The reduction coefficient of PID controller by using PSO algorithm method for Flexible single-arm robot system

Mohammad Mehdi Moradi

Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran.

Email: mehdimoradi8797@gmail.com

ABSTRACT

This study on the design of PID controllers for flexible single-arm robot system optimization PSO method is focused so that the coefficients of the PID controller are reduced. In this study, PID controller and PSO algorithm have been described and then by using MATLAB, PID control was simulated. Then by PSO algorithm, attempts to reduce the PID coefficients are given by simulation. Finally PID coefficients' values were compared with and without the PSO algorithm. The results showed that by using the number of birds and birds number steps, both equal to 30 (the sixth), the lowest values of the coefficients K_p, K_d ,K_i *are 0.741, 0.1491and 0, respectively.*

KEYWORDS: PID, PSO, single-arm robot, reduction of coefficients

1. INTRODUCTION

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an error value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process through the use of a manipulated variable. The PID controller algorithm involves three separate constant parameters, and is accordingly sometimes called three-term control: proportional, integral, and derivative. Unlike the simple form of PID, designing this controller is practically something more than tuning three parameters. Different functions affect the implementation of this controller including: controller structure, the process grade, constant ratio time system dominant to dead-time processes, the dynamics of the actuator element, the type of filter in derivation unit and tuning its parameter, non-linear treatment, and so on.

Each one of these functions can play a role in designing and tuning process of PID controller. Among these, we have two main subjects involved in designing derivation unit filter; compensating effects of active saturated elements in computer science, particle swarm optimization as a calculation way that can operate repetitive problems by trying to optimize a voluntary solution according to given calculations of qualification [16]. Most designing methods aim at maximizing efficiency and tuning controller parameters by improving design methods such as BFA algorithms that are used on PID controller. In this study, PID controller coefficient has been minimized by PSO algorithm and implicit application of this controller is maximized. We have used MATLAB software in order to implement this study and also PSO algorithm has been written in MATLAB software by using programming and minimizing PSO coefficient being surveyed and the calculation has been done based on obtained limitations of PSO algorithm on the single flexible arm [15,17].

2. DESIGNING, TUNING, AND IMPLEMENTING PID CONTROLLER

The controlled process is considered to be based on the following close-loop circuit.

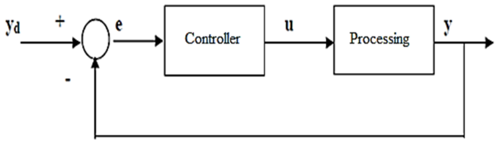

Fig .1. Diagram block of process controller with feedback loop

In many industrial processes, proportional (P), proportional-derivation (PD), proportional-integration (PI), or proportional-derivation-integration (PID) has been used as the main structure. The general form of PID controller is as follows:

$$u(t) = K\left(e(t) + \frac{1}{T_i}\int_0^t e(c)dc + T_d \frac{de(t)}{dt}\right) \qquad (1)$$

Where, u is controller command and e process error (e = yd - y). PID controller is formed by three terms.

1. Proportional term: controller command is amplified proportion to error rate and K interest.
2. Derivation term: controller command is proportional to rate of error variation.
3. Integration: controller command is proportional to additional error function from time zero to now so that the integration of this function is varied.

PID controller parameter consisted K, T_i, T_d where k is time efficiency, T_i is integration constant time, T_d derivation constant time. Now we are trying to show how these three elements work and what the reason is available on feedback loop [1].

If we consider the integration term of PID controller, in this case u(t)=Ke(t). The most important thing to consider is to pay attention to investigate on statistical systems.

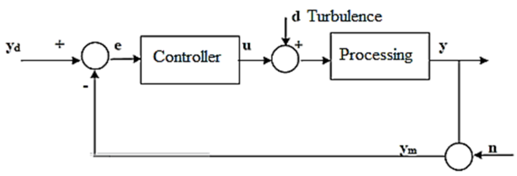

Fig .2. Block diagram of process control with feedback and noise and turbulence

Where, d is turbulence rate and n is environment noise implemented on system. In the determined circuit transfer function, we have:

$$y = \frac{G(s)k}{1+G(s)K} y_d + \frac{G(s)}{1+G(s)K} d - \frac{G(s)k}{1+G(s)K} n \qquad (2)$$

