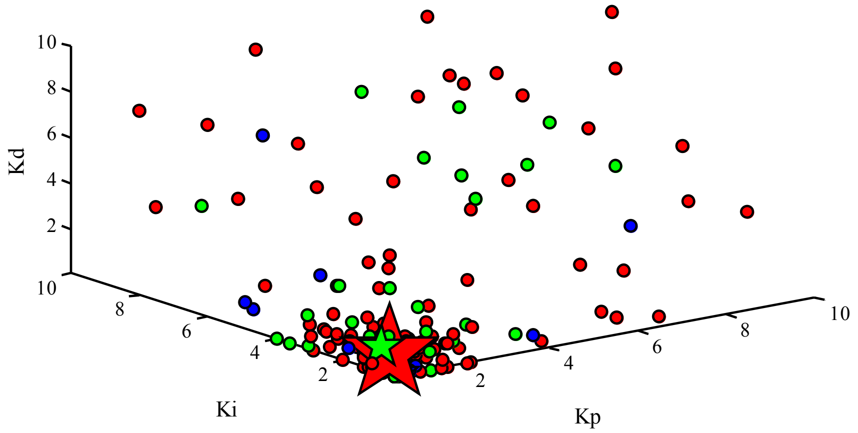


Figure 11. *Empires at iteration 30(=0.225).*

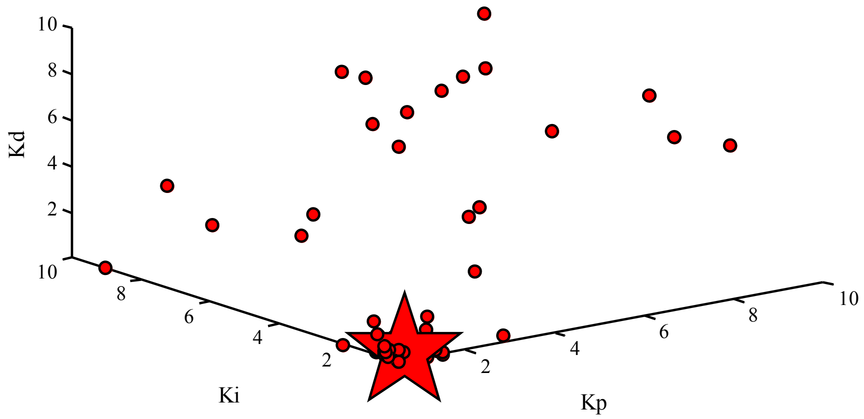


Figure 12. *Final solution (=0.225).*

Figures 17–20 show the terminal voltage step response of the AVR system in the four simulation examples, indicating that the ICA-PID controller is better than the GA-PID and SFLA-PID controllers and somewhat better than the PSO-PID controller. Table 5 shows the four performance criteria in the time domain of each example. As can be seen, all four controllers gave good PID controller parameters in each simulation example, providing good terminal voltage step response of the AVR system. As revealed by the above four performance criteria, the ICA-PID controller has better performance than the PSO-PID, the GA-PID and the SFLA-PID controllers.

Table 1. *Range of three controller parameters.*

Controller Parameters	Min. Value	Max. Value
K_p	0.5	10
K_i	0.5	10
K_d	0.1	10

Table 2. *Parameters of ICA.*

Number of Countries	200
Number of Empires	12
Assimilation Coefficient β	2
Assimilation Angle (rad) γ	0.5

Table 3. *Best solution using ICA-PID controller with the different values.*

	Iteration	Kp	Ki	Kd	Mp	IAE	tr	ts	Cost
0.225	200	1.01860	1.3970	0.5014	0.1343	0.2182	0.1201	0.4403	0.3993
0.350	200	1.01545	1.3900	0.5000	0.1335	0.2184	0.1200	0.6400	0.4247

Table 4. *Weighting factor and number of iterations.*

Simulation Example		Number of Iterations
Example I	0.225	50
Example II	0.225	100
Example III	0.350	50
Example IV	0.350	100

3.3.2. Convergence Characteristic

The convergence characteristics graph is presented in Figure 21. As can be seen, the ICA-PID controller has better Cost value than the PSO-PID, the GA-PID and the SFLA-PID controllers.

In addition, the maximum and minimum Cost values were obtained and the results are shown in Figure 22 and Table 5.

The Cost values of the ICA-PID controller generated fluctuation in a small range (ΔCost=0.0003), thus verifying that the ICA-PID controller has better convergence characteristic.

