British Journal of Applied Science & Technology
14(1): 1-14, 2016, Article no.BJAST.23146
ISSN: 2231-0843, NLM ID: 101664541

SCIENCEDOMAIN *international*
www.sciencedomain.org

Impact of Performance Evaluation Criteria on Intelligent Tuning

Mohammed Majid M. Al Khalidy[1*] and Luisella Balbis[2]

[1]*Department of Electrical and Electronics, University of Bahrain, Bahrain.*
[2]*Department of Computer Engineering, University of Bahrain, Bahrain.*

Authors' contributions

This work is carried out in collaboration between the authors. Author MMMAK proposed, designed, simulated, investigated, analyzed and verified the results. Author LB contributed in design and verified the results. Both authors wrote the first draft, read and approved the final manuscript.

Article Information

DOI: 10.9734/BJAST/2016/23146
Editor(s):
(1) Wei Wu, Applied Mathematics Department, Dalian University of Technology, China.
Reviewers:
(1) Hossam Gaber, University of Ontario Institute of Technology, Canada.
(2) Mukhtar Fatihu Hamza, University of Malaya, Malaysia.
(3) Ette Harrison Etuk, Rivers State University of Science and Technology, Nigeria.
Complete Peer review History: http://sciencedomain.org/review-history/12847

Original Research Article

Received 16th November 2015
Accepted 19th December 2015
Published 31st December 2015

ABSTRACT

This paper addresses the problem of automatically tuning in an intelligent manner so that a balance between efficiency and computational speed is reached. In this paper a new proposed technique which uses Particle Swarm Optimization (PSO) to compute the best optimal value for the PID parameters are presented. In this study two different performance criteria are used simultaneously for the optimization problem, namely Integral of Time-weighted Absolute Error (ITAE) and an output response based performance criteria (Fitness Function). The integration between the two performance criteria produces two distinct tuning techniques called Error-Fitness PSO (EFPSO) and Fitness-Error PSO (FEPSO). This paper also proposes new modified Time Varying Acceleration Coefficients (TVAC) that is used in the PSO algorithm. Finally, simulation experiment on a single degree of freedom robotic arm shows that the proposed techniques can produce optimal PID gains with good computational efficiency and improved step response characteristics. The proposed integration techniques can highly improve the PID tuning optimization in comparison with the one that use only one technique.

**Corresponding author: E-mail: mmajid@uob.edu.bh;*

Keywords: Particle swarms optimization; performance criteria; robotic arm; tuning of PID controllers.

1. INTRODUCTION

Proportional Integral Derivative (PID) controllers have been dominating the world of industrial controllers in the past decades. Their success lies in the simplicity of the controller structure, their robustness and flexibility [1].

The performance, safety and profitability of a PID-controlled system largely depend on the values of the controller's parameters, namely the proportional, integral and differential gains. Manual tuning of PID controllers requires experienced operators able to balance the effects of the three parameters. In order to avoid this time-consuming and labor intensive task, numerous tuning procedures and automatic tuning feature were developed in the past, some of which are still successfully applied in manufacturing and process industry [2].

In recent years, optimization techniques and intelligent control theory has led to the development of meta-heuristic methods for PID parameter tuning. Genetic algorithm, fuzzy logic, neural network, ant colony, Flower Pollination Algorithm, simulated annealing and Particle Swarm Optimization (PSO) all belong to this type of techniques [3-9]. Rajendra and Pratihar [10], Bhaskaran, et al. [11], and Hamza, et al. [12], presented a design for an optimized Interval which realized by GA and PSO.

Gaing [13], presented a new time domain performance criteria that uses the output response characteristics in the time domain. Zamani et al. [14], proposed a general performance criteria over both the time and frequency domain specifications.

Mojallali, et al. [15], Soni and Bhatt [16], Djoewahir, et al. [17], and Sowjanya and Srinivas [18], achieved quantification of system performance through performance evaluation index such as IAE, ISE, ITAE, and ITSE.

This paper proposes new methodology on intelligent tuning depends on the integration of two types of tuning techniques ITAE and Fitness Function W(K). The optimization search concentrates on finding the optimal PID controller's gains. The fitness function based on time domain as well as ITAE. Moreover, the new methodology has the ability to keep a balance between swarm diversity and convergence speed.

2. PARTICLE SWARM OPTIMIZATION (PSO)

Since its first formulation in 1995 [19], PSO has received much attention within the research community that has led to numerous variations of the original algorithm [20-24] as well as applications in various disciplines [25-31]. The canonical PSO can be formulated as an unconstrained N-dimensional minimization problem as shown in Eq. (1):

$$\text{Min } f(\mathbf{X}) \quad \mathbf{X} = [x_1, x_2, \dots x_N] \tag{1}$$

Where, *X* is a *N*-dimensional vector which represent a candidate solution, or particle, in the search space.

At each iteration of the algorithm, the ith particle is characterized by the current position in the search space $X_i = [X_i^1, X_i^2, \dots, X_i^N]$ and the current velocity $V_i = [V_i^1, V_i^2, \dots, V_i^N]$

In addition, each particle remembers the best position it has ever reached in the search space. This best position of the ith particle is called the personal best, or $pbest_i = [pbest_i^1, pbest_i^2, \dots, pbest_i^N]$. The fitness value associated with the pbest position is also recorded. Finally, the PSO algorithm records the best fitness value achieved among all particles in the swarm, called the global best fitness, and the candidate solution that achieved this fitness, called the global best position, or $gbest = [gbest^1, gbest^2, \dots, gbest^N]$.

The PSO algorithm is fairly simple. Each iteration k consists of three steps [32]:

1. Evaluate the fitness of each particle.
2. Update individual and global best fitnesses and positions.
3. Update velocity and position of each particle.

The fitness of each particle is assessed by supplying the candidate solution to the objective function (1), also called fitness function. The newly evaluated fitnesses are compared with the previously recorded individual and global best fitnesses, which are replaced if necessary.

The last step of the algorithm updates the velocity and positions of the j[th] dimension of the i[th] particle using the following formulas:

$$V_i^j(k+1) = wV_i^j(k) + c_1 r_1 (pbest_i^j(k) - X_i^j(k)) + c_2 r_2 (gbest^j\,(k) - X_i^j(k)) \quad (2)$$

$$X_i^j(k+1) = X_i^j(k) + V_i^j(k+1) \quad (3)$$

The parameters r_1 and r_2 are uniformly distributed random numbers between 0 and 1 which introduce a stochastic component in the search process.

In recent years improvements to the convergence and diversity of PSO have been made through dynamic adaptation of these parameters using different methods [33-35].