System interest rate is G(s)k rate in forward circuit which is called loop gain and it is shown by L(s) symbol. Note that the main purpose of making controller loop is tuning y to y_d output similarly. Mathematical interpretation of this suggestion is making function conversion closer to unit number as

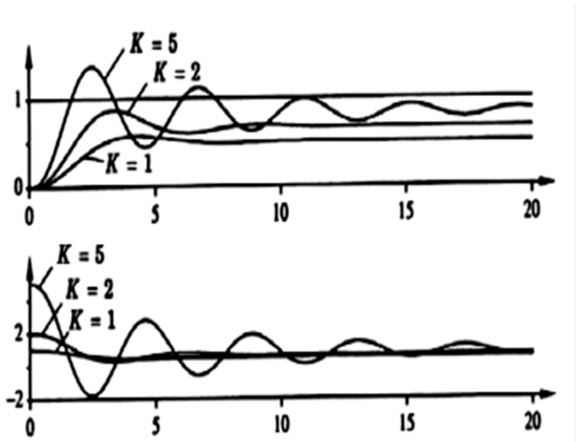

Fig .3.Increasing effect of control gain in stability and responsibility of system

follows:

$$\frac{y}{y_d} = \frac{G(s)K}{1 + G(s)K} = \frac{L(s)}{1 + L(s)} \qquad (3)$$

Loop interest rate L(s) is increased by increasing proportional controller interest k and the rate of $\frac{L(s)}{1+L(s)}$ is approached to unit number. Therefore, proportional interest is more effective regarding output tuning. On the other hand, we try to increase turbulence effects d over output.

$$\frac{y}{d} = \frac{G(s)}{1 + G(s)K} = \frac{G(s)}{1 + L(s)} \qquad (4)$$

It means that this function must be tended closing to zero. Interest existence k is again caused to be maximized by increasing loop interest k in denominator of the fraction L(s) and arbitrarily this conversion function is made closer to zero. Therefore we have more accuracy in output tuning and more weakness on turbulence effects. But control loop gain can be increased without any attention to other subjects, for example noise effects of measuring n on output.

$$\frac{y}{n} = \frac{L(s)}{1 + L(s)} \qquad (5)$$

This conversion function is approached to unit number by indiscriminate increase of k and loop gain L(s). It means that noise effect is increasingly observed as 100 percent in output and there is no ability for noise effect weakness. On the other hand, the stability of close-loop system is decreased by increasing control interest k.

For these two reasons, we must generally specify optimization rate for control interest of proportional control k to achieve the desired stability margin. One example of the effect of proportional control k on a process is shown in following figure. The system persistent error is decreased by increasing control gain but the response will be unstable and noisier.

2.1.Basic PSO Algorithm Model

Kennedy and Eberhart first established a solution for the complex non-linear optimization problem by imitating the behavior of bird flocks. They generated the concept of function-optimization by means of a particle swarm [2]. Consider the global optimum of an n-dimensional function defined by

$$f(x\_1, x\_2, x\_3, \cdots, x\_n) = f(x) \qquad (6)$$

Where, x_i is the search variable, which represents the set of free variables of the given

function. The aim is to find a value for x^*such that the function $f(x^*)$ is either a maximum or a minimum in the search space.
Consider the functions given by

$$f_1 = x_1^2 + x_2^2 \tag{7}$$

$$f_2 = x_1 \sin(4\pi x_2) - x_2 \sin(4\pi x_1 + \pi) + 1 \tag{8}$$

From the fig. 4 (a), it is clear that the global minimum of the function f1 is $(x_1, x_2) = (0,0)$, i.e. at the origin of function f1 in the search space. That means it is a unimodel function, which has only one minimum. However, is not so easy to find the global optimum for multi-model functions, which have multiple local minima. Fig. 4 (b) shows function f2 which has a rough search space with multiple peaks.

