International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization on TPE (IOTPE)

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

December 2011	Issue 9	Volume 3	Number 4	Pages 10-17

OPTIMAL TUNING OF PID CONTROLLER PARAMETERS ON A DC MOTOR BASED ON ADVANCED PARTICLE SWARM OPTIMIZATION ALGORITHM

A. Jalilvand [1] A. Kimiyaghalam [2] A. Ashouri [3] H. Kord [1]

1. Electrical Engineering Department, University of Zanjan, Zanjan, Iran, ajalilvand@znu.ac.ir, hkord63@yahoo.com
2. Islamic Azad University, Zanjan Branch, Young Researchers Club, Zanjan, Iran, a.kimiyaghalam@gmail.com
3. Department of Electrical Engineering, Islamic Azad University, Khodabandeh Branch, Khodabandeh, Iran
a_ashouri2007@yahoo.com

Abstract- Tuning of PID controller parameters is an important problem in control field. To solve this problem we used an Advanced Particle Swarm Optimization which is powerful stochastic evolutionary algorithm that is used to find the global optimum solution in search space. The proposed method has fast searching speed compared to standard PSO. Furthermore this method accelerates the convergence. However, it has been observed that the standard PSO algorithm has premature and local convergence phenomenon when solving complex optimization problem. This new algorithm is proposed to augment the original PSO searching speed. The optimum tuned PID controller is applied to a DC motor. The simulation results show that the PID controller designed by APSO demonstrates better results than GA, PSO and even Improved PSO technique.

Keywords: Improved PSO, APSO, PID Controller, Optimization Problem, DC Motor.

I. INTRODUCTION

Proportional-Integral-Derivative (PID) controller is one of the earliest control technique that is still used widely in industrial because of its easy implementation, robust performance and being simple of physical principle of parameters. For achieving appropriate closed-loop performance, three parameters of the PID controller must be tuned [1, 2].

Tuning methods of PID parameters are classified as traditional and intelligent methods. Conventional methods such as Zigeler and Nichols [3] and simplex method [4] are hard to determine optimal PID parameters and usually are not caused good tuning, i.e., it produces surge and big overshoot.

Recently, intelligent approaches such as genetic algorithm [5-7] and particle swarm optimization [8] have been proposed for PID optimization. Although Genetic Algorithm (GA) has received much interest and has been applied successfully to solve the problem of optimal PID controller parameters [9] but the genetic algorithm may be not efficient for solving some complex optimization problems. This degradation in efficiency is apparent especially in applications when the parameters being optimized are highly correlated [10].

PSO is a novel population-based metaheuristic, which utilize the swarm intelligence generated by the cooperation and competition between the particles in a swarm and has emerged as a useful tool for engineering optimization [11, 12].

Unlike the other heuristic techniques, it has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. Also, it suffices to specify the fitness function and to place finite bounds on the optimized parameters. Compared with the GA, PSO is characterized as a simple concept, easy implementation and good computational efficiency [13].

However, the standard PSO algorithm has also some disadvantages like premature convergence phenomenon similar to the GA. Although some improved methods, such as augment the swarm scale and dynamic adjustment inertia weight factors, can improve the optimization performance to some extent but their convergence speed is slow.

In this study Advanced Particle Swarm Optimization (APSO) algorithm is used to tune the PID controller parameters. Using this algorithm increases the searching speed. This technique puts the adaptively changing terms in original constant terms, so that parameters of the original PSO algorithm changes with the convergence rate which is presented by the fitness function.

As a result, the searching speed of this advanced method is much faster than that of the original method. Through testing the proposed method on a typical DC motor, the experimental results show that the APSO method has more excellent optimization performance than the GA, PSO and even Improved Particle Swarm Optimization (IPSO). APSO is indeed more efficient in improving searching capability and convergence characteristic. The simulation results show that the APSO-based PID controller has excellent performance.

II. MATERIALS AND METHODS

A. Design of PID Controller

One of the most common controlling methods in the market is the PID controller. Application of the PID controller involves choosing the K_P, K_I and K_D that provide satisfactory closed-loop performance. These parameters must be selected so that the characteristics: response speed, settling time and proper overshot rate, all of which guarantee the system stability, would be satisfied. The main method for this purpose is based on trial and error, which is time consuming. There are different processes for different composition of proportional, integral and differential. The duty of control engineering is to adjust the coefficients of gain to attain the error reduction and dynamic responses simultaneously. The transfer function of PID controller is defined as follows:

$$G_{PID}(s) = K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s} \quad (1)$$

PID control is a linear control methodology with a very simple control structure. This type of controller operates directly on the error signal, which is the difference between the desired output and the actual output and generates the actuation signal that drives the plant. In the design of PID controller the amount of K_I is identified to reach to an intended error in steady state. In PID controller design, K_P, K_I and K_D, related to the closed loop feedback system within the least time is determined and requires a long range of trial and error. As shown in Figure 1, PID controllers have three basic terms: proportional action, in which the actuation signal is proportional to the error signal, integral action, where the actuation signal is proportional to the time integral of the error signal and derivative action, where the actuation signal is proportional to time derivative of error signal.

