Tuning FPID Controller for an AVR System Using Invasive Weed Optimization Algorithm

Nasir Ahmad Alawad[1*], Nora Ghani Rahman[2]

[1, 2] Department of Computer Engineering, Faculty of Engineering, Almustansiriyah University, Baghdad, Iraq
E-mail: nasir.awad@uomustansiriyah.edu.iq

Received: December 29, 2019 *Revised: January 19, 2020* *Accepted: January 30, 2020*

Abstract – In this paper, a design method to determine the optimal fractional proportional integral derivative (FPID) controller's parameters is proposed for automated voltage regulator (AVR). The AVR model is obtained utilizing two model reduction techniques, namely particle swarm optimization (PSO) and biogeography-based optimization (BBO) with the assistance of MATLAB and Simulink software package. For tuning the FPID controller, invasive weed optimization (IWO) algorithm is used with four unique performance indices: integral square error (ISE), integral time square error (ITSE), integral absolute error (IAE) and integral time weighted absolute error (ITAE). The results show that the IWO-designed FPID - compared to the classical proportional integral derivative controller (PID) - offers a better performance in terms of overshoot, rise time and settling time.

Keywords **–** Automated voltage regulator; Model reduction techniques; Invasive weed optimization algorithm; Fractional proportional integral derivative controller; MATLAB package.

1. INTRODUCTION

Automated voltage regulator (AVR) is a device designed to control voltage automatically, i.e. taking a fluctuating voltage level and transforming it into a constant level. The AVR model is complex and in high order; however, there are many ways for simplifying it to the second order [1]. There are many methods for regulating AVR including classical processes, such as integer proportional integral derivative controller (PID), internal mode control (IMC), and linear quadratic regulator (LQR) [2, 3]. Other methods have used soft techniques in recent centuries, such as fuzzy logic control, genetic algorithms, and neural networks [4].

In the last two centuries, fractional calculus has been quite common research area in engineering studies although it has been recognized for about 300 years [5]. The first research was reported in [6] on the controller of the fractional order proportional integral derivative (FOPID). In the literature, in contrast to the conventional PID; two additional parameters are proposed for the fractional PID's; namely, integration (λ) and differentiation (μ) orders [7]. Researchers have indicated that in terms of certain characteristics such as flexibility and stability; these two parameters reinforce the hand of customers. Authors in [8] developed a fractional PID controller design technique. The fresh tuning technique was based on the classic technique of tuning Ziegler Nichols to set the fractional PID controller parameters. In [9], FOPID was designed for spherical tank liquid level control. The PID controller was designed using the integral square error (ISE) method of minimization. The FOPID controller reaction was contrasted with the simulation and experimental setup of the integer order PID (IOPID) controllers.

* Corresponding author.

Article's DOI: 10.5455/jjee.204-1581071239

There are many distinct methods to find approximations for λ and μ. It is not feasible to tell which one of them is better based on certain behavior [10]. A fresh set of PID and FOPID controllers' tuning laws based on minimizing the embedded absolute error were suggested. It yields a low overshoot and low settling, but limits the maximum sensitivity [11]. By comparing the results obtained from PID and FOPID controllers; fractional-order integral action was not advantageous, while the use of fractional-order derivative action provides an improvement in performance, focused on optimizing the integer order PID and fractional order [12, 13].

The PID controller of fractional order generalizes the PID controller of the integer order and moves it from point to point. This extension brings more flexibility to the design of the controller and it is used more precisely to regulate procedures in the actual globe. In [14], authors evaluated the design of the FOPID controller using the chaotic ant swarm (CAS) optimization technique. FOPID controller tuning emerged as a nonlinear optimization issue, where the objective function consists of over-shooting, steady-state error, rise time, and settling time. In [15], a procedure for tuning perfect PID fragmentary request parameters utilizing particle swarm optimization (PSO) calculations was suggested and a whole number request plans was given. Ideal parameters can be found in fewer cycles utilizing the appropriate wellness work.

The remaining part of this paper is organized as follows: in Section 2, the AVR system is introduced with detailed model and model parameters. In Section 3, the model reduction for AVR by using the biogeography-based optimization (BBO) algorithm approach is presented. Section 4 presents the fractional order PID controller and its mathematical foundation and tuning using invasive weed optimization (IWO) algorithm. In Section 5, discussions, simulation investigations and comparisons of the AVR controlled by optimal fractional proportional integral derivative controller (FPID) are presented to illustrate the superior robustness achieved by using fractional order controller. Finally, conclusions are drawn in Section 6.

2. AVR MATHEMATICAL MODEL

Detailed theoretical description of the AVR system can be found in [16, 17]. An AVR system, which meant to be simple, consists of four main components; amplifier, exciter, generator and sensor as shown in Fig. 1.