The algorithm keeps iterating till a determined stopping criterion is met, e.g. a fixed number of iteration, a specified error bound or a predefined target fitness value is reached.

In equation (2) different parameters appear which play a different role in the PSO algorithm. The first parameter w is called inertia weight. Its value typically ranges between 0.8 and 1.2, which can either reduce the particle's inertia or accelerate the particle in its original direction. Usually larger values of w support exploration of new search areas whereas smaller values of w facilitate fine-searching the current search area.

The parameters c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest and they are called cognitive and social coefficient respectively. The value of the cognitive coefficient c_1 is usually between 0 and 2 and determines the extent to which a particle is attracted towards its own personal best solution. Similarly, the social coefficient c_2 ranges between 0 and 2 and determines the extent to which a particle is attracted towards the global best solution found so far [28]. In [36] it has been demonstrated that convergence of the algorithm is satisfied under the following conditions:

$$\frac{c_1+c_2}{2} - 1 < w < 1 \quad (4)$$

3. PID TUNING BASED ON PSO

Generally, textbooks describe the control signal *u(t)* generated by the PID controller with the following equation [37]:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (5)$$

in which e is the difference between the between the measured process variable y and the reference or set point y_{sp}. The control signal is therefore a sum of three terms: the proportional term directly related to the error e through the constant K_p, the integral term proportional to the integral of the error e through the constant K_i and the derivative term proportional to the derivative of the error e through the constant K_d.

As mentioned before, the performances of the PID controller largely depends on the values of the parameters K_p, K_i, K_d. PSO-based tuning of these parameters is generally centered on error criteria, as shown in Fig. 1.

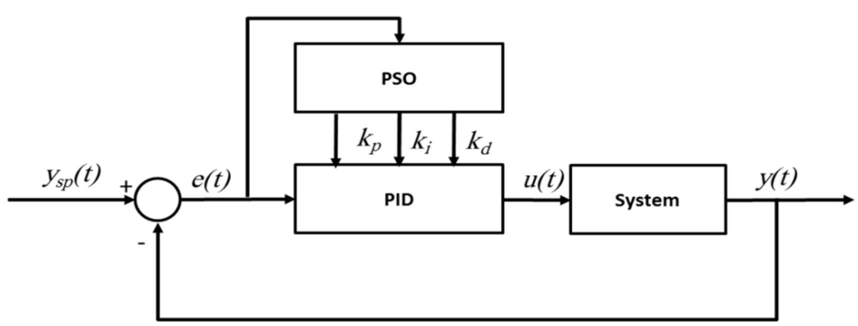

Fig. 1. PSO-PID control loop

In other words, the performance criterion considered in the PSO optimization problem is a standard PID integral performance criterion such as the Integral Absolute Error (IAE), or the integral of time-weighted absolute error (ITAE), or the integral of squared-error (ISE), or the Integral of Time-Weighted-Squared Error (ITSE) [18,15]. These criteria are expressed by the following equations:

$$IAE = \int_0^T |e|\, dt \tag{6}$$

$$ITAE = \int_0^T t|e|\, dt \tag{7}$$

$$ISE = \int_0^T e^2\, dt \tag{8}$$

$$ITSE = \int_0^T te^2\, dt \tag{9}$$

In [13] a new performance criterion is presented. This method uses the output response characteristics in the time domain, such as overshoot M_p, rise time t_r, settling time t_s, and steady-state error e_{ss} to evaluate the choice of K_p, K_i, K_d. The performance criterion *W(K)* is defined as:

$$min_{K\ stabilizing} W(K) = (1 - e^{-\beta})(M_p + e_{ss}) + e^{-\beta}(t_s - t_r) \tag{10}$$

Where, K=[K_p K_i K_d], and β is a weighting factor set in the range between 0.8 and 1.5.

4. PROPOSED PSO-PID CONTROLLER

In this work, two different tuning techniques are proposed, the Error-Fitness PSO (EFPSO) and Fitness-Error PSO (FEPSO) which use a combination of two performance criteria to search and find the optimal PID controller's gains. The two performance criteria used are the ITAE and the output response based performance criteria *W(K)* introduced above. The integration of these two performance criteria has been empirically investigated, analyzed and verified.

4.1 EFPSO

The EFPSO technique implicitly depends on the combination of two equations; the fitness function *W(K)*, that can satisfy the set design requirements depending on the set of numbers for the adaptation based on the expected parameters, and ITAE, as shown in the following equations:

$$Error = Input\ Signal - min_{K\ stabilizing} W(K) \tag{11}$$

$$ITAE = \int_0^T t|Error|\, dt \tag{12}$$

The complex derivation process and the time consuming of ITAE through each population leads to use the fitness function *W(K)* only after the updating of velocities and positions as illustrated in the flowchart of Fig. 2.

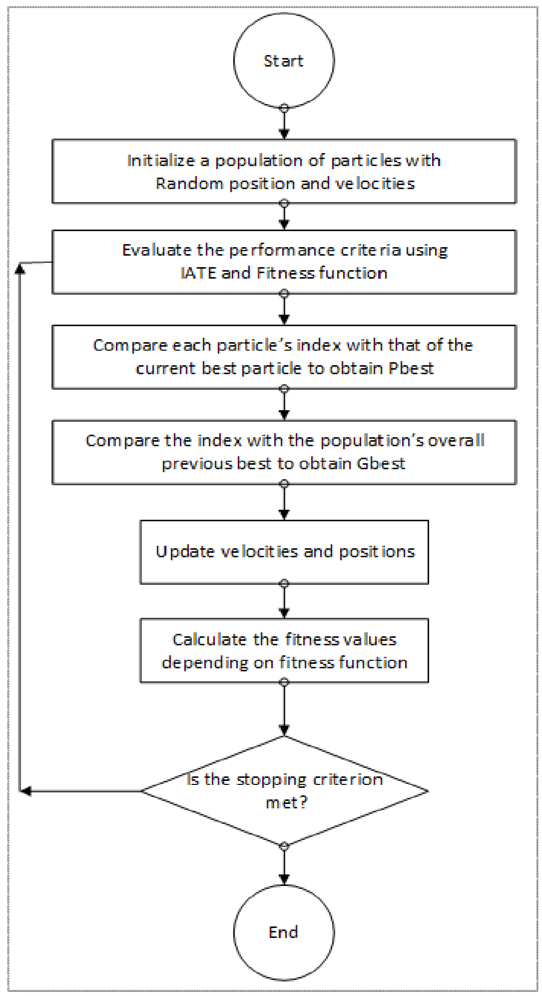

Fig. 2. PSO Using Error-Fitness Function technique (EFPSO)

4.2 FEPSO

FEPSO technique operates in a similar manner as the EFPSO. The latter differs from the first in the sequence of steps as illustrated in the flowchart of Fig. 3. The superiority of the FEPSO techniques with respect to the EFPSO is demonstrated by the better results obtained using FEPSO, although the computed optimal PID gains were greater than the gains that obtained from EFPSO.