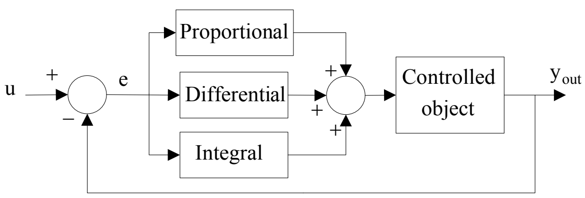

Figure 1. Closed loop PID controlled system

To design a particular control loop, the values of the three parameters (K_P, K_I and K_D) have to be adjusted so that the control input provides acceptable performance from the plant. These three parameters have been included in a chromosome as shown in Figure 2 to be optimized in the optimization procedure. In order to get an acceptable solution, there are several controller design methods that can be applied. For example, classical control methods in the frequency domain or automatic methods like Ziegler-Nichols, known as PID tuning methodology. Although these methods provide a first approximation, the response produced usually needs further manual retuning by the designer before implementation.

K_P	K_I	K_D

Figure 2. Chromosome structure

Fitness Function: The Fitness function is important to be properly defined. In this study, the fitness function is defined as follows:

$$F_{obj} = \left\{ (100 E_{ss}^{0.5} + 5 M_P^2) + (10 t_s + t_r) \right\} \quad (2)$$

where,
t_r is Rise time;
t_s is Settling time;
M_P is Overshoot;
E_{ss} is the steady state error.

The PID controller parameters could be evaluated roughly using conventional tuning method such as Ziegler-Nichols experiential method, to get a smaller search space [3].

B. Genetic Algorithm Overview

GA is a random search method that can be used to solve non-linear system of equations and optimize complex problems. The base of this algorithm is the selection of individuals. It does not need a good initial estimation for sake of problem solution [14, 15]. This algorithm can be used to solve many problems such as optimization of PID parameters.

GA evolves into new generations of individuals by using knowledge from previous generations. The fundamental principle of GA is that chromosomes which include blocks of genetic information that are contained in the optimal solution will increase in frequency if the opportunity of reproduction of each chromosome is related, in some way, to its fitness value. Thus, GA is both explorative and exploitative methods for solving problems that are not affordable by traditional methods.

A typical example occurs when a potential solution of a problem may be represented as a set of parameters, which in their turn are represented by strings of characters. Here, the n dimensions decision-making vector X is denoted with n and X_i marks $X = X_1, X_2, \ldots, X_n$. X_i is named as one gene and X is one chromosome or individual, which consists of n genes. We name this process as coding process. The operation object is the population consisted of M chromosomes. Genetic operations are applied to simulate the evolution mechanism of individuals of initial population.

The individuals with higher fitness values are passed down the next generation. One or more individuals after a series of evolution are the optimal solutions. GA generally includes the three fundamental genetic operators of reproduction, crossover and mutation. These operators conduct the chromosomes toward better fitness.

Selection operator selects the chromosome in the population for reproduction. The more fit the chromosome, the higher its probability of being selected for reproduction. Thus, selection is based on the survival-of-the-fittest strategy, but the key idea is to select the better individuals of the population, as in tournament selection, where the participants compete with each other to remain in the population [15]. After selection of the pairs of parent strings, the crossover operator is applied to each of these pairs.

The crossover operator involves the swapping of genetic material (bit-values) between the two parent strings. Based on predefined probability, known as crossover probability, an even number of chromosomes are chosen randomly. A random position is then chosen for each pair of the chosen chromosomes. The two chromosomes of each pair swap their genes after that random position. In this work, crossover is used with probability of 0.7.

Each individuals (children) resulting from each crossover operation will now be subjected to the mutation operator in the final step to forming the new generation. The mutation operator enhances the ability of the GA to find a near optimal solution to a given problem by maintaining a sufficient level of genetic variety in the population, which is needed to make sure that the entire solution space is used in the search for the best solution. In a sense, it serves as an insurance policy; it helps prevent the loss of genetic material [15].