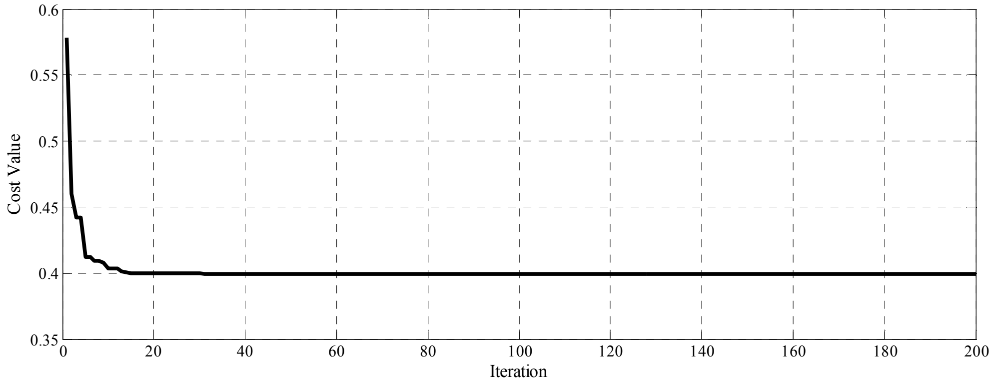


***Figure 13.** Convergence tendency of the ICA-PID controller (=0.225).*

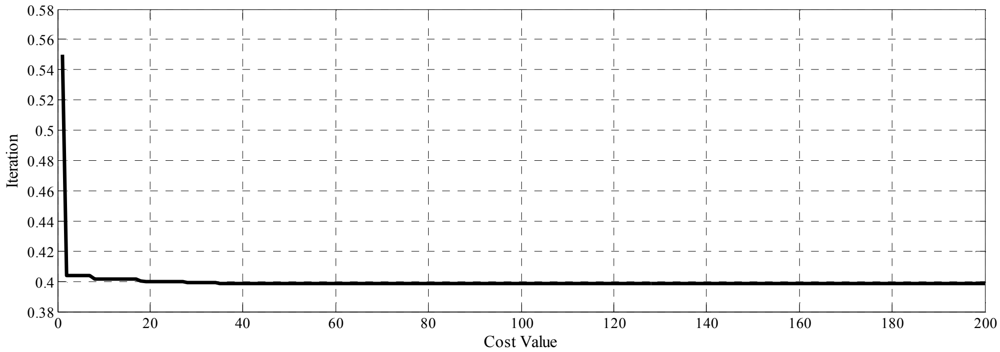


***Figure 14.** Convergence tendency of the ICA-PID controller (=0.35).*

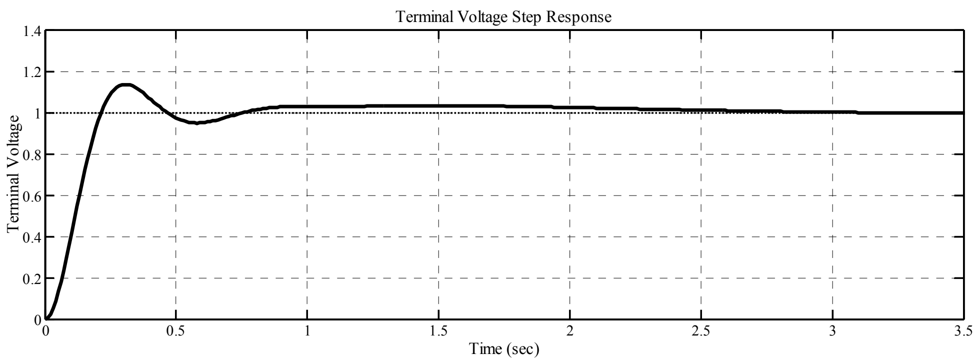


***Figure 15.** Terminal voltage step response of an AVR system with the ICA-PID controller (=0.225).*

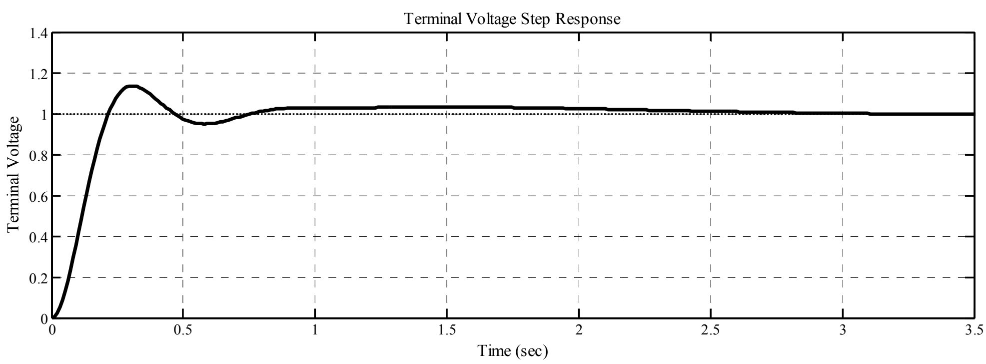


***Figure 16.** Terminal voltage step response of an AVR system with the ICA-PID controller (=0.35).*

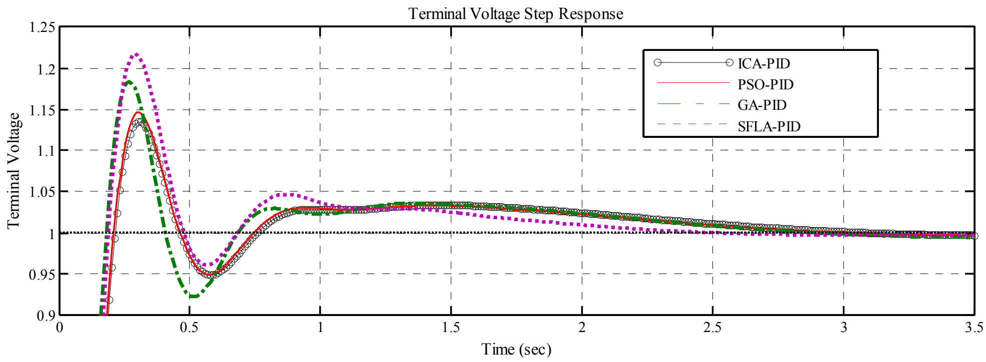