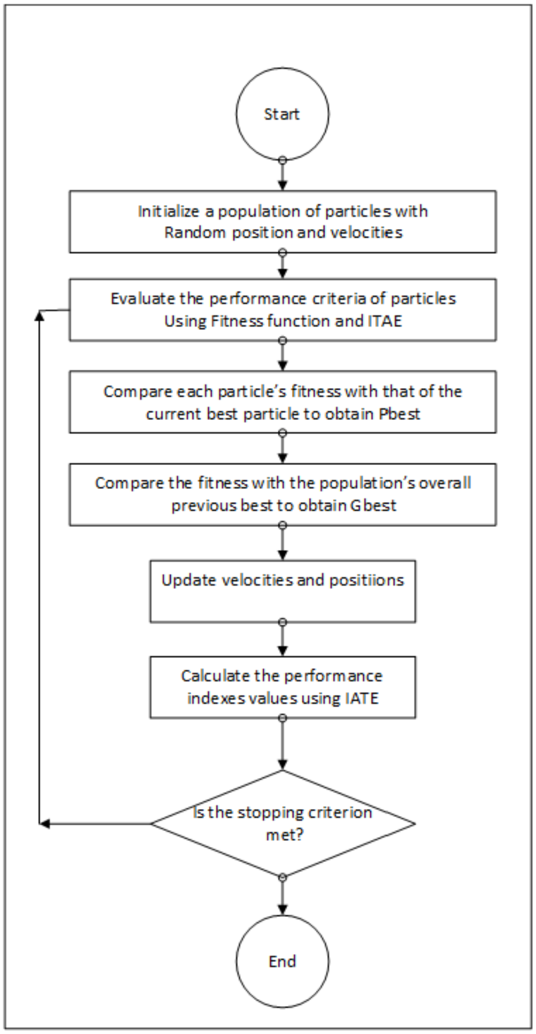

Fig. 3. PSO Using Fitness – Error technique (FEPSO)

5. SIMULATION RESULTS

5.1 Robotic ARM System

The proposed tuning techniques for the PID controller have been tested on a single degree of freedom robotic arm. We assumed the system as a two links robot connected by single joint driven by a DC motor as shown in Fig. 4.

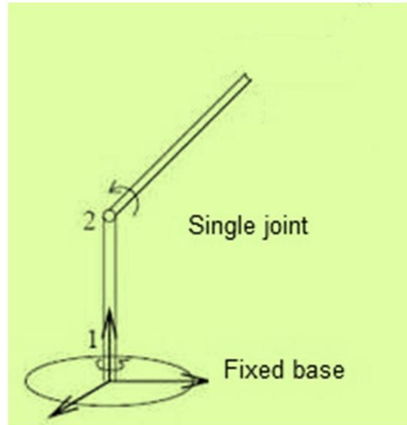

Fig. 4. Single DOF robotic arm

The DC motor directly provides a rotatory motion proportional to the applied input voltage. The electrical model of such a DC motor is shown in Fig. 5.

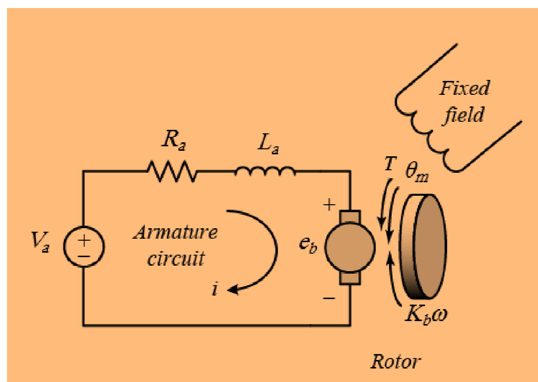

Fig. 5. Electrical model of DC motor

Where, V_a is the armature voltage, R_a is the armature resistance, L_a is the armature inductance, T is the motor torque, e_b is the back *emf* and θ_m is the angular displacement of the motor shaft.

In this work, we assumed that the input of the system is the voltage source V_a applied to the motor's armature, while the output is the angular position of the shaft θ_m

The control problem consists of controlling the angular positioning of the link by regulating the armature voltage of the motor.

The dynamic of the single joint can be approximated using the following transfer function [38]:

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_a}{[s(s^2 J_{eff} L_a + (L_a f_{eff})s + R_a f_{eff} + K_a K_b)]} \quad (13)$$

Since the electrical time constant is much bigger than the electrical time constant of the motor then the armature inductance La can be neglected.

Hence,

$$\frac{\theta_m(s)}{V_a(s)} = \frac{K_a}{[s(sR_a J_{eff} + R_a f_{eff} + K_a K_b)]} \quad (14)$$

Where, the following variables for the single joint are:

J_m = Moment of inertia of the motor referred to the motor shaft.
J_L = Moment of inertia of the motor referred to the load shaft.
$J_{eff} = J_m + n^2 J_L = 1$ (oz.in.s^2/rad).
L_a = 0 Henry: Armature inductance.
R_a=1.3 Ohm: Armature resistance.
n =1: Gear reduction ratio.
f_m = Viscous friction coefficient related to the motor shaft.
f_L = Viscous friction coefficient related to the load shaft.
$f_{eff} = f_m + n^2 f_L$ = 0.11 N-m.s/rad
K_a = 1 N-m/A: Motor torque proportional constant.
K_b = 0.33 V-s/rad: Proportionality constant between ω (angular speed) and e_b (back emf).

If there is a gear train between the motor and load, then the angular displacement of the load θ_L is different from the angular displacement of the motor θ_m. The angles are related by the gear ratio relationship, described by the following equation:

$$\frac{\theta_L}{\theta_m} = n \quad (15)$$

Therefore, equation (14) becomes

$$\frac{\theta_L(s)}{V_a(s)} = \frac{nK_a}{[s(sR_a J_{eff} + R_a f_{eff} + K_a K_b)]} \quad (16)$$

The block diagram for the position control of the robotic arm is shown in Fig. 6.

The robotic arm system was first controlled by a PID tuned using the classical Ziegler-Nichols method [39,40]. The computed PID gains K_p, K_i and K_d are illustrated in Table 1.