In this work mutation is used with probability of 0.2 per bit. The process continues and it is terminated after production of 50 generations (iterations). Regarding the fact that the goal is optimization of PID parameters (K_P, K_I and K_D), the selected chromosome is as Figure 2.

The flowchart of the proposed GA, for this purpose is shown in figure 3. It should be mentioned that number of initial population are considered 50 for our case study.

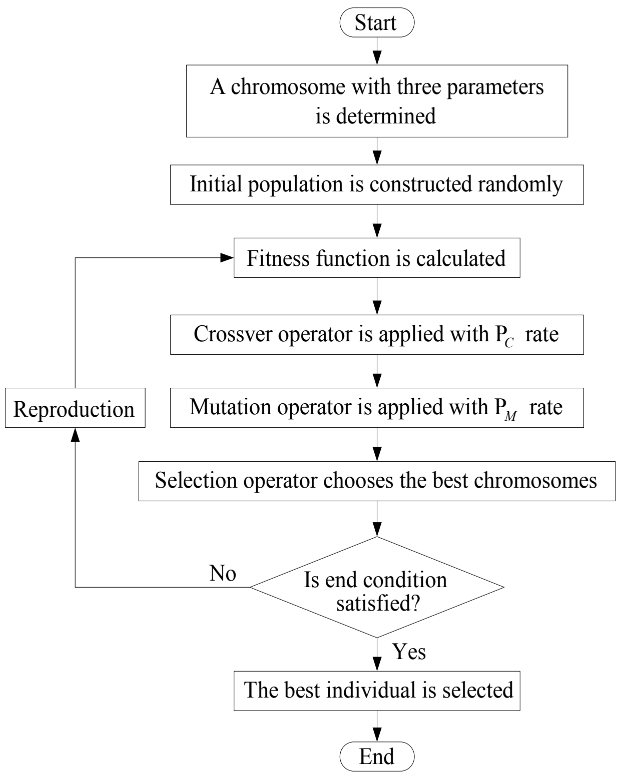

Figure 3. Flowchart of the proposed GA

C. Particle Swarm Optimization Algorithms

PSO algorithm, which is tailored for optimizing difficult numerical functions and based on metaphor of human social interaction, is capable of mimicking the ability of human societies to process knowledge [11]. It has roots in two main component methodologies: artificial life (such as bird flocking, fish schooling and swarming) and, evolutionary computation. Its key concept is that potential solutions are flown through hyperspace and are accelerated towards better or more optimum solutions. Its paradigm can be implemented in simple form of computer codes and is computationally inexpensive in terms of both memory requirements and speed. It lies somewhere in between evolutionary programming and the genetic algorithms.

As in evolutionary computation paradigms, the concept of fitness is employed and candidate solutions to the problem are termed particles or sometimes individuals, each of which adjusts its flying based on the flying experiences of both itself and its companion. It keeps track of its coordinates in hyperspace which are associated with its previous best fitness solution and also of its counterpart corresponding to the overall best value acquired thus far by any other particle in the population.

Vectors are taken as presentation of particles since most optimization problems are convenient for such variable presentations. In fact, the fundamental principles of swarm intelligence are adaptability, diverse response, proximity, quality and stability [16]. It is adaptive corresponding to the change of the best group value. The allocation of responses between the individual and group values ensures a diversity of response. The higher dimensional space calculations of the PSO concept are performed over a series of time steps. The population is responding to the quality factors of the previous best individual values and the previous best group values. The principle of stability is adhered to since the population changes its state if and only if the best group value changes.

As it is reported in [11], this optimization technique can be used to solve many of the same kinds of problems as GA and does not suffer from some of GAs difficulties. It has also been found to be robust in solving problem featuring non-linearing, non-differentiability and high dimensionality. It is the search method to improve the speed of convergence and find the global optimum value of fitness function.

PSO starts with a population of random solutions "particles" in a D-dimension space. The ith particle is represented by $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. Each particle keeps track of its coordinates in hyperspace, which are associated with the fittest solution it has achieved so far. The value of fitness for particle i is stored as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ that its best value is represented by (pbest). The global version of the PSO keeps track of the overall best value (gbest) and its location, obtained thus far by any particle in the population. PSO consists of, at each step, changing velocity of each particle toward its pbest and gbest according to Equation (3). The velocity of particle i is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$.

Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and gbest. The position of the ith particle is then updated according to Equation (4) [12, 14]:

$$v_{id}(t+1) = \omega \times v_{id}(t) + c_1 r_1 (P_{id} - x_{id}(t)) + c_2 r_2 (P_{gd} - x_{id}(t)) \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + c v_{id}(t+1) \quad (4)$$

where, P_{id} and P_{gd} are *pbest* and *gbest*. It is concluded that *gbest* version performs best in terms of median number of iterations to converge.