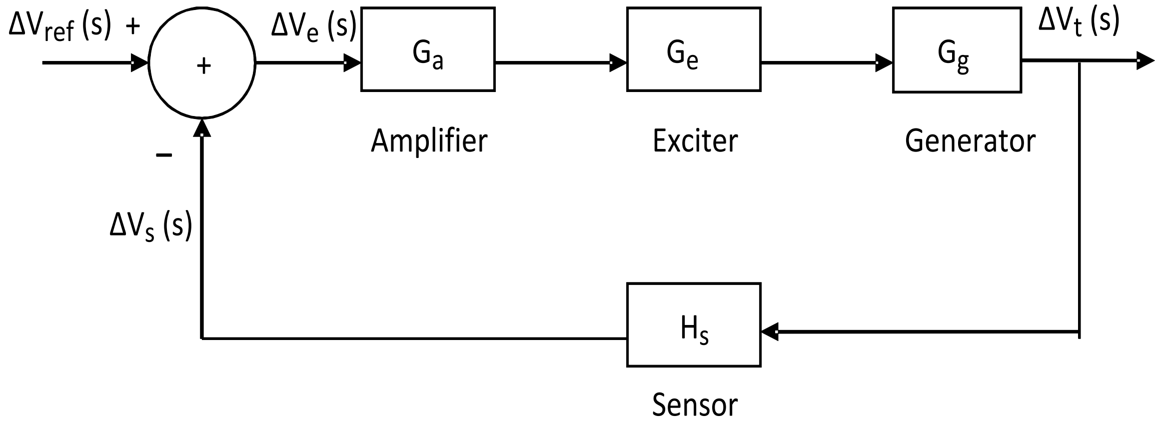

Fig. 1. Block diagram of the AVR system.

By deriving the transfer functions for each component as described in [18]; the whole closed-loop transfer function can be achieved as:

$$\frac{V_t(s)}{V_{ref}(s)} = Go(s) = \frac{0.0599\,s^3 + 5.994\,s^2 + 8.252\,s + 825.2}{0.00573\,s^6 + 0.644\,s^5 + 7.686\,s^4 + 55.571\,s^3 + 465.86\,s^2 + 879.208\,s + 1086} \tag{1}$$

Its poles are: -99.8497 + 0.0000i, -10.2222 + 0.0000i, -0.1133 + 7.8699i, -0.1133 - 7.8699i, -1.0462 + 1.3794i, and -1.0462 - 1.3794i. The dominant poles are: -0.1133 + 7.8699i, and -0.1133 - 7.8699i which are very close to the imaginary axis.

The transfer function of the AVR, as shown in Eq. (1), is sixth order; so we would be able to approximate it to second order, depending on the dominant poles.

3. AVR MODEL ORDER REDUCTION

Model reduction is a method for acquiring a decreased order model that maintains certain significant characteristics such as stability. It is close, in some sense, to the initial model in terms of overshoot percentage (Mp%), rise time (Tr), settling time (Ts), and steady-state error (Ess). Model order reduction (MOR) methods are classified into time domain and frequency domain [19-21]. Nowadays, many methods for decreasing order model are dependent on the optimization techniques such as PSO, genetic algorithm (GA), differential evolution (DE), and GA-based MOR which gained some attention from the scientists.

Recently, a PSO method has been used to achieve a decreased order model for large-scale linear single-input and single-output (SISO) schemes [22]. The technique is based on ISE, while a blended method for decreasing the order of linear large-scale structures was proposed in [23]. When attempted on a specific scheme; each technique has its own benefits and disadvantages. Furthermore, no strategy always delivers the highest outcomes of all systems. In this context, the BBO method is performed with the upsides of holding the accurate predominant elements in the structured model, acquiring another hearty model with a lower order, and keeping up a base enduring state reaction error. In this paper, both BBO and PSO approaches are utilized for MOR.

3.1. Biogeography-Based Optimization

It is a fresh form of evolutionary algorithm (EA) based on population. Biogeography is a biology branch. It is a synthetic discipline that depends highly on earth science, population biology, systematic, and ecology theory and information collection [24]. It studies the migration of species from less to more habitable locations between islands and how they share data with others through probability-based migration. In biogeography, the movement of species from one island to another relies on suitability index factors including water resources, vegetation diversity, temperature, and land area. These index factors are depicted as real number vectors. Many scientists have implemented the BBO in order to perform optimization in several applications. The BBO algorithm is used to tune a PID controller's parameters earlier. In this research, a technique for implementing the BBO algorithm is used. It has better search velocity and optimization compared to an AVR model order reduction algorithm with PSO. The BBO algorithm is comprised of two major sub-algorithms: migration and mutation [25].

3.1.1. Migration

In this stage - known as the modification ratio - the arrangement is adjusted. In order to alter a specific arrangement, its movement rate is utilized to –probabilistically - choose whether or not to adjust every suitability index variable (SIV) in that arrangement. On the off chance that a given SIV in a given arrangement is chosen to be changed; we utilize the resettlement paces of different answers to – probabilistically - choose which of the arrangements ought to relocate an arbitrarily chosen SIV [26].

3.1.2. Mutation

The term mutation alludes to an abrupt change or an unexpected variety. With regards to BBO, it shows variety in the populace because of unconstrained changes. Change rate can be determined as pursues [26]:

$$m(s) = m_{\max}\left(\frac{1-P_s}{p_{\max}}\right) \quad (2)$$

where m_{max} is user-defined parameter and Ps is the probability of each island containing S species.