Table 1. PID gains based on Ziegler Nichols technique

Gain	Value
K_p	0.666
K_i	0.00877
K_d	0.1266

With these parameters, the closed loop transfer function of the control system becomes:

$$\frac{\theta_L(S)}{\theta_d(S)} = \frac{0.1948S^2 + 1.025S + 0.013\ 49}{S^3 + 0.9225S^2 + 1.025S + 0.013\ 49} \quad (17)$$

The step response of the control system tuned using the Ziegler-Nichols method is shown in Fig. 7.

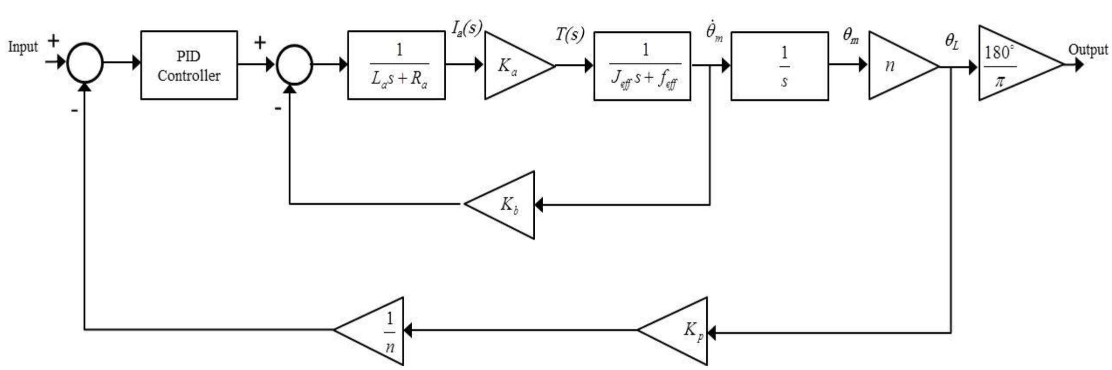

Fig. 6. Position and velocity block diagram for DC motor robot arm

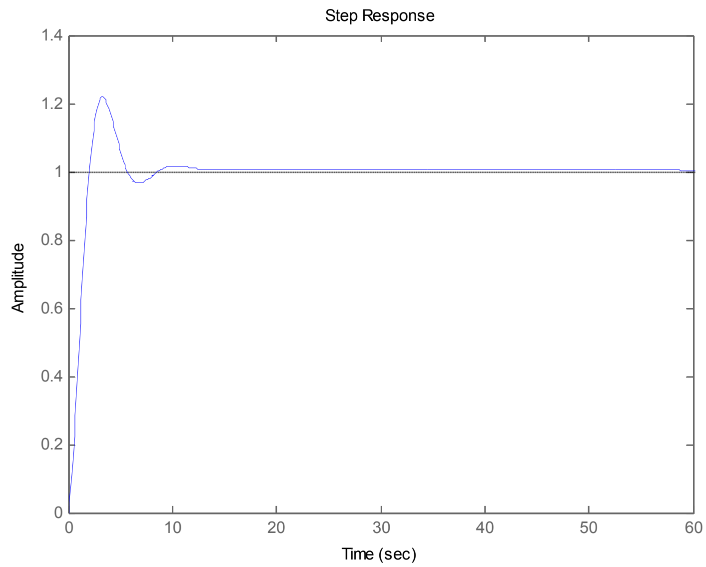

Fig. 7. Step response for closed loop system

5.2 Experimental Results

The performance of the proposed techniques has been verified through simulation. The optimal PID gains have been obtained using EFPSO and FEPSO techniques. The output of the system was carried out through step response to verify the stability response, speed and steady state error for the system.

A nonlinearly decreasing inertia weight is used to investigate the influence of path variant of decreased inertia weight. Nonlinearity degree of the path adjusted by nonlinear index number x. This number can significantly affect the performance of PSO. The nonlinearly decreasing inertia weight is [32]:

$$w = w_{min} + (w_{max} - w_{min}) \cdot \left(\frac{k_{max}-k}{k_{max}-1}\right)^x \quad (18)$$

Where, x is the nonlinear index number. Normally, x is set to 1.5 for the best performance. k_{max} is the number of maximum iteration and k is the current number of iterations.

The parameter x was set to 1.5 for the best performance, whereas the inertia weight parameters w_{max} and w_{min} were set to 0.9 and 0.4 respectively, which provided a balance between global and local explorations.

Time Varying Acceleration Coefficients (TVAC) were used to avoid premature convergence in initial iterations of the search and to improve convergence to the global optimum solution during the final stages of the search.

In this work we proposed and modified the TVAC to be as given by Eq. (19) and (20):

$$c_1 = (c_{1s} - c_{1L})\left(\frac{i}{I_{max}}\right) + c_{1L} \quad (19)$$

$$c_2 = (c_{2s} - c_{2L})\left(\frac{i}{I_{max}}\right) + c_{2L} \quad (20)$$

Where c_{1s} and c_{2s} are the initial values of the acceleration coefficients c_1 and c_2. c_{1L} and c_{2L} are the final values of the acceleration coefficients c_1 and c_2. The suggested value of c_1 from 2.5 to 0.5 and c_2 from 0.5 to 2.5 results in good performance of PSO [41].

5.2.1 Simulation for EFPSO technique

To verify the effectiveness of the proposed EFPSO technique simulations were carried out with the control parameters tuned according to the trials and swarm populations shown in Table 2.

More than one trial were randomly generated for each swarm population. It was noticed that the best output responses was obtained in correspondence to the maximum performance indexes. This means that the boundary of each gain for the PID controller can be obtained and its effectiveness on the output response can be observed. ITAE and *W(K)* values had a convergence in shape and divergence in position and magnitude and that gives a remarkable ability for more controllable tuning parameters between these two techniques. In other words, the particles position can be measured and adjusted prematurely or currently regarding to the optimization method (comparing algorithm in Fig. 2) which related to the designer, as illustrated in Figs. 8, 9, 10 and 11, where the step response for maximum performance indexes also is shown to verify the performance of the EFPSO technique. For Swarm population of 30 it was difficult to get a stable system as the boundary of gains was very narrow.