However, *pbest* version with neighborhoods of two is most resistant to local minima. The results of past experiments about PSO show that ω was not considered at an early stage of PSO algorithm. However, ω affects the iteration number to find an optimal solution. If the value of ω is low, the convergence will be fast, but the solution will fall into the local minimum. On the other hand, if the value will increase, the iteration number will also increase and therefore the convergence will be slow. Usually, for running the PSO algorithm, value of inertia weight is adjusted in training process. It was shown that PSO algorithm is further improved via using a time decreasing inertia weight, which leads to a reduction in the number of iterations [16]. In Equation (3), term of $c_1 r_1 (P_{id} - x_{id}(t))$ represents the individual movement and term of $c_2 r_2 (P_{gd} - x_{id}(t))$ represents the social behavior in finding the global best solution.

D. Improved PSO Algorithm

According to Equation (3), the velocity update of the particle consists of three parts: The first term is its own current velocity of particles; the second term is cognitive part which represents the particle's own experiences; the third term is social part which represents the social interaction between the particles [17-19]. With respect to (3), it is realized that best position of particles take places proportional to $pbest_i$. It can be seen that: when a particle's current position coincides with the global best position ($gbest_i$), the particle will only leave this point if the inertia weight and its current velocity are different from zero. If the particles' current velocities in swarm are very close to zero, then these particles will not move once they catch up with the global best particle, which means that all the particles will converge to the best position (*gbest*) discovered so far by the swarm [20]. At this moment if this positions corresponding fitness is not the problems expected global optimal, then the premature convergence phenomenon appears.

In order to overcome this drawback and improve optimization synthesis, an Improved Particle Swarm Optimization (IPSO), by introducing the mutation operator often used in genetic algorithm [15] is proposed in this study. This process can make some particles jump out local optima and search in other area of the solution space. In this method, the mutation probability (P_M) is dynamically adjusted according to the diversity in the swarm. The goal with mutation probability is to prevent the PSO to converge prematurely to local minima.

It should be noted the P_M is considered 0.1 in this study. Figure 4 shows the flowchart of the improved PSO algorithm.

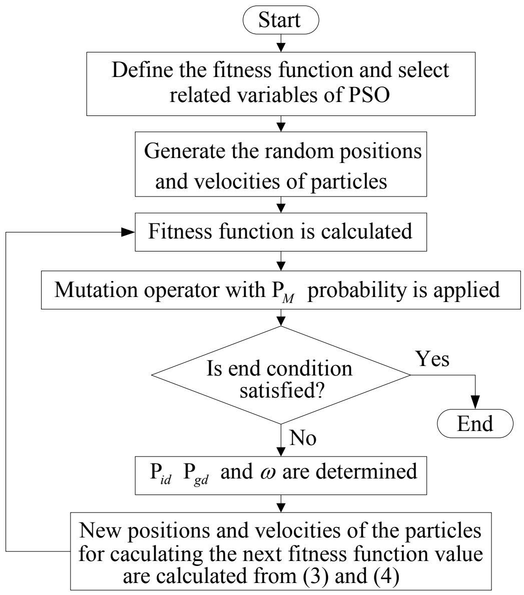

Figure 4. Flowchart of the IPSO algorithm

E. Advanced PSO Algorithm

High searching speed is essential in determining the proper parameters when much iteration is involved. Therefore, several methods have been proposed to improve the PSO algorithm speed and convergence toward the global minimum until now. One method to use is the advanced PSO algorithm. This technique can improve PSO performance by putting the adaptively changing terms. These changing terms are caused that the parameters of the original PSO algorithm can change according to the convergence rate which is presented by the fitness. Thus, the original PSO is changed like this [21-22]:

$$\begin{aligned} r_1 &= 1 - \frac{P_{id}}{P_i} + rand \\ r_2 &= 1 - \frac{P_{gd}}{P_i} + rand \end{aligned} \quad (5)$$

where, *rand* is the random value between 0 and 1. r_1 can influence the movement of the second term (individual term) as a weight factor. In early searching stage, the difference of between pbest and gbest are the fitness values at the best position of between pbest and P_i is relatively bigger than that in the last stage.