We need to create random values by using the mutation factor. This mutation has likelihood and we have to make the changes based on this. If this probability of mutation exceeds the random value; the mutated value must be modified. Thus; we arrive at the best solution. We have both the migration and the mutation update to prevent the pre-convergence problem. In [26], the performance index is used to decrease the mistake between the initial and decreased scheme.

The steps used in BBO Algorithm [26] are:

1. Initialize a set of solutions to a problem.
2. Compute "fitness" (HSI) for each solution, where HSI is the habitat suitability index of the environment.

$$J=\sum_{i=0}^{n}[y(ti)-yk(ti)] \quad (3)$$

3. Compute S, λ, and μ for each solution.

$$\lambda = L*(1-\frac{S}{S_{\max}}) \quad (4)$$

$$\mu = \frac{E*s}{s_{\max}} \quad (5)$$

where λ = immigration rate, μ = emigration rate, S_{max} = maximum number of species, E = maximum emigration rate, and L =maximum immigration rate.

4. Modify habitats (migration) based on λ, μ .
5. Mutation.
6. Typically, implementation of elitism.
7. Go to step 2 for the next iteration if needed.

The parameters of BBO algorithm in this work are considered as follows: Generation count limit =150, population size=50, mutation probability = 0.06, number of elites = 2.

Referring to Eq. (1) of the original system and using all solution steps of BBO algorithm by MATLAB program; it is found that the reduced function is:

$$R_{BBo}(s) = \frac{0.00032174s + 2.4014}{1.2773s^2 + 2.3858s + 3.1713} \tag{6}$$

The dominant poles are: -0.9339 + 1.2691i, -0.9339 - 1.2691i.

3.2. Particle Swarm Optimization

PSO is a robust evolutionary strategy inspired by the social behavior of animal species living in large colonies like birds, ants, or fish. The PSO algorithm maintains multiple potential solutions at one time. During each iteration of the algorithm, each solution is evaluated by an objective function to determine its fitness, and each solution is represented by a particle in the fitness landscape (search space). The particles "fly" or "swarm" are used through the search space to find the maximum value returned by the objective function (ISE) [6].

The PSO algorithm consists of three steps:

1. Evaluate fitness of each particle.
2. Update individual and global bests.
3. Update velocity and position of each particle.

These steps are repeated until some stopping condition is met. The PSO algorithm parameters are regarded as follows in this research:

1. Inertial weight: 0.9.
2. Acceleration factors c1 and c2 are 0.12 and 1.2, respectively.
3. Population size: 50
4. Maximum iteration: 150

Referring to Eq. (1) of the original system and using all solution steps of PSO algorithm by MATLAB program; it is found that the reduced function is:

$$R_{PSO}(s) = \frac{-0.35002s + 3.6394}{1.704s^2 + 3.583 + 5.138} \tag{7}$$

The dominant poles are: -1.0513 + 1.3820i, -1.0513 - 1.3820i.

When comparing the original system's dominant poles; it is clear that $R_{PSO}(s)$ is very close to it, but the numerator coefficients appears non-minimum phase. Fig. 2 shows the transient response of original and reduced model with BBO and PSO. Response by PSO is non-minimum phase system as derived from Eq. (7), and this response - for a step input - has an undershoot of 1.2242%. It is accepted that within the field of control engineering; non-minimum stage procedures are a troublesome study space.

In some important industrial procedures, non-minimum phase zeros inevitably occur, where the relative stability of PSO is less than BBO because there is a zero in the right hand side of S-plane. Therefore, BBO approach is used in this paper.

When comparing the transient parameters (Mp%, Ts, Tr, and Ess) of the original system Go(s) with reduced systems $R_{PSO}(s)$ and $R_{BBO}(s)$; it is clear that $R_{BBO}(s)$ is almost identical to Go(s) as shown in Table 1 where Tp is the peak time.