Table 2. Simulation result for EFPSO

Trial	K_d	K_p	K_i	Performance indexes	Best fitness	Swarm population
1	22.8614	36.1007	3.9029	5.5727e-008	1.0020	100
2	6.9747	5.4601	1.4334	5.2687e-008	0.9243	100
3	20.5825	28.7669	4.0385	**5.7324e-008**	1.0001	100
4	16.0205	20.3322	7.0730	**6.3158e-008**	0.8708	70
5	9.9372	9.0213	3.6974	3.6082e-010	0.9924	70
6	14.9605	15.7296	9.5327	3.7073e-008	0.9426	70
7	17.1273	15.8685	14.1294	**6.3806e-008**	0.9901	50
8	9.6634	8.1553	3.0706	2.9757e-008	0.9592	50
9	5.4900	5.2084	0.0607	1.6806e-007	0.7042	50
10	8.2752	2.0875	0.5690	9.9217e-008	0.8077	30
11	Difficult to get a stable system					30
12	Difficult to get a stable system					30

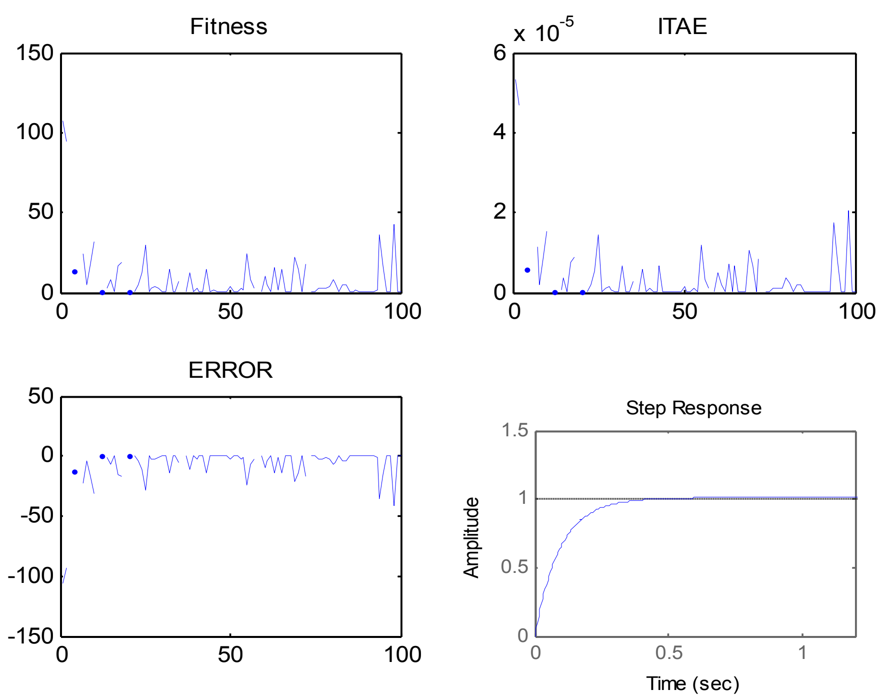

Fig. 8. Fitness, ITAE, error and step response of robot arm using EFPSO for 100SP

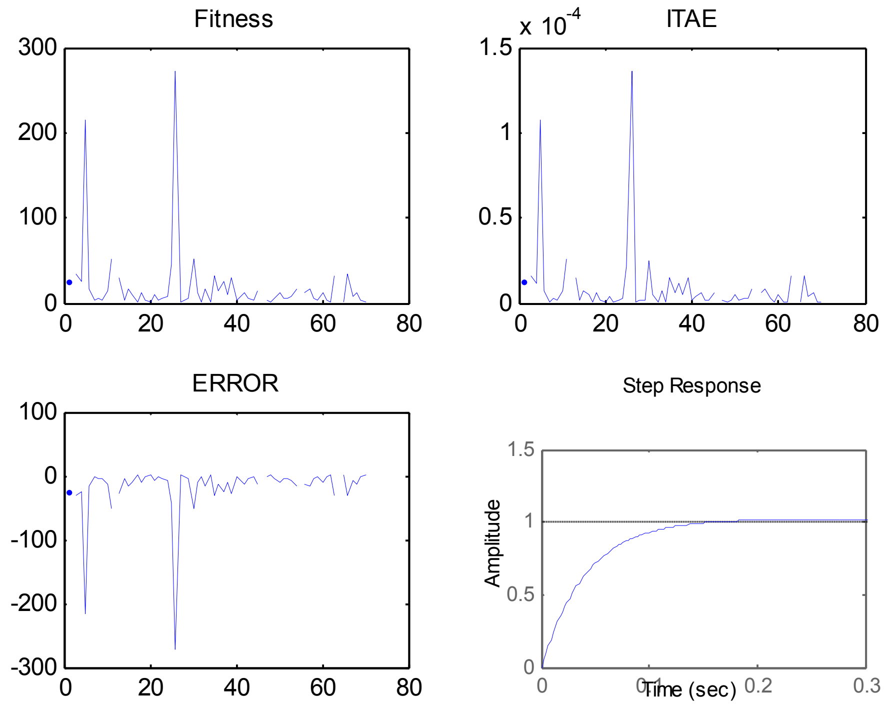

Fig. 9. Fitness, ITAE, Error and step response of robot arm using EFPSO for 70SP

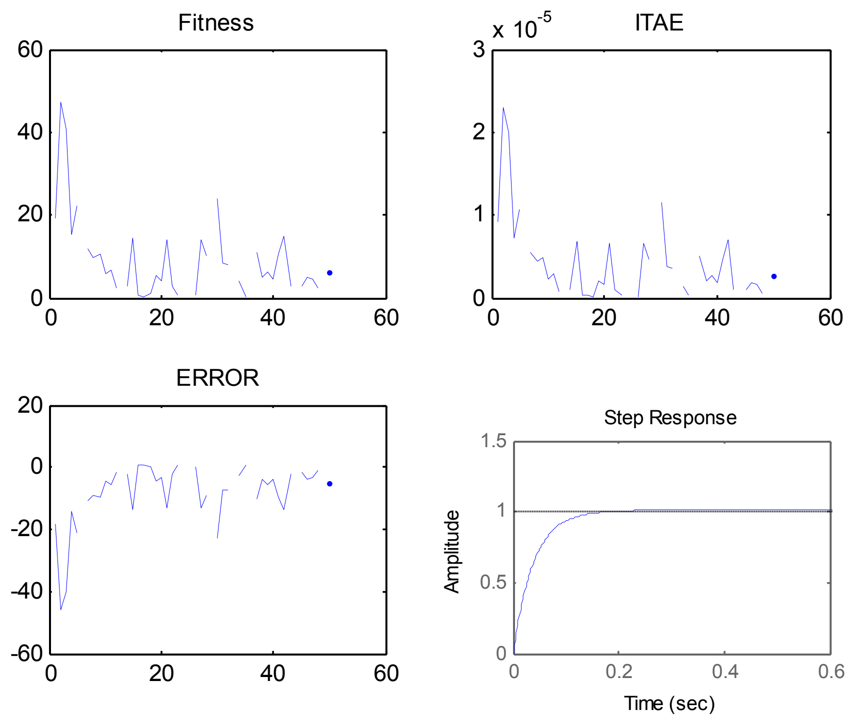

Fig. 10. Fitness, ITAE, error and step response of robot arm using EFPSO for 50SP