Accordingly, the value of $(1 - P_{id} / P_i)$, is also bigger than that in the last stage. As an individual particle approaches near the individual best position, the movement of individual particle becomes gradually slow. So we can expect faster convergence than the original. r_2 has an effect on the movement of the third term (group). Likewise, it is interpreted as follows:

$$P_{gd} \le P_{id} \le P_i \quad (6)$$

Because *gbest* is supposed as optimal and lowest value in entire particles' fitness values, Equation (2) can be derived. Equation (10) can be easily derived from Equation (9). If the particles converge to the optimal value, *pbest* and P_i will have the same value, *gbest*. Therefore, the replaced $(1 - P_{id} / P_i)$ and $(1 - P_{gd} / P_i)$ will become zero, so that the second and third terms will move slowly. It can derive the fast searching. Figure 5 shows the flowchart of the advanced PSO algorithm.

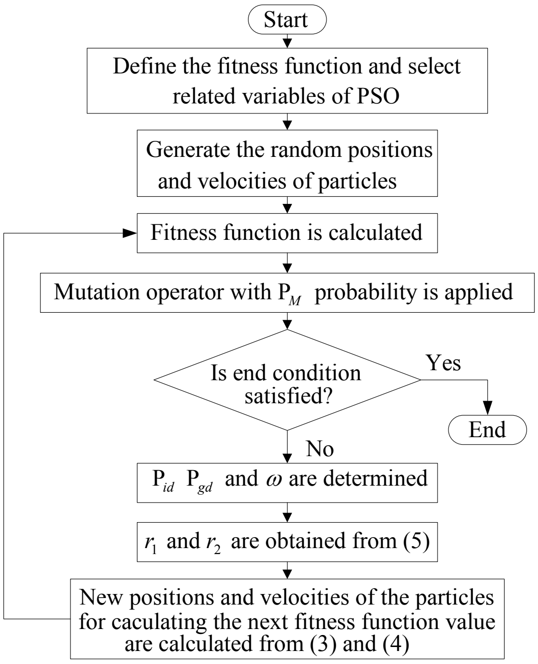

Figure 5. Flowchart of the APSO algorithm

III. MODELING OF THE DC MOTOR

In normal conditions and without controller, the DC motor does not have an acceptable performance, this fact will be shown in later sections. Because of analysis a DC motor and show its performance; in this section it is described how to develop a linear model for a DC motor, and how to analyze the model under Matlab/Simulink. For this we need a conceptual realization of a DC motor.

A. Physical System

Electric circuit of the armature and the free body diagram of the rotor of a DC motor, are shown in Figure 6. The rotor and the shaft are assumed to be rigid. Consider the following values for the physical parameters:

- moment of inertia of the rotor J =0.01 kg/ m^2;
- damping of the mechanical system b =0.1 N.m.s;
- (back-) electromotive force constant K =0.01 N.m/A;
- electric resistance R = 1Ω;
- electric inductance L = 0.5. H

The input is the armature voltage, V_{in}, (driven by a voltage source). Measured variables are the angular velocity of the shaft (ω) in radians per second, and the shaft angle (θ) in radians.

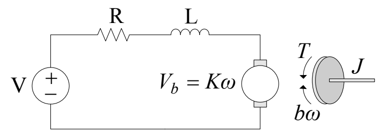

Figure 6. Schematic representation of the considered DC motor

B. System Equations

The motor torque, T, is related to the armature current, i, by a constant factor K:

$$T = Ki \tag{7}$$

The back electromotive force (emf), V_b, is related to the angular velocity by:

$$V_b = K\omega = K\frac{d\theta}{dt} \tag{8}$$

From Figure 6, we can write the following equations based on the Newton's law combined with the Kirchhoff's law:

$$J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = Ki \tag{9}$$

$$L\frac{di}{dt} + Ri = V - K\frac{d\theta}{dt} \tag{10}$$

C. Transfer Function

Using the Laplace transform, (9) and (10) can be written as:

$$Js^2\theta(s) + b\theta(s) = KI(s) \tag{11}$$

$$LsI(s) + RI(s) = V(s) - Ks\,\theta(s) \tag{12}$$

where, s denotes the Laplace operator. From (12) we can express $I(s)$:

$$I(s) = \frac{V(s) - Ks\theta(s)}{R + Ls} \tag{13}$$

and substitute $I(s)$ in (11) to obtain:

$$Js^2\theta(s) + bs\theta(s) = K\frac{V(s) - Ks\theta(s)}{R + Ls} \tag{14}$$

This equation for the DC motor is shown in the block diagram in Figure 7. From (14), the transfer function from the input voltage ($V(s)$) to the output angle (θ) directly follows:

$$G_a(s) = \frac{\theta(s)}{V(s)} = \frac{K}{s\left((R + Ls)(Js + b) + K^2\right)} \tag{15}$$