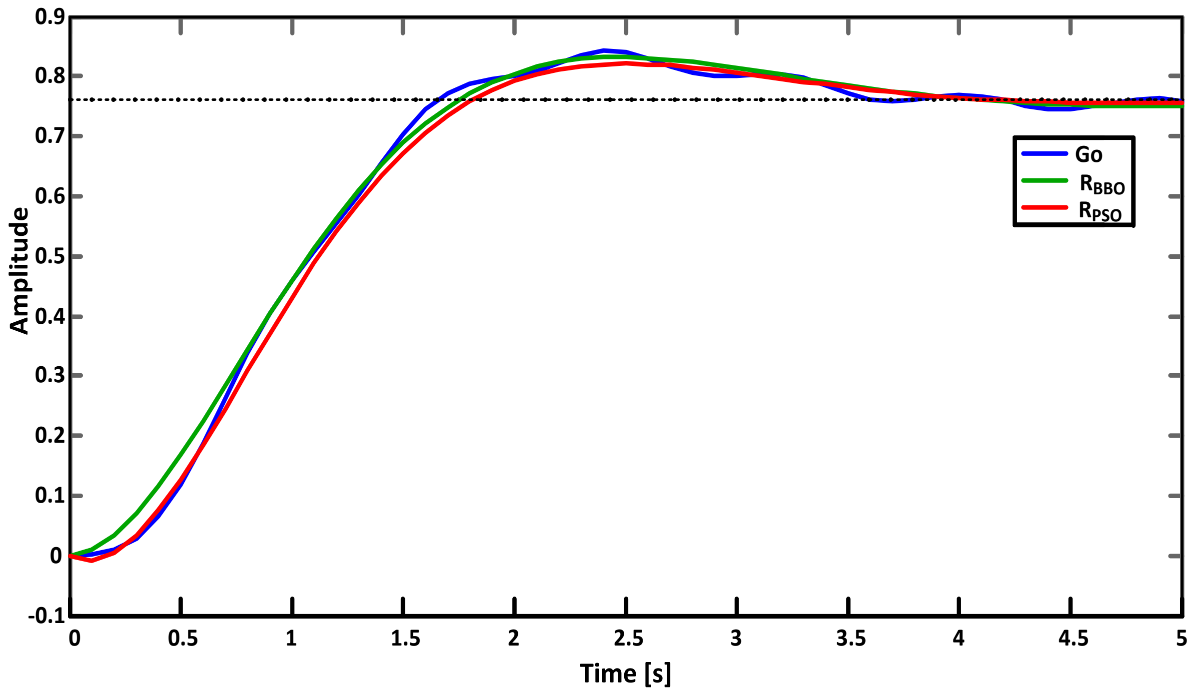

Fig. 2. Transient responses of the original and the reduced - with BBO and PSO - model.

Table 1. Results of comparison between the transient response parameters.

System	Mp [%]	Ts [s]	Tp [s]	Tr [s]	Ess
Go(s)	10.9	4.5	2.42	1.03	0.76
$R_{PSO}(s)$	9.3	3.52	2.37	1.17	0.76
$R_{BBO}(s)$	9.9	3.76	2.46	1.06	0.757

4. FPID CONTROLLER TUNING

Fractional calculus is regarded for almost 300 years. On the other hand, the concept of FPID is proposed in [6, 7] for the first time. In the literature, a quite few research shows that the controller types with fractional order are more environmentally friendly than the traditional integer order controllers [27, 28]. When compared with the classical PID controllers; the fractional order controllers have two extra manipulate parameters: integration (λ) and differentiation (μ) orders, which enable the controller to provide the greater flexibility and stability. Various processes based on the use of a number of tuning guidelines for FPID controllers are proposed in [29]. Also, the research, primarily based on artificial intelligence techniques for FPID controller, is found in the literature. In [30], PSO algorithm is used for FPID [31] and software of DE algorithm to diagram FPID controllers are given in [32]. Many researchers are using Cuckoo search algorithm (CSA), Tabu search algorithm (TSA), artificial bee colony (ABC), and GA in designing the FPID [33-36]. In this paper, IWO algorithm is used.

4.1. Invasive Weed Optimization

Like most of the algorithms in evolutionary computation; IWO does not need the gradient of the function in its optimization process. From a specific point of view; IWO can be categorized as social as PSO algorithm. IWO is considered as both the mathematical model and the laptop simulation of human lifestyle evolution [37, 38]. The IWO procedure starts evolving with initializing a population. That is, a population of initial options is randomly generated over the answer space. Then, individuals of the population produce seeds relying on their comparative fitness in the population. In other words, the range of seeds for each member varies linearly between S_{min} for the worst member and S_{max} for the fantastic member. These seeds are then randomly scattered - over the search area - via commonly dispensed random numbers with mean equal to zero and an adaptive fashionable deviation. The equation for figuring out the widespread deviation (SD) for every era is presented as [39]:

$$\sigma_{iter} = \frac{(iter_{\max} - iter)^n}{(iter_{\max})^n}(\sigma_{initial} - \sigma_{final}) + \sigma_{final} \tag{8}$$

where $iter_{max}$ is the highest amount of iterations, the present iteration SD is iter and the nonlinear modulation index is n.

The generated seeds are regarded as the prospective alternatives for the next generation accompanied by their parents. Finally, a competitive exclusion is carried out in the algorithm, i.e. the population reaches its maximum after a number of iterations and an elimination mechanism is adopted. The seeds are listed together with their relatives and those with better fitness survive and become reproductive. Fig. 3 shows a flow chart of the IWO algorithm. The parameters of the IWO algorithm - considered in this work - are as follows: S_{min} = 0, S_{max} = 5, $iter_{max}$ = 150, n = 2, initial population size = 2, maximum population size = 10, initial value of standard deviation = 0.5, final value of standard deviation = 0.001.