Figure 17. *Terminal voltage step response of an AVR system with different controllers (Example I, =0.225, iterations=50).*

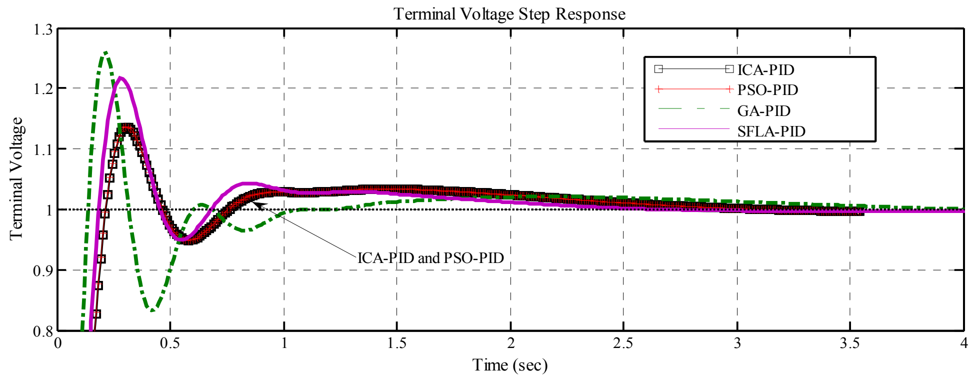


Figure 18. *Terminal voltage step response of an AVR system with different controllers (Example II, =0.225, iterations=100).*

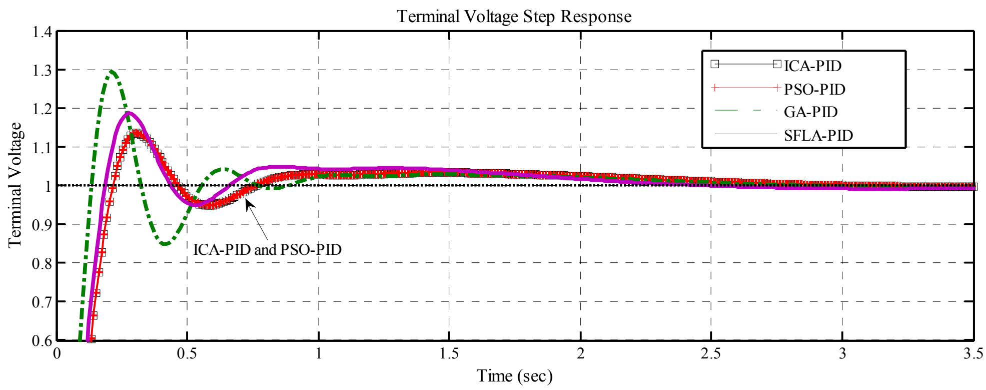


Figure 19. *Terminal voltage step response of an AVR system with different controllers (Example III, =0.35, iterations=50).*