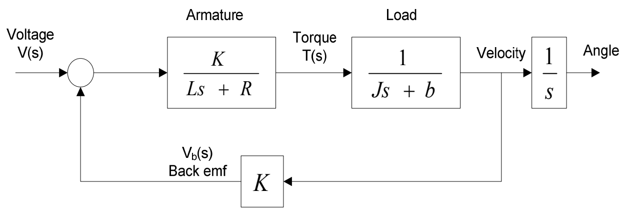

Figure 7. The block diagram of the DC motor

From the block diagram in Figure 7, it is easy to see that the transfer function from the input voltage ($V(s)$) to the angular velocity (ω) is:

$$G_a(s) = \frac{\omega(s)}{V(s)} = \frac{K}{(R+Ls)(Js+b)+K^2} \quad (16)$$

In this study, in order to acquire better performance and fast convergence of the PSO algorithms, parameters which are used in these algorithms have been initialized according to Table 1. It should be noted that mentioned algorithms are run several times and then optimal results are selected.

Table 1. Value of parameters for PSO, IPSO and APSO algorithm

	Method		
Parameter	PSO	IPSO	APSO
Problem dimension	3.0	3.0	3.0
Number of particles	30.0	30.0	20.0
Number of iterations	30.0	50.0	50.0
C_1	0.5	1.0	0.5
C_2	0.5	1.5	0.5
ω_{max}	0.9	0.9	0.9
ω_{min}	0.4	0.4	0.4

IV. RESULTS AND DISCUSSIONS

In this section, The GA, PSO IPSO, and APSO are applied to find the optimal parameters of the PID controller for the closed loop controlled DC motor. Without controller, the DC motor in this case has a slow step response according to Figure 8. We use PID controller to improve the step response of the motor. The transfer function of the DC motor is described in Equation (16). The mentioned approaches (APSO, IPSO, PSO and GA) are tested on this case study system and results are given in Tables 2 and 3.

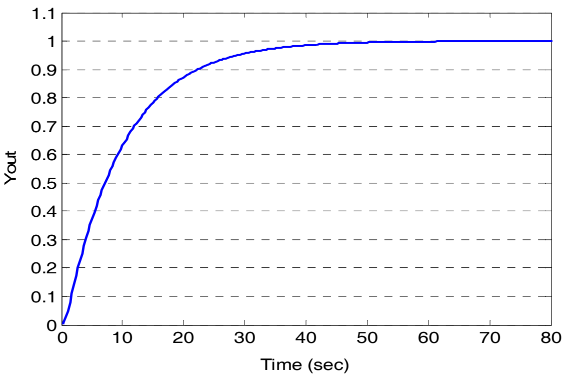

Figure 8. Step response of the DC motor

Table 2. Calculated parameters of PID controllers by PSO algorithms and GA

	Parameters		
Methods	K_P	K_I	K_D
APSO	46.2943	0.0063	27.8531
IPSO	51.7622	0.0009	29.005
PSO	69.1955	0.0063	28.9303
GA	35.1915	0.4039	18.2127

The corresponding performance index curves of the PSO methods and GA are depicted in Figure 9. Also Figure 10 describes the step response curve of APSO, PSO and GA approaches.

Table 3. Calculated parameters of fitness function by PSO algorithms and GA

	Parameters			
Methods	t_s	t_r	M_P	E_{ss}
APSO	0.3	0.45	0.0161	0.0
IPSO	0.55	0.4	0.0387	0.0
PSO	0.779	0.328	0.1092	0.0
GA	0.55	0.8	0.003	0.0