In order to examine the controller's efficiency, the following four distinct objective features are used: integral square error (ISE), integral time square error (ITSE), integral absolute error (IAE) and integral time weighted absolute error (ITAE). They are provided in Eqs. (9–12).

$$\text{ISE}=\int_0^\infty e^2(t)dt. \tag{9}$$

$$\text{ITSE}=\int_0^\infty t.e^2(t)dt. \tag{10}$$

$$IAE = \int_0^\infty |e(t)|dt \tag{11}$$

$$ITAE = \int_0^\infty t|e(t)|dt \tag{12}$$

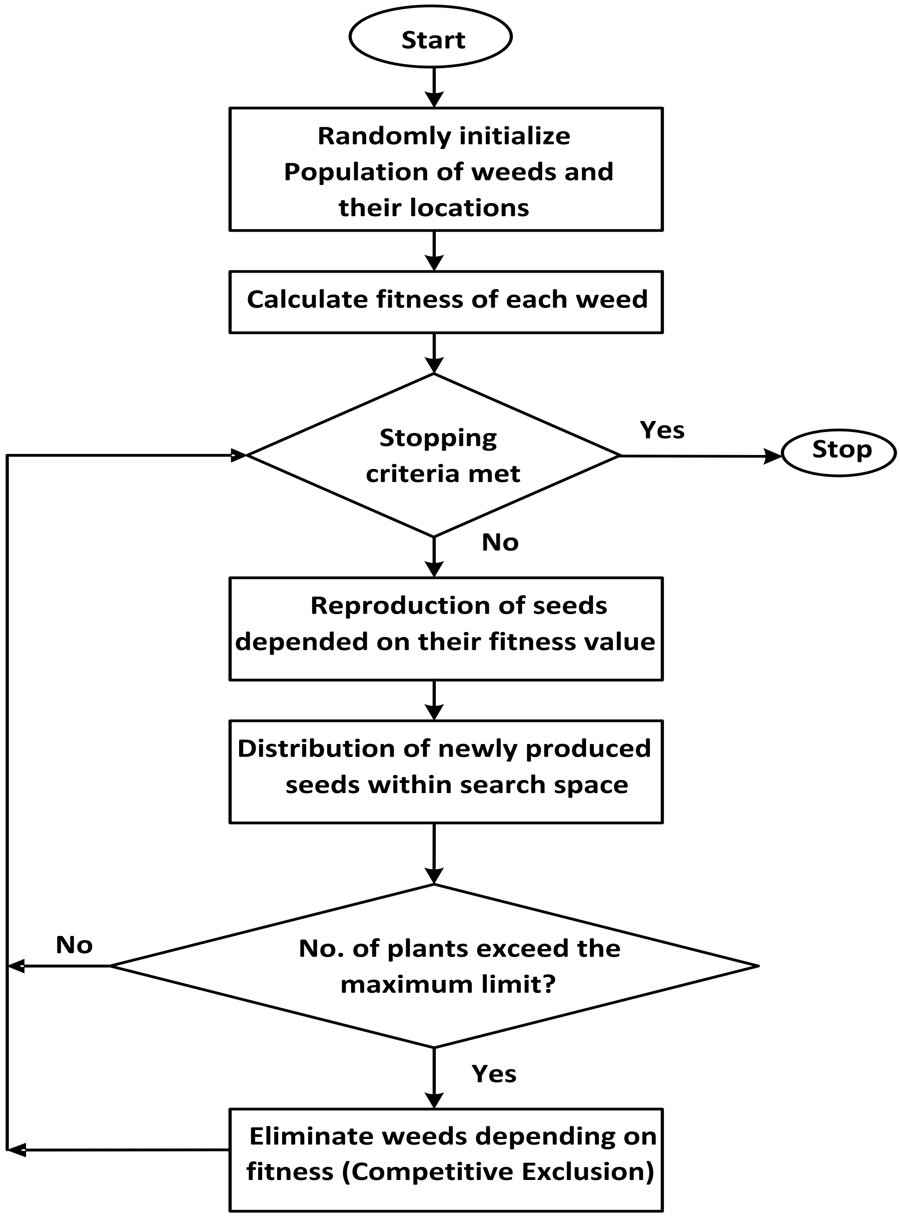

Fig. 3. Flowchart of IWO algorithm.

The continuous transfer function of FPID is also obtained through Laplace transform as:

$$C(s)= Kp+\frac{Ki}{s^{\lambda}}+Kds^{\mu} \tag{13}$$

Firstly, the tuning is executed to obtain the parameters of PID controller K_p, K_i and K_d using MATLAB statement: pidtool (G_o,'pid'), where G_o is the original system. So that K_p = 2.9253, K_i = 2.9774, and K_d = 0.6473, where M_p = 10%, T_s = 3.76 s, T_r = 1.17 s, T_p = 1.07 s and E_{ss} = 0. The designed FPID controller is approximated by oustaloup filter. The fractional derivative and integrals of FPID controller have been approximated by N = 5 and filter frequency [W_l, W_h] = [0.001, 1000]. Table 2 shows the parameters' values of the FPID controller - with different performance indices - obtained using the IWO algorithm, while Table 3 shows the transient parameters with different performance indices.

Table 2. Parameters' values of FPID controller - with different performance indices - obtained using the IWO algorithm.

Performance index	K_p	K_i	λ	K_d	μ
ISE	10	10	0.7893	10	1.4122
ITSE	10	10	1.0010	6.4773	1.0209
IAE	9.6231	10	1.0142	4.4807	1.1263
ITAE	5.1369	8.666	1	3.3364	0.8752

Table 3. Transient parameters' values of FPID controller - with different performance indices – obtained using the IWO algorithm.