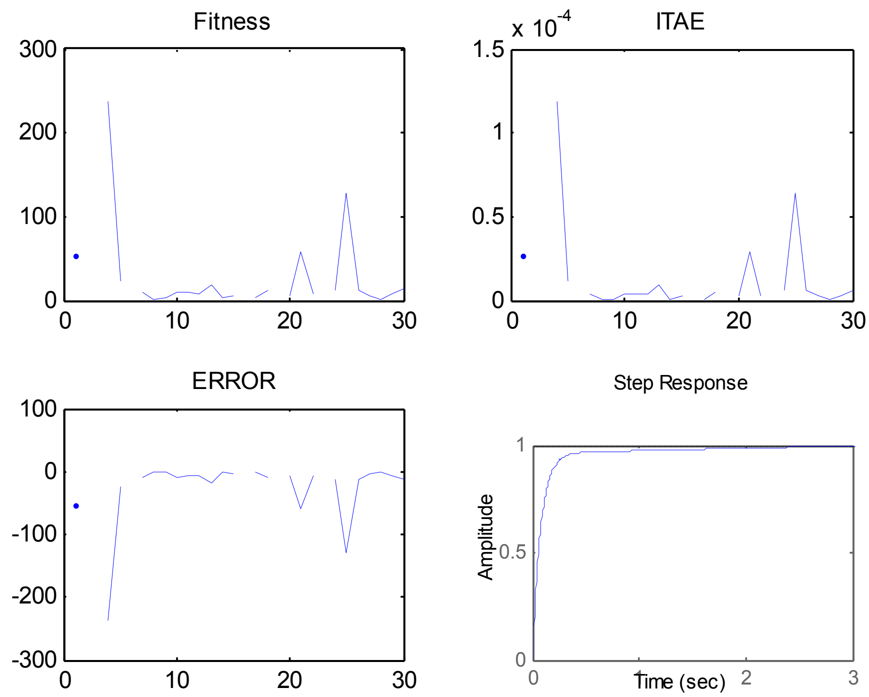

Fig. 11. Fitness, ITAE, error and step response of robot arm using EFPSO for 30SP

5.2.2 Simulation for FEPSO technique

In This section, the simulation shows the stability boundary for various trials for difference swarm populations for the PID controller gains, as illustrated in Table 3.

The results indicate that the obtained PID gains were higher compared with that one obtained through EFPSO. The results also show that the best output response was obtained in correspondence of minimum fitness value for the same swarm population. The proposed tuning FEPSO technique can produce optimal PID gains that affect the system to get better output response compared with the classical PSO, as illustrated in Figs. 12, 13, 14 and 15.

Table 3. Simulation result for FEPSO

Trial	K_d	K_p	K_i	Best fitness	Performance indexes	Swarm population
1	287.0102	27.8888	13.2988	**0.1972**	5.1764e-008	100
2	144.5535	17.0928	14.0902	0.1982	1.3534e-008	100
3	70.4271	23.7048	31.1467	0.1973	7.7777e-008	100
4	21.8193	12.8355	2.9018	**0.1976**	1.5233e-007	70
5	651.0239	12.6835	6.6429	0.1978	4.0027e-008	70
6	15.7645	10.1152	4.3912	0.2001	6.6046e-009	70
7	40.9398	16.8193	12.3807	**0.2010**	1.4577e-008	50
8	16.3631	9.7694	0.3701	0.2016	7.0698e-008	50
9	10.4255	6.2375	0.8608	0.2064	1.0175e-008	50
10	5.1896	2.8900	0.5757	**0.2345**	1.8025e-007	30
11	7.3202	5.8264	0.0703	0.2417	1.3463e-007	30
12	4.6902	3.1122	1.5118	0.2367	1.9786e-006	30

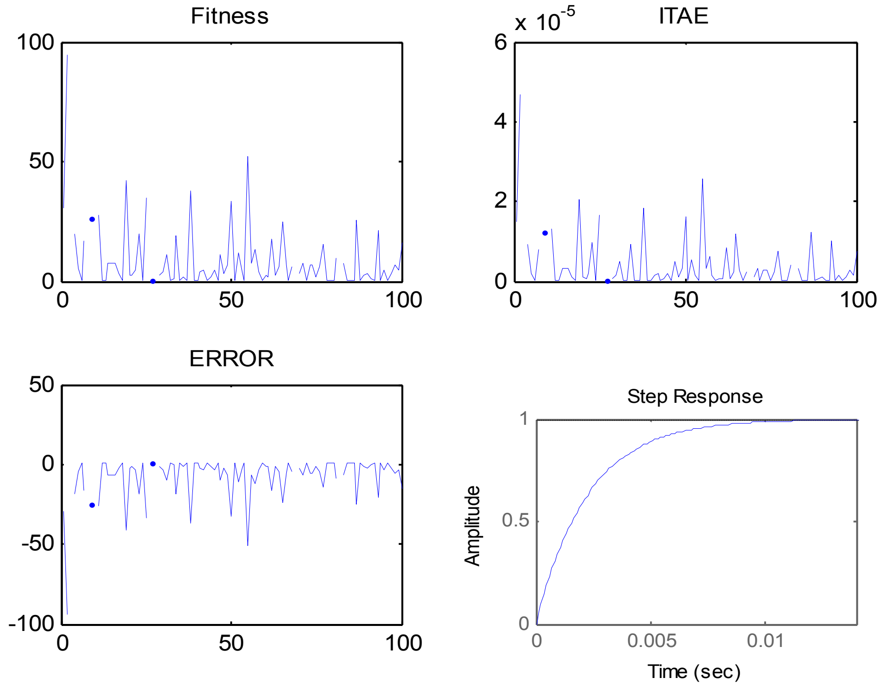

Fig. 12. Fitness, ITAE, error and step response of robot arm using FEPSO for 100SP