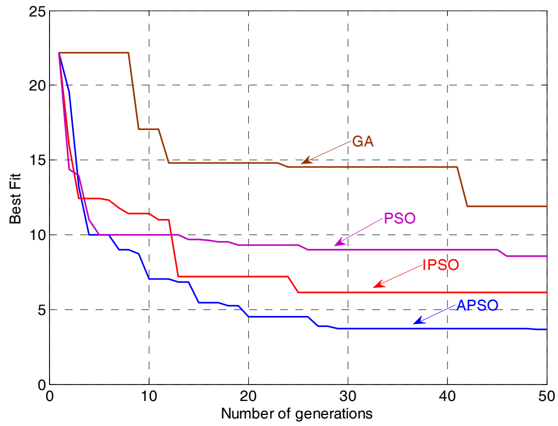

Figure 9. Convergence curves of mentioned algorithms

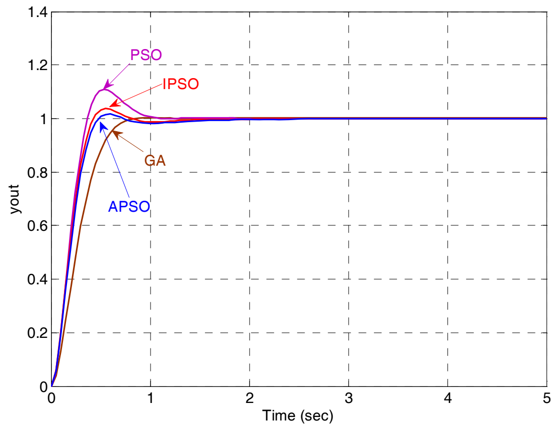

Figure 10. Step response of the PID controlled DC motor

As shown in Figure 9, it can be found easily that solution of the optimization problem by APSO is caused that the fitness function is more optimized than other mentioned approaches. Also, Figure 10 shows that the obtained dynamic quality of step response by APSO is better than PSO and even IPSO method. Moreover, the settling time of system output is obtained by APSO approach is less than other solution methods (GA, PSO and IPSO). Finally, regarding to results evaluation of both case studies, it can be concluded that PID controller design by APSO algorithm is better than GA, PSO and even IPSO one.

V. CONCLUSIONS

In this paper, the PID controller has been designed and optimized by Advanced Particle Swarm Optimization (APSO) algorithm. The proposed method is tested on DC motor in comparison with GA, PSO and IPSO approaches in order to demonstrate its effectiveness and robustness for solution of the desired optimization problem. Results evaluation reveals that the precision and convergence speed of proposed APSO based method is more than GA, PSO and even IPSO. In addition, design of PID controller using APSO is caused that the rate of overshoot in step response curve is reduced in comparison with PSO and IPSO. In other words, although the PSO and IPSO are more conventional for optimization aims but APSO causes the amount of fitness function is calculated more precisely and therefore more optimal solutions are obtained. It should be noted that APSO performance in design and optimization process can be more improved by increasing the number of iterations.

REFERENCES

[1] K.J. Astrom, T. Hagglund, "PID Controllers: Theory, Design and Tuning", International Society for Measurement and Con., 2nd Ed., pp. 200-210, 1995.
[2] F.D. Bianchi, R.J. Mantz, C.F. Christiansen, "Multivariable PID Control with Set-Point Weighting via BMI Optimization", Automatica, Vol. 44, pp. 472-478, 2008.
[3] J.G. Zigeler, N.B. Nichols, "Optimization Setting for Automatic Controller", Trans. ASME, Vol. 64, pp. 756-769, 1942.
[4] L. Zhang, "Simplex Method based Optimal Design of PID Controller", Inform. Control, Vol. 33, pp. 376-379, 2004.
[5] H. Han, A. Luo, "Nonlinear PID Controller Based on Genetic Tuning Algorithm", Control Decision, Vol. 20, pp. 448-450, 2005.
[6] C. Li, J. Lian, "The Application on Immune Genetic Algorithm in PID Parameter Optimization for Level Control System", IEEE International Conference on Automation and Logistics, Jinan, China, pp. 782-786, 2007.
[7] J.S. Yang, "PID Control for a Binary Distillation Column Using a Genetic Searching Algorithm", WSEAS Trans. Syst., Vol. 5, pp. 720-726, 2006.
[8] T.H. Kim, I. Maruta, T. Sugie, "Robust PID Controller Tuning Based on the Constrained Particle Swarm Optimization", Automatica, Vol. 44, pp. 1104-1110, 2008.
[9] R.A. Krohling, J.P. Rey, "Design of Optimal Disturbance Rejection PID Controllers Using Genetic Algorithm", IEEE Trans. Evol. Comput., Vol. 5, pp. 78-82, 2001.
[10] D.B. Fogel, "Evolutionary Computation: Toward a New Philosophy of Machine Intelligence", IEEE Press Series on Computational Intelligence, 3rd Edition, New York, pp. 128-138, 2006.
[11] H. Shayeghi, A. Jalili, H.A. Shayanfar, "Multistage Fuzzy Load Frequency Control Using PSO", Energy Convers. Manage., Vol. 49, pp. 2570-2580, 2008.
[12] J. Kennedy, R. Eberhart, Y. Shi, "Swarm Intelligence", Morgan Kaufmann Publishers, 1st Edition, San Francisco, pp. 80-95, 2001.
[13] Z.L. Gaing, "A Particle Swarm Optimization Approach for Design of PID Controller in AVR System", IEEE Trans. Energy Convers., Vol. 19, pp. 384-391, 2004.
[14] S. Jalilzadeh, A. Kazemi, H. Shayeghi, M. Mahdavi, "Technical and Economic Evaluation of Voltage Level in Transmission Network Expansion Planning Using GA", Energy Convers. Manage., Vol. 49, pp. 1119-1125, 2008.
[15] H. Shayeghi, S. Jalilzadeh, M. Mahdavi, H. Haddadian, "Studying Influence of Two Effective Parameters on Network Losses in Transmission Expansion Planning Using DCGA", Energy Convers. Manage., Vol. 49, pp. 3017-3024, 2008.
[16] M. Clerc, J. Kennedy, "The Particle Swarm Explosion, Stability and Convergence in a Multidimensional Complex Space", IEEE Trans. Evol. Comput., Vol. 6, pp. 301758-73, 2002.
[17] J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge", Proceeding of the IEEE International Conference on Evolutionary Computation, ICEC1997, Indianapolis, pp. 303-308, 1997.
[18] M.M. Khodier, C.G. Christodoulou, "Linear Array Geometry Synthesis with Minimum Sidelobe Level and Null Control Using Particle Swarm Optimization", IEEE Trans. Antennas Propagat., Vol. 53, pp. 2674-2679, 2005.
[19] S. He, J.Y. Wen, E. Prempain, Q.H. Wu, J. Fitch, S. Mann, "An Improved Particle Swarm Optimization for Optimal Power Flow", International Conference on Power System Technology, Powercon 2004, Singapore, pp. 1633-1637, 2004.
[20] A.A.A.A. Esmin, G.L. Torres, A.C.Z. de Souza, "A Hybrid Particle Swarm Optimization Applied to Loss Power Minimization", IEEE Trans. Power Syst., Vol. 20, pp. 859-866, 2005.
[21] S. Jalilzadeh, A. Kimiyaghalam, A. Ashouri, "Application of IADPSO Approach for TNEP Problem Considering of Loss and Uncertainty in Load Growth", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 4, Vol. 1, No. 3, pp. 37-43, September 2010.
[22] S. Jalilzadeh, A. Kimiyaghalam, A., Ashouri, "Evaluating the Effects of Uncertainty in Fuel Price on Transmission Network Expansion Planning Using IADPSO Approach", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 7, Vol. 3, No. 2, pp. 1-9, June 2011.