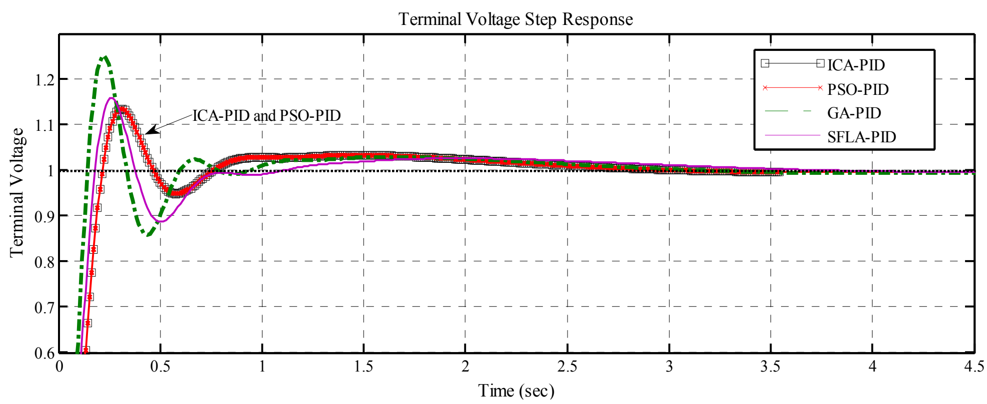


Figure 20. *Terminal voltage step response of an AVR system with different controllers (Example IV, =0.35, iterations=100).*

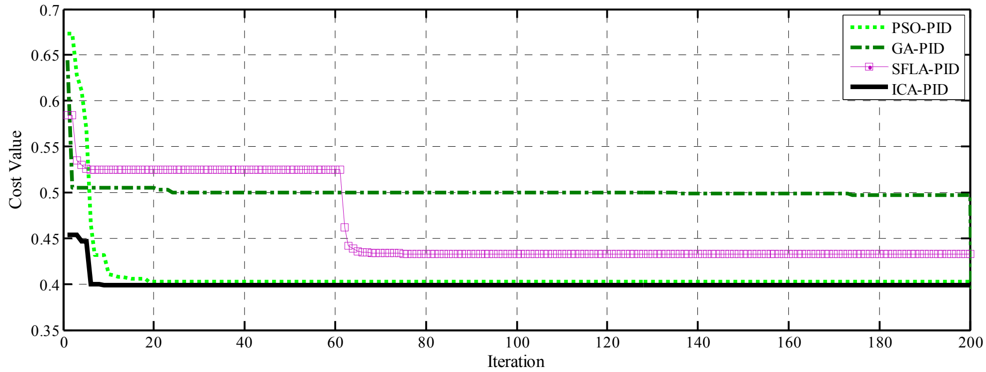


Figure 21. *Convergence tendency of the Cost value of four methods (α=0.255).*

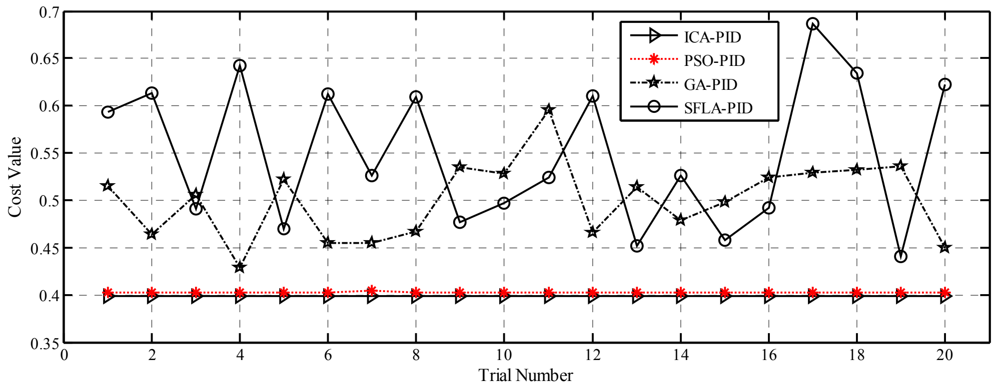


Figure 22. *Comparison of the statistical Cost values of every four methods (α=0.255 and iteration=50).*