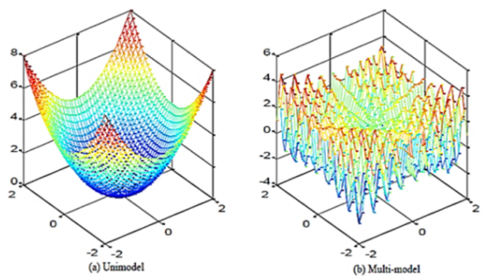

Fig. 4. Plot of the functions f_1 and f_2 [3]

So many agents have to start from different initial locations and continue exploring the search space until at least one agent reaches the global optimal position. During this process all agents can communicate and share their information among themselves [3]. This thesis discusses how to solve the multi-model function problems. The Particle Swarm Optimization (PSO) algorithm is a multi-agent parallel search technique which maintains a swarm of particles and each particle represents a potential solution in the swarm [4]. All particles fly through a multidimensional search space where each particle is adjusting its position according to its own experience and that of neighbors. Suppose x_i^t denote the position vector of particle i in the multidimensional search space (i, R^n) in the multidimensional search space by

$$x_i^{t+1} = x_i^t + v_i^{t+1} \text{ with } x_i^0 \sim U(x_{min}, x_{max}) \tag{9}$$

Where v_i^t is the velocity vector of particle that drives the optimization process and reflects both the self-experience knowledge and the social experience knowledge of all the particles.
$U(x_{max}, x_{min})$ Represents the uniform distribution where x_{min} and x_{max} are minimum and maximum values, respectively.
Therefore, in a PSO method, all particles are initiated randomly and evaluated to compute fitness along with finding the personal best (best value of each particle) and global best values (best value of particle in the entire swarm). After that a loop starts to find an optimum solution. In the loop, first the particles' velocity is updated by the personal and global bests, and then each particle's position is updated by the current velocity. The loop is ended with a predetermined stopping criterion [5]. Basically, two PSO algorithms, namely the Global Best (gbest) and Local Best (lbest) PSO, have been developed which differ in the size of their neighborhoods. These algorithms are discussed in Sections 3.1.1 and 3.1.2, respectively.
The global best PSO (or gbest PSO) is a method where the position of each particle is

influenced by the best-fit particle in the entire swarm. It uses a star social network topology (Section 3.5) where the social information is obtained from all particles in the entire swarm [6] [7]. In this method each individual particle, $i \in [1,...,n]$ where $N \rangle 1$ has a current position in search space x_i current velocity v_i and a personal best position in search space, $P_{best,i}$. The personal best position $P_{best,i}$ corresponds to the position in search space where particle has had the smallest value as determined by the objective function f, considering a minimization problem. Additionally, the position yielding the lowest value amongst all the personal best $P_{best,i}$ is called the global best position which is denoted by G_{best} [4]. The following equations (10) and (11) define how the personal and global best values are updated, respectively.
Considering minimization problems, the personal best position $P_{best,i}$ at the next time step, $t+1$, is calculated as, where $t \in [0,...,n]$

$$P_{best,i}^{t+1} = \begin{cases} P_{best,i}^{t} \\ x_i^{t+1} \end{cases} \Rightarrow \begin{cases} if\, f\left(x_i^{t+1}\right) > P_{best,i}^{t} \\ if\, f\left(x_i^{t+1}\right) \le P_{best,i}^{t} \end{cases} \quad (10)$$

Where $f : \mathbb{R}^n \to \mathbb{R}$ is the fitness function, the global best position G_{best} at time step is calculated as [9]:

$$G_{best} = min\ \left\{P_{best,i}^{t}\right\}\ ,where\ i \in [1,...,n]\ and\ n > 1 \quad (11)$$

Table 1: Ziegler-Nichols ultimate sensitivity test [12]

Controller type	K_p	T_i	T_d
P	$0.05K_{cr}$	∞	0
PI	$0.45K_{cr}$	$0.833T_{cr}$	0
PID	$0.6K_{cr}$	$0.5T_{cr}$	$0.125T_{cr}$

Therefore it is important to note that the personal best $P_{best,I}$ is the best position that the individual particle has visited since the first time step. On the other hand, the global best position G_{best} is the best position discovered by any of the particles in the entire swarm [4]. For gbest PSO method, the velocity of particle i is calculated by

$$v_{ij}^{t+1} = v_{ij}^{t} + C_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + C_1 r_{1j}^{t}\left[L_{best,i}^{t} - x_{ij}^{t}\right]$$

Where, v_{ij}^t is the velocity vector of particle in dimension j at time t; x_{ij}^t is the position vector of particle in dimension at time; $P_{best,i}^t$ is the personal best position of particle in dimension found from initialization through time t; G_{best} is the global best position of particle in dimension found from initialization through time t; c_1 and c_2 are positive acceleration constants which are used to level the contribution of the cognitive and social components, respectively; r_{1j}^t and r_{2j}^t are random numbers from uniform distribution $U(0,1)$ at time t [10].