BIOGRAPHIES

Abolfazl Jalilvand was born in Takestan, Iran, 1972. He received the B.Sc. degree in Electrical and Electronic Engineering from Electrical and Computer Engineering Faculty of Shahid Beheshti University, Tehran, Iran, in 1995. He received M.Sc. and Ph.D. degrees from Electrical and Computer Engineering Faculty of University of Tabriz, Tabriz, Iran, in Power Engineering and Control Engineering in 1998 and 2005, respectively. After completing his Ph.D., he joined the Electrical

Engineering Department of University of Zanjan, Zanjan, Iran, as an assistant professor where he was head of Electrical Engineering department from 2008 to 2010. Currently, he is the dean of faculty of engineering and an associate professor at the same university. He has more than 90 papers in journals and conferences. His main research interests include the hybrid control systems, Petri nets, intelligent control, modeling and control of power electronic converters, control and stabilization of power systems, application of intelligent methods in power systems and so on. He is a member of the institute of electrical and electronics engineers (IEEE).

Ali Kimiyaghalam was born in Tehran, Iran, 1983. He received the B.Sc. degree in Electrical Engineering from Islamic Azad University, Abhar Branch, Iran and M.Sc. degree in Electrical Engineering from University of Zanjan, Zanjan, Iran in 2007 and 2009, respectively. He has more than 35 papers in journals and conferences. His research interests are in the area of power system planning, reliability and power quality.

Ahmad Ashouri was born in Khodabandeh, Iran, 1984. He received the B.Sc. degree in Electrical Engineering from Islamic Azad University, Abhar Branch, Iran and M.Sc. degree in Electrical Engineering from University of Zanjan, Zanjan, Iran in 2007 and 2009, respectively. He joined to Islamic Azad University, Khodabandeh Branch, Iran, as lecturer in 2010. Currently, he is a Ph.D. candidate at Islamic Azad University, Science and Research Branch, Tehran, Iran. He has more than 15 papers in journals and conferences. His areas of research interest are in the area of Petri nets, power system planning, reliability and power quality.

Hossein Kord was born in Zanjan, Iran, 1985. He received B.Sc. degree from University of K.N. Toosi (Tehran, Iran) and M.Sc. degree from University of Zanjan (Zanjan, Iran), all in Power Electrical Engineering, in 2008, and 2010 respectively. Currently, he works as an Electrical Engineer. His research activities are focused on renewable energies, power system planning, reliability and power quality.