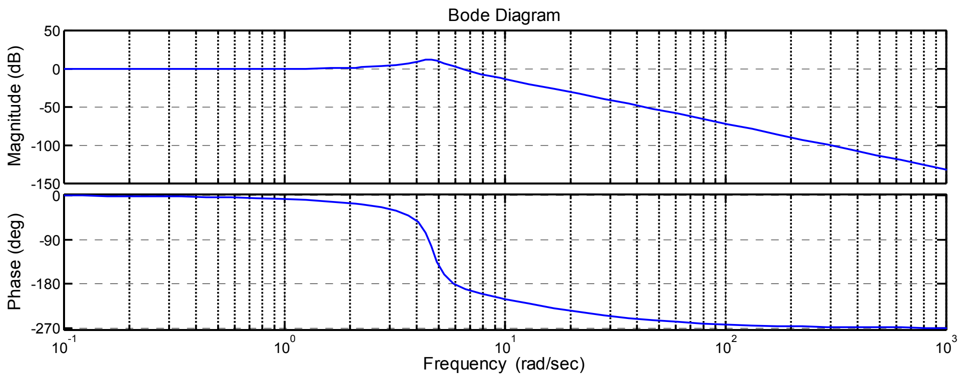


Figure 23. *Bode diagram of the system AVR without controller.*

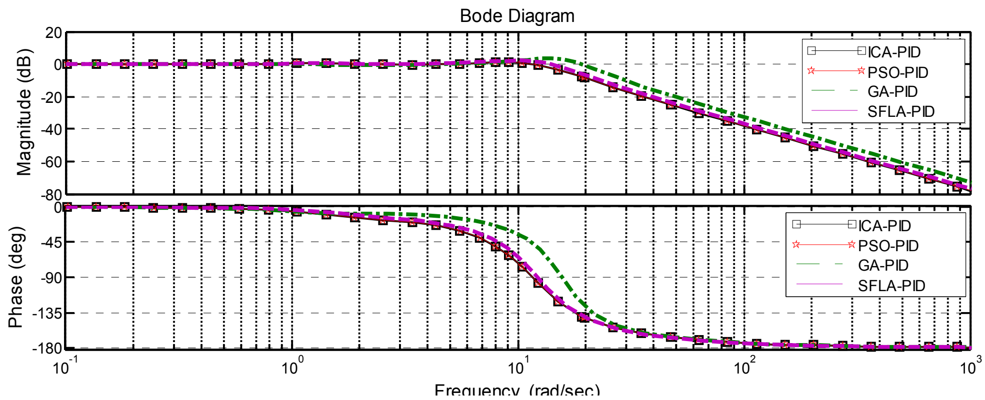


Figure 24. *Bode diagram of the system AVR with ICA-PID, PSO-PID, GA-PID and SFLA-PID controllers (α=0.225 and iteration=100).*

3.3.3. Bode Stability Analysis

Application Bode diagrams of system frequency responses are provided to assess the relative stability of a closed-loop system given the frequency response of the open-loop system. By analyzing the frequency response, you can determine what the open-loop and closed-loop frequency responses of a system imply about the system behavior. The Bode magnitude diagram shows the gain plotted against the frequency. The Bode phase diagram shows the phase, in degrees, as a function of the frequency.

The results were shown in Figures 23–24 and Table 6. As can be seen, all four controllers could give good frequency response, providing complete stability of the AVR system.