Performance index	T_s [s]	T_r [s]	M_p [%]	E_{ss}
ISE	6.7437	2.0701e-07	1.0031	0
ITSE	1.5706	0.2058	0	0
IAE	0.8068	0.3043	0.0026	0
ITAE	0.8176	0.3259	2.1734	0

5. RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed controller and its robust performance based on IWO; firstly, Fig. 4 is presented. It shows a comparison between the transient response of the original system and that of the FPID under different performance indices. It is clear that IAE gives minimum transient values - where the M_p% approaches zero - and it has fast response speed with minimum T_s. To show the effectiveness of the proposed method; a comparison is made between the classical PID and the designed FPID with IAE. The results are depicted in Fig. 5.

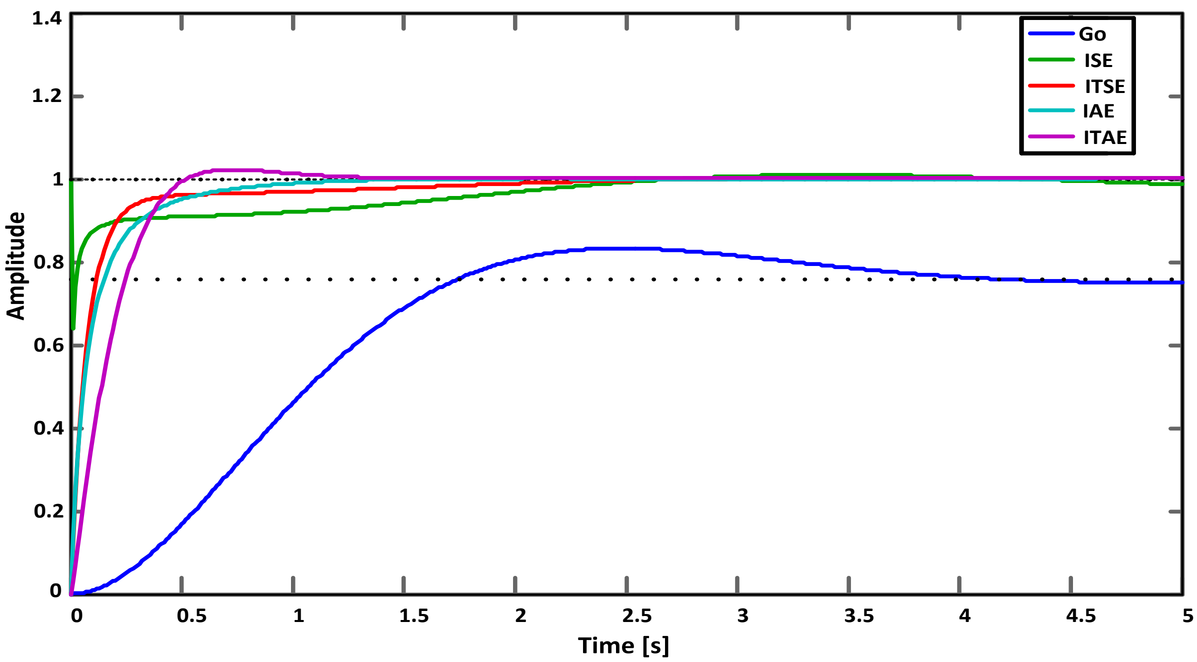

Fig. 4. Transient responses of the original system and the designed - under different performance indices - FPID.