2.2. Ziegler-Nichols Rules for tuning PID Controller

Two tuning methods were proposed by Ziegler and Nichols in 1942 and have been widely utilized either in the original form or in modified forms. One of them, referred to as Ziegler–Nichols ultimate sensitivity method, is to determine the parameters as given in Table 1 using the data K_{cr} and T_{cr} obtained from the ultimate sensitivity test. The other, referred to as Ziegler–Nichols step response method, is to assume the model FOPDT and to determine the parameters of the PID controller as given in Table 2 using the parameters R and L of FOPDT which are determined from the step response test [11].

Frequency-domain stability analysis tells that the above way of applying the Ziegler–Nichols step response method to processes with self-regulation tends to set the parameters on the safe side, in the sense that the actual gain and phase margins become larger than the values expected in the case of integrating processes.

These methods to determine PID parameter using empirical formula, as well as several other tuning methods developed on the same principle, are often referred to as “classical” tuning methods. Some of the other classical tuning methods are, Chien–Hrones–Reswick formula, Cohen–Coon formula, refined Ziegler–Nichols tuning, Wang–Juang–Chan formula.

The control system performs poor in characteristics and even it becomes unstable, if improper values of the controller tuning constants are used.

So it becomes necessary to tune the controller parameters to achieve good control performance with the proper choice of tuning constants [13, 14].

Table 2: Ziegler-Nichols step response method ($RL \neq 0$) [12]

Controller type	K_p	T_i	T_d
P	1/RL	∞	0
PI	0.9/RL	L/0.3	0
PID	1.2/RL	2L	0.05L

3. SCHEDULING PSO FOR PID CONTROLLER PARAMETERS

In this paper, a PID controller has used PSO Algorithms to find the optimal parameters of DC motor speed control system. The structure of the PID controller with PSO algorithms is shown in Fig 5.