Table 5. *Comparison between ICA-PID, PSO-PID, GA-PID and SFLA-PID controllers*

Example		Iterations	Type of Controller	*Kp*	*Ki*	*Kd*	*Mp*	*IAE*	*tr*	*ts*	*Cost*
Example I	0.225	50	PSO-PID	1.0592	1.4639	0.5122	0.1434	0.2163	0.1204	0.4400	0.4047
			GA-PID	1.1420	1.7968	0.6239	0.1812	0.2160	0.1192	0.6007	0.4139
			SFLA-PID	1.3783	2.0171	0.5540	0.2149	0.2090	0.1200	0.4400	0.4546
			ICA-PID	1.0150	1.3940	0.5004	0.1336	0.2185	0.1188	0.4402	0.3990
Example II	0.225	100	PSO-PID	1.0200	1.3997	0.5018	0.1345	0.2182	0.1203	0.3998	0.3995
			GA-PID	1.2128	1.5876	0.9269	0.2532	0.2046	0.1197	0.5600	0.5078
			SFLA-PID	1.3774	1.9668	0.5780	0.2168	0.2072	0.1207	0.4980	0.4726
			ICA-PID	1.0155	1.3900	0.5000	0.1335	0.2184	0.1200	0.3982	0.3987
Example III	0.35	50	PSO-PID	1.0166	1.3896	0.5000	0.1337	0.2182	0.1210	0.4400	0.4250
			GA-PID	1.4576	2.5797	0.9500	0.2884	0.2052	0.1197	0.5000	0.5450
			SFLA-PID	1.1566	2.1325	0.5931	0.1871	0.2243	0.1200	0.5600	0.5054
			ICA-PID	1.0158	1.3899	0.5000	0.1336	0.2183	0.1200	0.4400	0.4247
Example IV	0.35	100	PSO-PID	1.0157	1.3950	0.5000	0.1337	0.2186	0.1235	0.3965	0.4250
			GA-PID	1.2607	2.0782	0.8640	0.2407	0.2097	0.1200	0.6000	0.5307
			SFLA-PID	0.9866	1.2767	0.6497	0.1505	0.2150	0.1200	0.6400	0.5036
			ICA-PID	1.0155	1.3900	0.5000	0.1335	0.2184	0.1222	0.4378	0.4247

Table 6. *Phase margin and gain margin (=0.225 and Iteration =100).*

Type of Controller	Phase Margin (deg)	Gain Margin (dB)
Without Controller	-5.34	-2
ICA-PID	86.4	Inf
PSO-PID	86.1	Inf
GA-PID	56.4	Inf
SFLA-PID	71.9	Inf

4. Conclusions

We used the PSO, GA, SFLA and ICA algorithms to design an optimal PID controller for an AVR system. The PID controller is designed in such a way that it minimizes the sum of settling time, rise time, maximum overshoot and integral absolute error. The simulation results which are obtained over a practical AVR system clearly indicate that the ICA can perform an efficient search for optimal value of the controller parameters compared to PSO, GA and SFLA algorithms and show that the ICA is a successful technique for optimization and designing PID controllers. As was shown Earlier, the ICA-PID controller has also smaller *Cost* value than the PSO-PID, the GA-PID and the SFLA-PID controllers.

References

[1] Devaraj D, Selvabala B (2009) Real-coded genetic algorithm and fuzzy logic approach for real-time tuning of proportional–integral–derivative controller in automatic voltage regulator system. *IET Generation, Transmission & Distribution* 3, 641–9.

[2] Bingul Z,Karahan O (2011) Tuning of Fractional PID controllers using PSO algorithm for robot trajectory control. In: *Mechatronics – ICM 2011*, Kocaeli, Turkey, 955–60.

[3] Shabib G, Mesalam AG, Rashwan AM (2010) Optimal tuning of PID controller for AVR system using modified particle swarm optimization. In: *World Scientific and Engineering Academy and Society- WSEAS 201*0, Wisconsin, USA, 104-10.

[4] Madinehi N, Shaloudegi K, Abedi M, Abyaneh HA (2011) Optimum design of PID controller in AVR system using intelligent methods In: *PowerTech 2011*, Tehran, Iran, 1-6.

[5] Valizadeh S, Jamali M, Lucas C (2008) A particle-swarm-based approach for optimum design of BELBIC controller in AVR system. In: *Control, Automation and System- ICCAS 2008*, Tehran, Iran, 2679-84.

[6] Tu CJ, Chuang LY, Chang JY, Yang CH (2007) Feature selection using PSO-SVM. In: *International MultiConference of Engineers and Computer Scientists - IMECS 2006*, 138-43.

[7] Nejad HC, Jahani R, Sarlak GR (2011) Applying shuffled frog leaping algorithm for economic load dispatch of power system. *American Journal of Scientific Research* 20, 82-9.

[8] Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *Proc. IEEE Congress on Evolutionary Computation*, no. 4425083, 4661–7.

[9] G. Mojtaba, G. Sahand, A. Jamshid, G. Mohsen, and F. Hasan, Application of chaos-based chaotic invasive weed optimization techniques for environmental OPF problems in the power system, Chaos, Solitons & Fractals, 2014, 69:271–284.