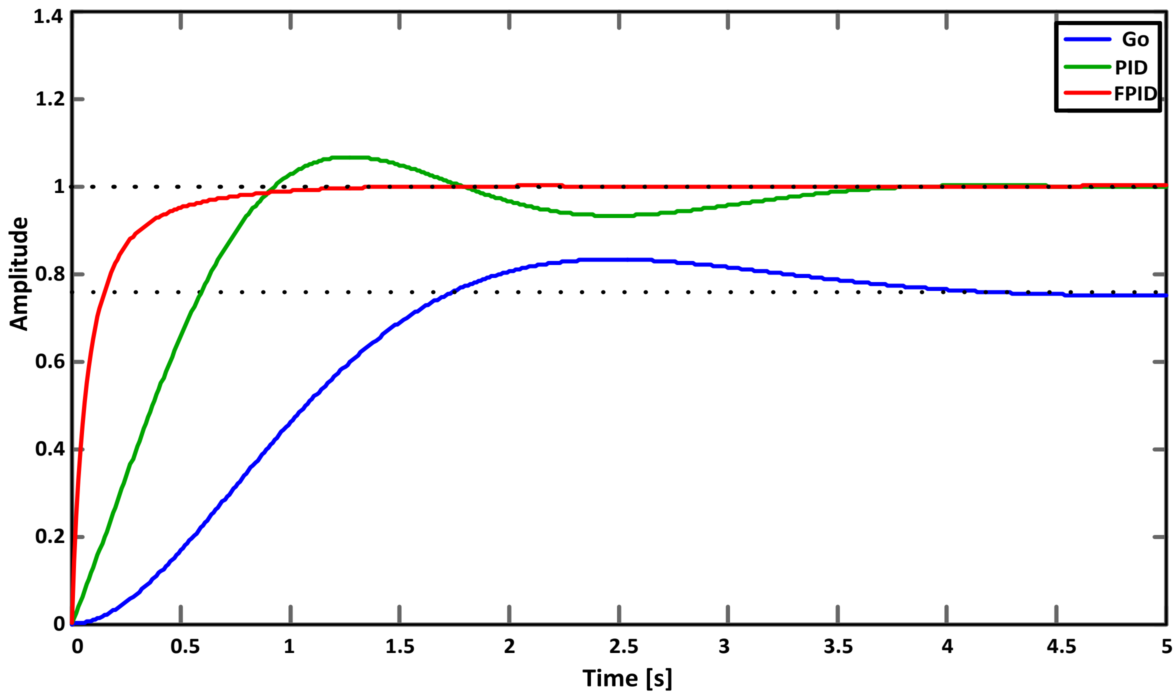

Fig. 5. Transient responses of the original system, classical PID and the designed - with IAE - FPID.

As shown in Fig. 5, there is more improvement in transient response, when using FPID controller with IAE, compared with classical control PID. Overshoot, settling time and AVR speed refraction are decreased by the proposed method (IWO-FPID). Since IWO based controller has a good performance in both time and oscillation damping value; it can be used in practical applications.

6. CONCLUSIONS

In this paper, a design method to determine optimal FPID controller's parameters using the IWO algorithm is proposed. The regulated voltage of AVR is controlled by FPID-IWO controller. The simulation of AVR showed that the proposed controller can perform an efficient search for the optimal FPID controller. The results showed that - compared to the classical PID - the proposed method improves the dynamic performance of the system. The proposed FPID-IWO controller presented better performance and possessed good robustness over PIDs. Moreover, it was found that the proposed FPID-IWO controller converged quicker than the PID. The proposed method introduces accuracy as well as convergence speed and simplicity.

Acknowledgment: The authors would like to thank Al-Mustansiriyah University, Baghdad, Iraq for supporting the present work.

REFERENCES

[1] G. Klempner, I. Kerszenbaum, *Principle of Operation of Synchronous Machine*, John Wiley and Sons, USA, 2004.

[2] P. Mathur, S. Joshi, "PID tuning of automatic voltage regulator," *International Journal of Research and Analytical Reviews*, vol. 2, no. 3, pp. 18-22, 2015.

[3] O. Fernaza, H. Dibyo, "Study of linear quadratic regulator (LQR) control method and its application to automatic voltage regulator (AVR) system," *Innovative Systems Design and Engineering*, vol. 5, no. 10, pp. 50-57, 2014.

[4] A. Thiab, Y. Thaier, "Comparison robustness of automatic voltage regulator for synchronous generator using neural network and neuro-fuzzy controllers," *Iraqi Journal of Computers, Communication and Control Engineering*, vol. 15, no. 2, pp. 1-10, 2015.

[5] R. Garrappa, E. Kaslik, M. Popolizio, "Evaluation of fractional integrals and derivatives of elementary functions: overview and tutorial," *Mathematics*, vol. 7, no. 5, pp. 1-21, 2019.

[6] I. Podlubny, L. Dorcak, I. Kostial, "On fractional derivatives, fractional-order dynamic systems and and PI/sup /spl lambda//D/sup /spl mu//-controllers," *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5, pp. 4985–4990, 1997.

[7] I. Podlubny, "Fractional-order systems and PI/sup/spl lambda//D/sup/spl mu//-controllers," *IEEE Transactions of Automatic Control*, vol. 44, no. 1, pp. 208–214, 1999.

[8] K. Bettou, A. fatah, F. Mesquine, "A new design method for fractional PID controller," *International Journal of Sciences and Techniques of Automatic control and computer engineering*, vol. 2, no. 1, pp. 414-429, 2008.

[9] K. Sundaravadivu , B. Arun , K. Saravanan, "Design of fractional order PID controller for liquid level control of spherical tank," *2011 IEEE International Conference on Control System, Computing and Engineering*, pp. 291-295, 2011.

[10] C. Monje, Y. Chen, D. Xue, B. Vinagre, V. Feliu, *Fractional-order Systems and Controls. Fundamentals and Applications*, Springer, 2010.

[11] F. Padula, A. Visioli , "Tuning rules for optimal PID and fractional-order PID controllers," *Journal of Process Control*, vol. 21, no. 1, pp. 69–81, 2011.

[12] C. Jun-yi, B. Gang Cao, "Design of fractional order controllers based on particle swarm optimization," *2006 1ST IEEE Conference on Industrial Electronics and Applications*, pp. 1-6, 2006.

[13] C. Jun-yi, J. Liang, B. Gang Cao, "Optimization of fractional order PID controllers based on genetic algorithms," *2005 International Conference on Machine Learning and Cybernetics*, vol. 9, pp. 5686-5689, 2005.

[14] T. Yinggan, M. Cui, C. Hua, Y. Yang, L. Li, "Optimum design of fractional order (PID mu)-D-lambda controller for AVR system using chaotic ant swarm," *Expert Systems with Applications*, vol. 39, no. 8, pp. 6887-6896, 2012.