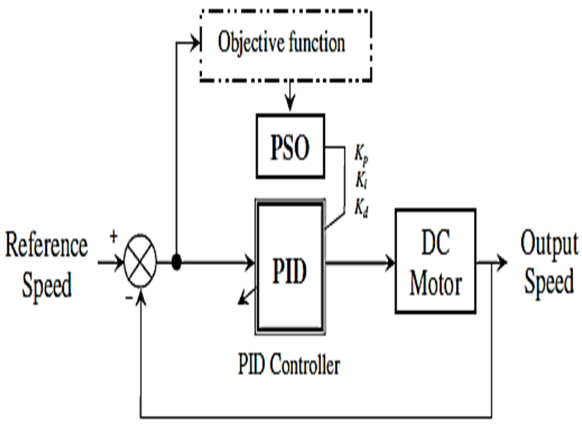

Fig.5. The block diagram of proposed PID Controller with PSO algorithms

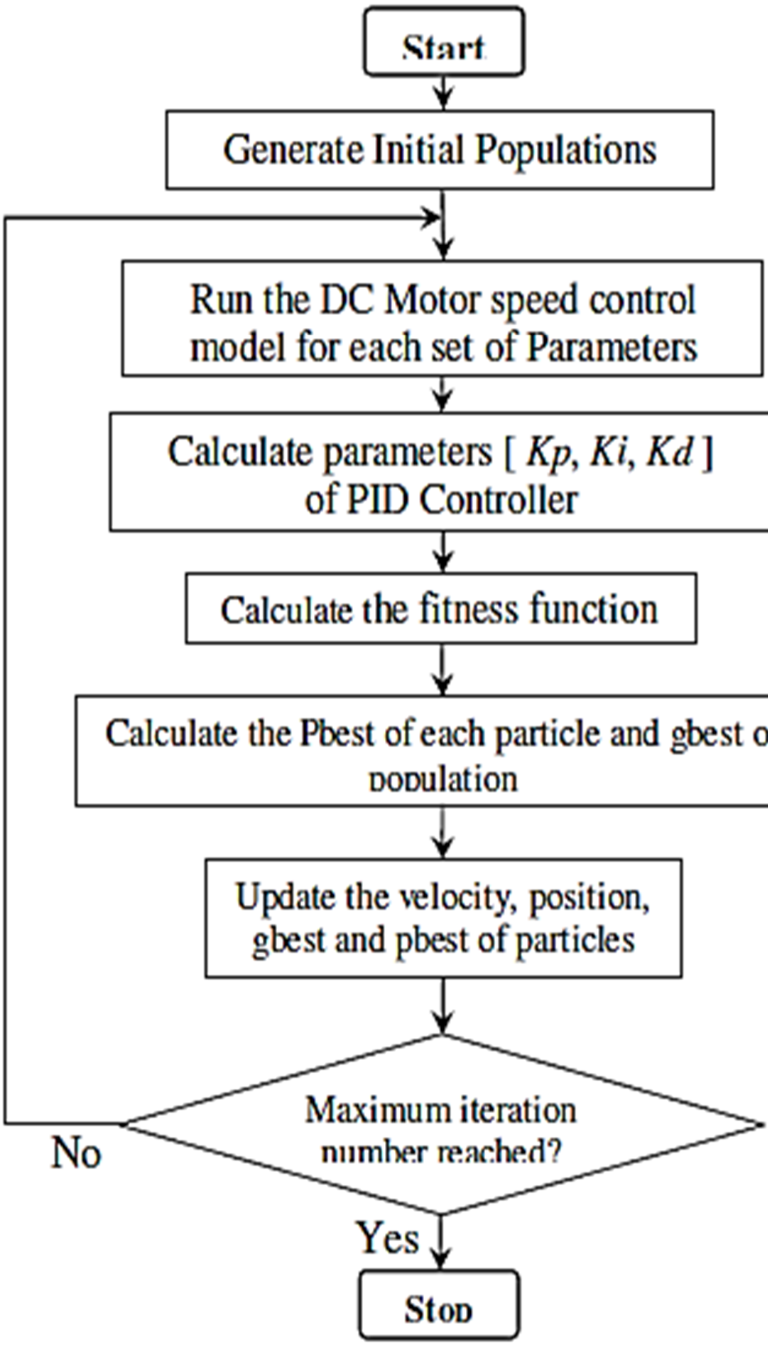

Fig.6. The flowchart of the PSO-PID control system

4. DISCUSSION AND RESULTS

4.1 The Simulation of Coefficients readuction in PID Controller by using algorithm

In order to survey c oefficients reduction in PID controller the setting has been simulated by PSO algorithm in Simulink tool box in MATLAB software that diagrams block of this simulation presented in fig 5.

The close-loop responsible of designed models has been summarily presented based on table 3.

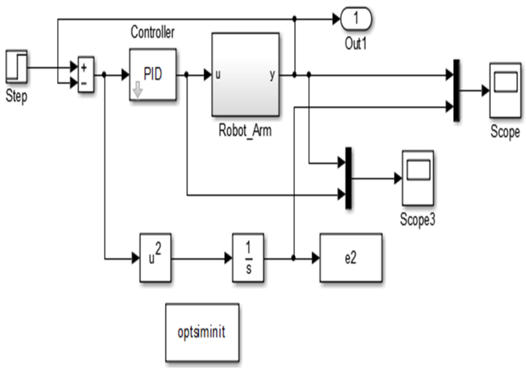

Fig .7. Diagram block of simulating PID controller by using PSO algorithm

In order to investigate the effects of primary population swarm and the numbers of birds scale, several modes have been chosen (primary population = 30, 35, 40, 45, 50, the number of scale = 40, 35, 30, 45 and 50) and simulating of PSO algorithm has been implemented in order to reduce PID coefficients and the results were shown below. The results of each one of investigated models has been presented in table 4 like the ones obtained in figs 3, 4, and 5.

Table 3: The surveyed models based on number and scale of birds

Model	The number of birds	The number of birds scale
First	30	50
Second	35	50
Third	40	50
Fourth	45	50
Fifth	50	50
Sixth	30	30
Seventh	30	35
Eighth	30	40

Table 4: Comparing the obtained results of all operated models

Model	Parameters	Values	Model	Parameters	Values
One	K_D	0/6281	Six	K_D	0/1491
	K_P	0/2839		K_P	0/0741
Two	K_D	0/2392	Seven	K_D	0/2119
	K_P	0/5184		K_P	0/2755
Three	K_D	0/5606	Eight	K_D	0/2306
	K_P	0/2893		K_P	0/0082
Four	K_D	0/5943	Nine	K_D	0/4387
	K_P	0/2916		K_P	0/3009
Five	K_D	0/6126			
	K_P	0/3483			