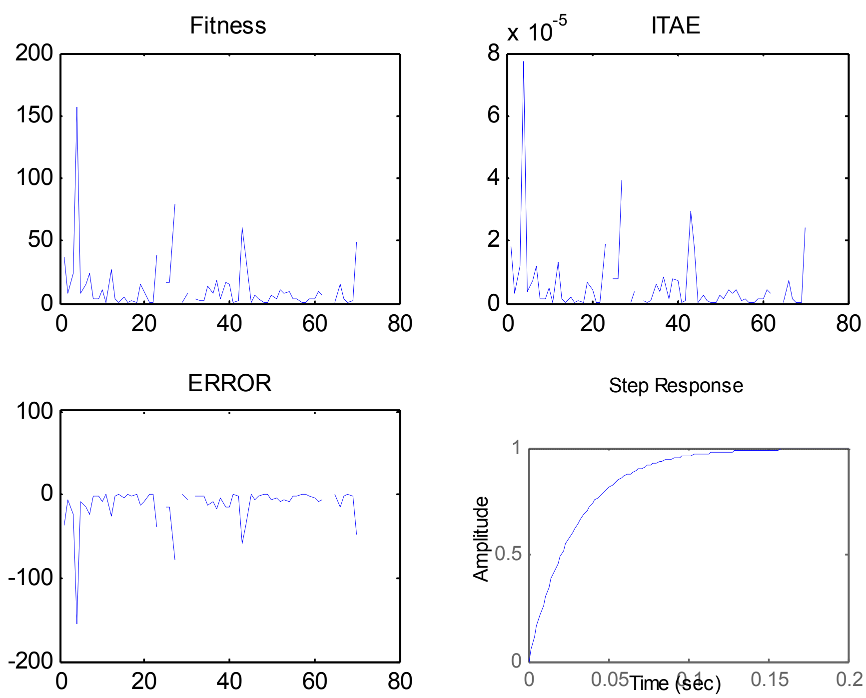

Fig. 13. Fitness, ITAE, error and step response of robot arm using FEPSO for 70SP

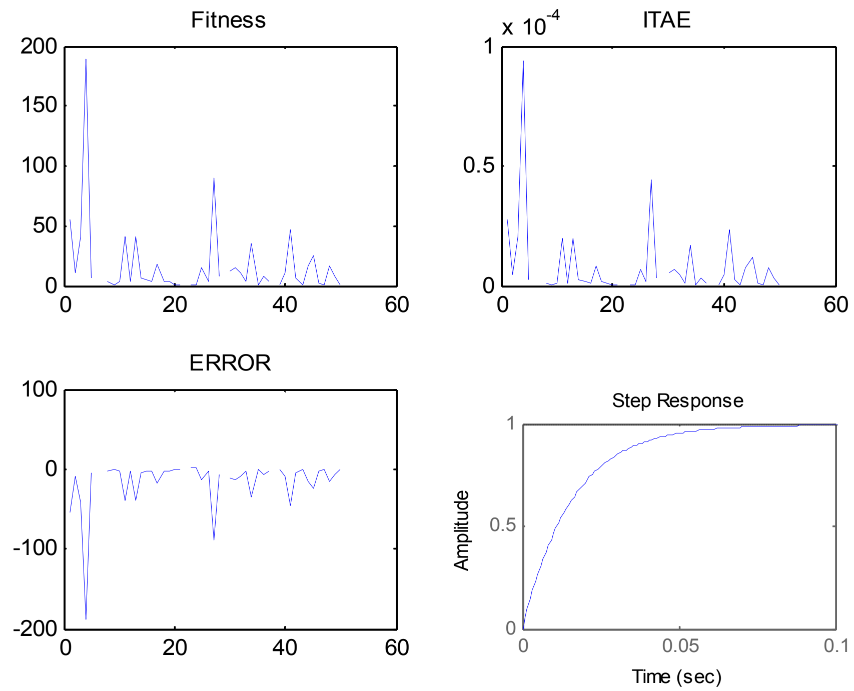

Fig. 14. Fitness, ITAE, error and step response of robot arm using FEPSO for 50SP

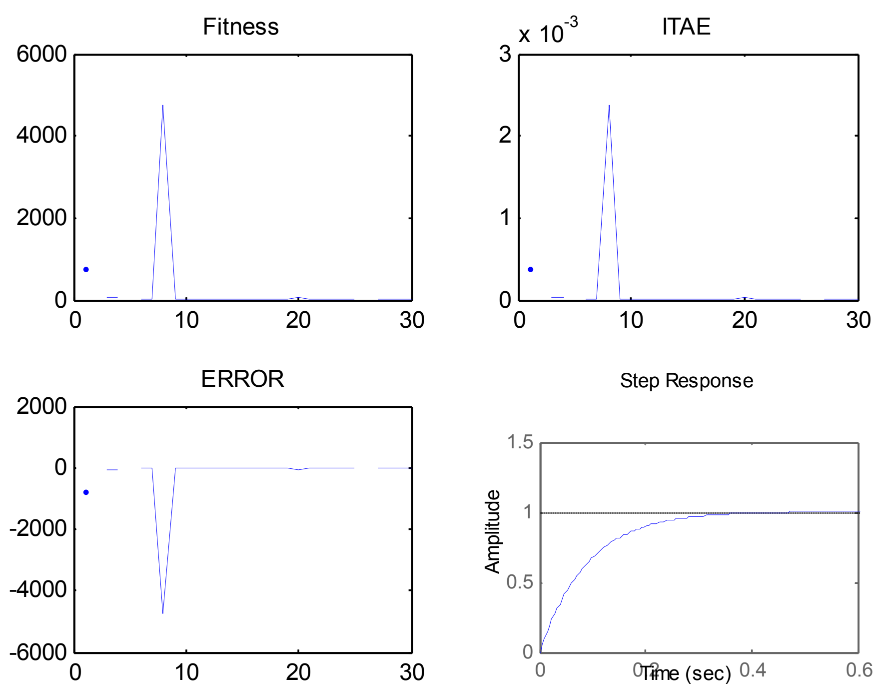

Fig. 15. Fitness, ITAE, error and step response of robot arm using FEPSO for 30SP

6. CONCLUSIONS

In this work, two techniques were proposed for optimal tuning of PID gains. The techniques rely of the use of PSO and a combination of two performance criteria; ITAE and Fitness Function *W(K)*. The two proposed techniques were empirically investigated, analyzed and verified.