[10] Mojtaba G, Sahand G, Ebrahim AA, Azizi V. Solving non-linear, non-smooth and non-convex optimal power flow problems using chaotic invasive weed optimization algorithms based on chaos. Energy. 2014;73:340–353.

[11] Ghasemi M, Ghavidel S, Rahmani S, Roosta A, Falah H. A novel hybrid algorithm of imperialist competitive algorithm and teaching learning algorithm for optimal power flow problem with non-smooth cost functions. Eng Appl Artif Intell 2014;29:54–69..

[12] Ghasemi M, Ghavidel S, Ghanbarian MM, Massrur HR, Gharibzadeh M. Application of imperialist competitive algorithm with its modified techniques for multi-objective optimal power flow problem: a comparative study. Inform Sci 2014.

[13] Ghasemi M, Ghavidel S, Ghanbarian MM, Habibi A. A new hybrid algorithm for optimal reactive power dispatch problem with discrete and continuous control variables. Appl Soft Comput 2014.

[14] Ghasemi M, Ghavidel S, Ghanbarian MM, Gitizadeh M. Multi-objective optimal electric power planning in the power system using Gaussian bare-bones imperialist competitive algorithm. Inform Sci 2015;294:286–304.

[15] Ghasemi M, Roosta A. Imperialist competitive algorithm for optimal reactive power dispatch problem: a comparative study. J World's Electr Eng Technol 2013; 2:13–20.

[16] Saadat H, eds 1999 (2005) Power System Analysis. Mcgraw-Hill International edition.

[17] Mandal A, Zafar H, Ghoshl P, Das S, Abraham A (2011) An efficient memetic algorithm for parameter tuning of PID controller in AVR system. In: *Hybrid Intelligent System- HIS 2011*, Jadavpur Univ, Kolkata, 265-70.

[18] Zamani M, Karimi-Ghartemani M, Sadati N, Parniani M (2009) Design of a fractional order PID controller for an AVR using particle swarm optimization. *Control Engineering Practice* 17, 1380–7.

[19] Lai LL, Ma JT (1997) Improved genetic algorithms for optimal power flow under both normal and contingent operation states. *International Journal of Electrical Power & Energy Systems* 19, 287-92.

[20] Li Y, Zhou J, Yang J, Liu L, Qin H, Yang L (2008) The chaos-based shuffled frog leaping algorithm and its application. In: *Natural Computation- ICNC 2088*, Huazhong Univ. of Sci. & Technol., Wuhan1, 481-5.

[21] Ota T, Omatu S (1996) Tuning of the PID control gains by GA. In: *Emering Technologies and Factory Automation- HIS 1996*, Osaca Prefectural Univ, Sakai 1, 272-4.

[22] Huynh T (2008) A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers. In: *Industrial Technology- ICIT 2008*, Univ. of Technol, Ho Chi Minh City, 1-6.

[23] M. Ghasemi, J. Aghaei, E. Akbari, S. Ghavidel, and L. Li, "A differential evolution particle swarm optimizer for various types of multi-area economic dispatch problems," Energy, vol. 107, pp. 182–195, 2016.

[24] M. Ghasemi, S. Ghavidel, M. M. Ghanbarian, et al., Multi-objective optimal power flow considering the cost, emission, voltage deviation and power losses using multi-objective modified imperialist competitive algorithm, Energy 78 (2014) 276–289.

[25] M. Ghasemi, M. Taghizadeh, S. Ghavidel, A. Abbasian, Colonial competitive differential evolution: An experimental study for optimal economic load dispatch, Applied Soft Computing 40 (2016) 342–363.

[26] M. Ghasemi, M. M. Ghanbarian, S. Ghavidel, S. Rahmani, E. Mahboubi- Moghaddam, Modified teaching learnin galgorithm and double differential evolution algorithm for optimal reactive power dispatch problem: a comparative study, Inf. Sci. 278(2014) 231–249.

[27] M. Ghasemi, S. Ghavidel, M. Gitizadeh, and E. Akbari, "An improved teaching–learning-based optimization algorithm using Levy mutation strategy for non-smooth optimal power flow," Int. J. Electr. Power Energy Syst. 65, 375 (2015).

[28] Ghasemi M, Taghizadeh M, Ghavidel S et al (2015) Solving optimal reactive power dispatch problem using a novel teaching–learning based optimization algorithm. Eng Appl Artif Intell 39:100–108.