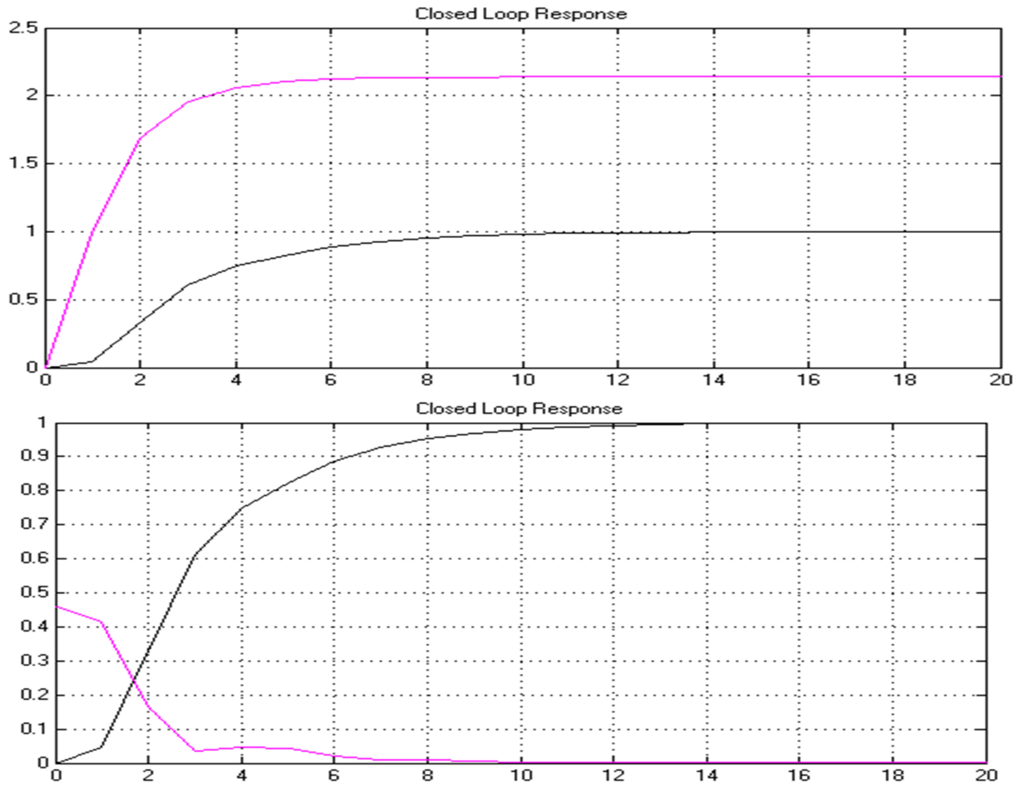

Fig.8. (A) The response of close-loop PID controller in sixth model (B) Comparison of system implementation output and input

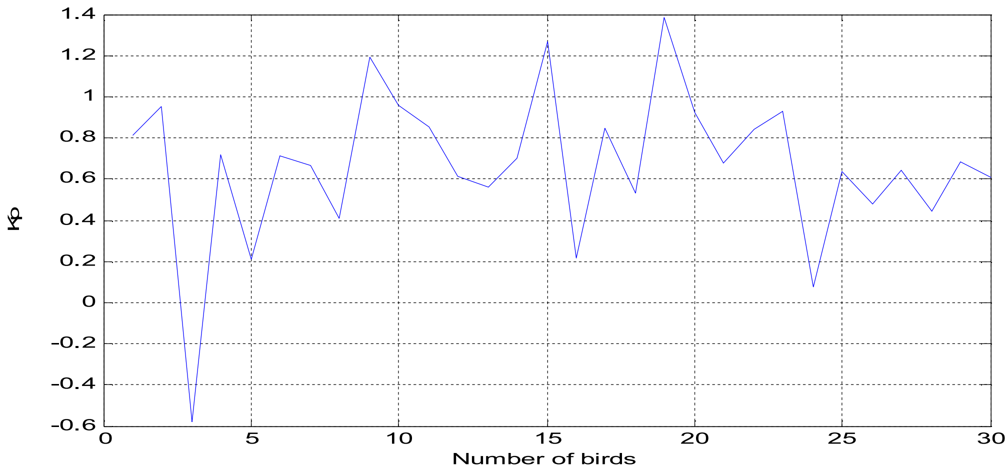

Fig .9. The rate of in term of the numbers of population swarm in sixth model

Fig .10. The rate of K_D in term of the numbers of population swarm in sixth model