The sophistication of the proposed techniques coming from keeping a balance between swarm diversity and convergence speed in addition, the simulation results showed that both EFPSO and FEPSO outperform classical PSO, with FEPSO technique performing better than the EFPSO. Finally, the new proposed time varying acceleration coefficients (TVAC) shows a good ability for particles to determine the extent to which a particle is attracted towards its own personal best solution, and towards the global best solution.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Astrom KJ, Hagglund T. PID controller: Theory, design and tuning. Instrument Society of America, Research Triangle Park.
2. O'Dwyer A. Handbook of PI and PID controller tuning rules. London, U.K.: Imperial College Press, 2nd edition; 2006.
3. Tang KS, Kim Fung Man, Chen G, Kwong S. An optimal fuzzy PID controller. IEEE Trans. on Industrial Electronics. 2001; 48(4):757-765.
4. Chengzhi C, Xiaofeng G, Yang L. Research on ant colony neural network PID controller and application, in Proc. 8th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. 2007;253-258.
5. Boubertakh H, Tadjine M, Glorennec PY, Labiod S. Tuning fuzzy PID controllers using ant colony optimization. in Proc. 17th Mediterranean Conf. on Control & Automation, Thessaloniki; 2009.
6. Yang Zhi, Chen Zhitang, Fan Zhengping. Tuning PID controller based on improved particle swarm optimization algorithm. Control Theory and Applications. 2010; 27(10):1345-1352.
7. Zhang Jiajun. PID controller parameters based on particle swarm optimization. Computer Simulation. 2010;27:191-193.
8. Hamza MF, Yap HJ, Choudhury IA. Recent advances on the use of meta-heuristic optimization algorithms to optimize the type-2 fuzzy logic systems in intelligent control. Neural Computing and Applications. 2015;1-21.
9. Chiroma H, et al. A review of the applications of bio-inspired flower pollination algorithm. Procedia Computer Science. 2015;62:435-441.
10. Rajendra R, Pratihar DK. Particle swarm optimization algorithm vs genetic algorithm to develop integrated scheme for obtaining optimal mechanical structure and adaptive controller of a robot. Intelligent Control and Automation. 2011;2:430-449.
11. Bhaskaran S, Varma R, Ghosh J. A comparative study of GA, PSO and APSO: Feed point optimization of a patch antenna. International Journal of Scientific and Research Publications. 2013;3(5). ISSN 2250-3153.
12. Hamza MF, Yap HJ, Choudhury IA. Genetic algorithm and particle swarm optimization based cascade interval type 2 fuzzy PD controller for rotary inverted pendulum system. Mathematical Problems in Engineering, Hindawi Publishing Corporation; 2015. Article ID 695965.
13. Gaing ZL. A particle swarm optimization approach for optimum design of PID controller in AVR system. IEEE Tras. On Energy Conversion. 2004;19(2).
14. Zamani M, Ghartemani MK, Sadati N, Parniani M. Design of a fractional order PID controller for an AVR using particle swarm optimization. Elsevier, Control Engineering Practice. 2009;17(12):1380–1387.
15. Mojallali H, Gholipour R, Khosravi A, Babaee H. Application of chaotic particle swarm optimization to PID parameter tuning in ball and hoop system. Int. Journal of Computer and Electrical Engineering. 2012;4(4).
16. Soni YK, Bhatt R. BF-PSO optimized PID controller design using ISE, IAE, IATE and MSE error criteria. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). ISSN: 2278 – 1323, 2013;2(7).
17. Djoewahir A, Tanaka K, Mu S. Adaptive PSO-based self-tuning PID controller for ultrasonic motor. Int. Journal of Innovation, Management and Technology. 2013;9(10): 3903-3914.
18. Sowjanya KL, Srinivas LR. Tuning of PID controllers using particle swarm optimization. Int. Journal of Industrial Electronics and Electrical Engineering. 2015;3(2):2347-6982.
19. Kennedy J, Eberhart RC. Particle swarm optimization. In: Proc. of IEEE Intentional

Conference on Neural Networks, Perth. 1995;1942–1948.
20. Jiang Y, Hu T, Huang CC, Wu X. An improved particle swarm optimization algorithm. Applied Mathematics and Computation. 2007;193(1):231–239.
21. Tripathi PK, Bandyopadhyay S, Pal SK. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. Information Sciences. 2007; 177(22):5033–5049.
22. Wang YJ, Yang YP. Particle swarm optimization with preference order ranking for multi-objective optimization. Information Sciences. 2009;179(12):1944–1959.
23. van den Bergh F, Engelbrecht AP. A cooperative approach to particle swarm optimization. IEEE Trans. on Evolutionary Computation. 2004;8(3):225–239.
24. Zhao X. A perturbed particle swarm algorithm for numerical optimization. Appl. Soft Computing. 2010;10(1):119–124.
25. Lim A, Lin J, Xiao F. Particle swarm optimization and hill climbing for the bandwidth minimization problem. Appl Intell. 2007;26(2):175–182.
26. Tušar T, Korošec P, Papa G, Filipi˘c B, Šilc J. A comparative study of stochastic optimization methods in electric motor design. Appl Intell. 2007;27(2):101–111.
27. Van den Bergh F, Engelbrecht AP. Training product unit networks using cooperative particle swarm optimizers. In Proc. of IEEE Int. Joint Conference on Neural Networks. 2001;126–132.
28. Khan SA, Engelbrecht AP. A fuzzy particle swarm optimization algorithm for computer communication network topology design. Appl Intell. 2012;36(1):161-177.
29. Al-Jaafreh M, Al-Jumaily M. Particle swarm optimization based stroke volume influence on mean arterial pressure. In Proc. of the IEEE International Conference on Biomedical and Pharmaceutical Engineering. 2006;508–512.
30. Xu C, Zhang Q, Wang B, Zhang R. Improved particle swarm optimization algorithm for 2D protein folding prediction. In Proc. of the IEEE 2nd International Conference on Bioinformatics and Biomedical Engineering. 2008;816–819.
31. Ochoa A, Hernandez A, Gonzalez S, Jons S, Padilla A. Hybrid system to determine the ranking of a returning participant in Eurovision. Proc. of the 8th IEEE International Conference on Hybrid Intelligent Systems. 2008;489–494.
32. van den Bergh F. An analysis of particle swarm optimizers. PhD dissertation, University of Pretoria; 2001.
33. Alfi A. PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. Acta Automatica Sinica. 2010;37(5).
34. Valdez F, Melin P, Castillo O. An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. Applied Soft Computing. 2011;11(2):2625–2632.
35. Melin P, Olivas F, Castillo O, Valdez F, Soria J, Valdez M. Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. Expert Systems with Applications. 2013;40:3196–3206.
36. Van den Bergh F, Engelbrecht AP. A new locally convergent particle swarm optimizer. Proc. of IEEE Conference on Systems, Man, and Cybernetics. 2002;96–101.
37. Dorf RC, Bishop RH. Modern control systems. 9th edition, Prentice – Hall (NJ); 2001.
38. Agrawal R, Kabiraj K, Singh R. Modeling a controller for an articulated robotic arm. Intelligent Control and Automation. 2012;3: 207-210.
39. Bhattacharyya SP, Silva GJ, Dutta A. PID parameters for Time-Delay systems, Birkhauser, Texas; 2004.
40. Xue D, Chen YQ, Atherton DP. Linear feedback control analysis and design with MATLAB, Siam; 2007.
41. Borkar AL. Review of particle swarm optimization techniques. Int. Journal of Innovative Research in Science & Engineering, ISSN (Online). 2010;2347-3207.

Peer-review history:
The peer review history for this paper can be accessed here:
http://sciencedomain.org/review-history/12847