[15] H. Ramezanian, S. Balocchain, "Optimal design a fractional-order PID controller using particle swarm optimization algorithm," *International Journal of Control and Automation*, vol. 6, no. 4, pp. 55-68, 2013.

[16] P. Kundur, *Power System Stability and Control*, McGraw-Hill, New York, 1994.

[17] Z. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, p. 384- 391, 2004.

[18] M. Zamani, M. Ghartemani, N. Sadati, M. Parniani, "Design of a fractional order PID controller for an AVR using particle swarm optimization," *Control Engineering Practice*, vol. 17, no. 12, pp. 1380-1387, 2009.

[19] T. Chen , C. Chang, K. Han, "Reduction of transfer functions by the stability equation method," *Journal of the Franklin Institute*, vol. 308, no. 4, pp. 389–404, 1979.

[20] K. Kumer, S. Nagar, J. Tiwari, "Model order reduction of interval systems using routh approximation and cauer second form," *International Journal of Advances in Science and Technology*, vol. 3, no. 2, pp. 35–41, 2011.

[21] G. Parmar, S. Mukherjee, R. Prasad, "Order reduction of linear dynamic systems using stability equation method and GA," *International Journal of Computer, Information, and Systems Science, and Engineering*, vol. 1, no. 2, pp. 26–32, 2007.

[22] S. Ponda, J. Yadav, N. Padidar , C. Ardil, "Evolutionary techniques for model order reduction of large scale linear systems," *International Journal of Applied Science, Engineering and Technology*, vol. 5, no. 1, pp. 22–28, 2009.

[23] C. Vishwakarma, R. Prasad, "MIMO system reduction using modified pole clustering and genetic algorithm," *Modelling and Simulation in Engineering*, vol. 2009, 2009.

[24] J. Brown, B. Riddle, M. Lomolino, *Bio-geography*, Sinauer Associates, Sunderland, MA, USA, 2009.

[25] G. Kasilingam, J. Pasupuleti, "BBO algorithm-based tuning of PID controller for speed control of synchronous machine," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, no. 4, pp. 3274 – 3285, 2016.

[26] S. Aravind, R. Scaria, "Biogeography-based optimization for the solution of the combined heat and power economic dispatch problem," *International Journal of Engineering and Innovative Technology*, vol. 3, no. 1, pp. 429-432, 2013.

[27] S. Das, A. Gupta, "On the selection of tuning methodology of FOPID controllers for the control of higher order processes," *ISA Transactions*, vol. 50, no. 3, pp. 376-388, 2011.

[28] M. Bouafoura, B. Braiek," $PI^{\lambda} D^{\mu}$ controller design for integer and fractional plants using piecewise orthogonal functions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 5, pp. 1267-1278, 2010.

[29] F. Padula, A.Visioli, "Tuning rules for optimal PID and FPID controllers", *Journal of Process Control*, vol. 21, pp. 69-8, 2011.

[30] D. Maiti, S. Biswas, A. Konar, "Design of a fractional order PID controller using particle swarm optimization technique," *2nd National Conference in Recent Trends in Information Systems*, pp.457-462, 2008.

[31] M. Aghababa, "Optimal design of fractional –order PID controller for five bar linkage robot using a new particle swarm optimization algorithm," *Soft Computing*, vol. 20, no. 10, pp. 4055-4067, 2016.

[32] A. Biswas, S. Das, A. Abraham, S. Dasguph, "Design of fractional-order PID controllers with an improved differential evolution," *Engineering Applications of Artifical Intelligence*, vol.22, pp. 343-350, 2009.

[33] M. Itik, E. Sahin, M. Ayas, "Fractional order control of conducting polymer artificial muscle," *Expert systems with applications*, vol. 42, pp. 8212-8220, 2015.

[34] S. Das, I. Pan, A. Gupta, "Improved model reduction and tuning of fractional-order $Pi^{\lambda} d^{\mu}$ controllers for analytical rule extraction with genetic programming," *ISA Transactions*, vol. 51, no. 2, pp. 237-261, 2012.

[35] A. Kesarkar, N. Selvaganesan, "Tuning of fractional-order PID controller using artificial bee colony algorithm," *Systems science and control engineering*, vol. 3, no. 1, pp. 99-105, 2015.

[36] A. Tes, C. Yeroglu, "Optimal fractional order PID design via tabu serach based algorithm," *ISA Transactions*, vol. 60, pp. 109-118, 2016.

[37] A. Mehrabian, C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization," *Ecological Informatics*, vol. 1, no. 4, pp. 355- 366, 2006.

[38] M. Khalilpour, N. Razmjooy, H. Hosseini, P. Moallem, "Optimal control of DC motor using invasive weed optimization (IWO) algorithm," *Majlesi Conference on Electrical Engineering*, Majlesi New Town, Isfahan, Iran, pp. 1-6, 2011.

[39] A. Ahmed, B. Ruhul, "Performance comparison of invasive weed optimization and particle swarm optimization algorithm for the tuning of power system stabilizer in multi-machine power system," *International Journal of Computer Applications*, vol. 41, no. 16, pp. 29-36, 2012.