5. CONCLUSION

In the present study, it is tried to reduce PID controller coefficient by using PSO algorithm. The result of simulation showed that particle swarm optimization is able to reduce these coefficients and it has been shown accordance with presented figures that the functions of Mp and Tr are minimized by using PSO algorithm method as the best observed ratio of K_p , K_d and K_i respectively, equal to 0.0082, 0.1491, 0. It should be noted that these values have been obtained in different modes and models. Therefore the highest model is sixth model that has simultaneously been obtained by the least ratio of K_p , K_d and K_i equal to 0.0741, 0.1491 and 0. In future studies we can use other metaheuristic algorithms for PID controller coefficient such as genetic algorithm, genetic programming, Ant colony optimization, bee colony, colonial competition algorithm, Intelligent Water Drops algorithm, and also Tabu search.

REFERENCES

[1] Betin F., Pinchon D., and Capolino G. (2000). “Fuzzy logic applied to speed control of a stepping motor drive,” IEEE Transactions on Industrial Electronics, vol. 47, pp.610-622.

[2] Qingding G. and Yanna S. (2000). “H control based on internal model theory for linear permanent magnet synchronous servo motor (LPSM),” Control Theory & Applications, vol.17, pp. 509-512.3

[3] Satyobroto Talukder (2011). Mathematical Modelling and Applications of Particle Swarm Optimization, Submitted to the School of Engineering at Blekinge Institute of Technology In partial fulfillment of the requirements , Department of Mathematics and Science.

[4] Betin F., Pinchon D., and Capolino G. (2000). “Fuzzy logic applied to speed control of a stepping motor drive,” IEEE Transactions on Industrial Electronics, vol.47, pp.610-622.

[5] Systematic Design Approach for a New Series of Ultra-NEMA Premium Copper Rotor Motors, by Fuchsloch, J. and E.F. Brush (2007), in EEMODS 2007 Conference Proceedings, 10-15 June,Beijing.

[6] Kalman, R.E. (1960). “A New Approach to Linear Filtering and PredictionProblems,”

Transactions of the ASME--Journal of Basic Engineering, vol.82,pp. 35-45.

[7] Stengel R., "Optimal Control and Estimation: First Edition," Dover Publication, Inc., New York.

[8] Behal A., Feemster M., Dawson D., and Mangal A. (2000). "Sensorless Rotor Velocity Tracking Control of the Permanent Magnet Stepper Motor," Proceedings of the IEEE International conference on Control Applications, pp. 150-154, Alaska, September.

[9] Simon D. (2000). "Design and rule base reduction of a fuzzy filter for the estimation of motor currents," International Journal of Approximate Reasoning, pp. 145-167.

[10] Crnosija P., Kuzmanvoic B., and Ajdukovic S. (2000). "Microcomputer implementation of optimal algorithms for closed-loop control of hybrid stepper motor drives," IEEE Transactions on Industrial Electronics, vol. 47, pp. 1319-1325.

[11] A.Nabae, S. Ogasawara, and H. Akagi, (1986) "A Novel Control Scheme for Current-Controlled PWM Inverters," IEEE Transactions on Industry Applications, vol. IA-22, pp. 697-701, July/Aug.

[12] Araki M., Control Systems, Robotics, and Automation – Vol. II - PID Control -,Kyoto University, Japan.

[13] Gaing Z. L., (2004). A particle swarm optimization approach for optimum design of PID controller in AVR system, IEEE Trans. Energy Conversion, vol. 19, p. 384 –391.

[14] Kim T. H., Maruta I., Sugie T. (2008). Robust PID controller tuning based on the constrained particle swarm optimization, Automatica, Vol. 44, Issue 4, p. 1104 – 1110.

[15] Akbari, M., et al. "Nonlinear H∞ controller for flexible joint robots with using feedback linearization." International Journal on Computer Science and Engineering (IJCSE), ISSN (2011): 0975-3397.

[16] Pourmahmood, Mohammad, Mohammd Esmaeel Akbari, and Amin Mohammadpour. "An efficient modified shuffled frog leaping optimization algorithm." Int. J. Comput. Appl 32.1 (2011): 0975-8887.

[17] Akbari, M. E., M. A. Badamchizadeh, and M. A. Poor. "Implementation of a fuzzy TSK controller for a flexible joint robot." Discrete Dynamics in Nature and Society 2012 (2012).