DC Motor Drive with P, PI, and Particle Swarm Optimization Speed Controllers

Nadia Qasim Mohammed
Electrical Engineering Department
Baghdad University
Baghdad, Iraq

ABSTRACT
This paper implements a Particle Swarm Optimization (PSO) speed controller for controlling the speed of DC motor. Also traditional Proportional (P), Proportional-Integral (PI) controller have been developed and simulated using MATLAB/SIMULINK. The simulation results indicate that the PI controller has big overshoot, sensitivity to controller gains. The optimization with particle swarm verified the reduction of oscillations as well as improve the steady state error, rise time and overshot in speed response. The simulation results confirmed the system is less sensitive to gain.

Keywords
Practical Swarm Optimization (PSO), Proportional (P), Proportional-Integral (PI) controller, DC motor.

1. INTRODUCTION
A highly efficient DC motor drives are commonly used for industrial applications due to their torque speed characteristics that make them suitable for a wide speed control range. A high starting torque with a constant torque region is obtained with a speed control after variation in the load. The torque and speed responses of a DC motor is suitable with various mechanical loads [1]. Practically, the speed controller which implemented in a DC motor is simpler than the AC motor [2, 3].

The tuning of speed controller can be decided by a genetic algorithm (GA) [1]. As well as the gains of a PI controller is adjusted by optimization technique [3]. In several decades, the Proportional-Integral (PI) controllers have been used for dc speed control regarding the design simplicity and good performance. The system is considered the settling time, nonlinear modeling of dc motor which make the control robust [3].

This paper proposes the application of a particle swarm optimization technique for tuning parameters of a PI speed controller. The Particle Swarm Optimization (PSO) technique is recently applied in a few fields emerging because it's a powerful optimization tool. This paper is organized as follows: Section II is give illustrates of mathematical modeling of DC Motor with a classical PI controller. Section III gives particle swarm optimization. Section IV gives tuning of PI controller based on PSO. The simulation results are discussed in section V.

2. DC MOTOR DRIVE WITH PI CONTROLLER
The PI controller block diagram is shown in Figure (1). In order to make zero steady state error for a step change of speed by adjusting the integral part of the controller.

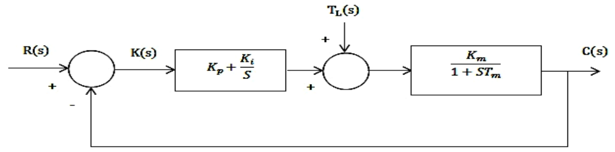

Figure 1. PI controller block diagram

Equation (1) gives transfer function of the control system.

$$\frac{C(S)}{R(S)} = \frac{K_m(SK_p + K_i)}{T_m S^2 + (1 + K_m K_p)S + K_i} \quad (1)$$

Where

K_i is the integral gain of PI controller,

K_p, is the and the proportional gain of PI controller,

T_m, is the mechanical time constant of motor, and

K_m, is the motor gain constant.

Whilst, Equation (2) gives the transfer function between output of the control system and the $T_L(S)$ is a load torque disturbance.

$$\frac{C(S)}{T_L(S)} = \frac{S(1 + T_m)}{T_m S^2 + (K_m + T_m K_i + K_p)S + K_i} \quad (2)$$

Figure 2 shows the MATLAB Model of P and PI controllers.

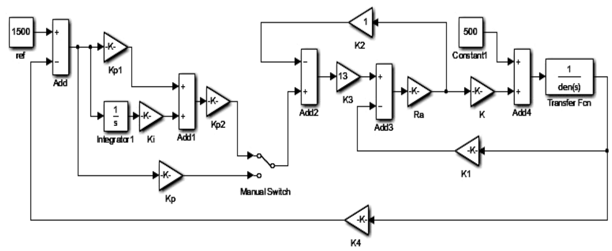

Figure 2. P, and PI controller

Where:

R_a = armature resistance (Ω).

L_a = self-inductance of armature (H).

J = moment of inertia (kg.m^2).

B = friction coefficient (Nm*s/rad).

The DC motor parameters are given in Table I.

Table I Parameter values of DC motor

Specification of DC motor	R_a = 1.818Ω	L_a = 0.5H	J = 0.0465 kg.m^2	B = 0.004Nm*s/rad

3. PARTICAL SWARM OPTIMIZATION (PSO)

The PSO is regarded to be vigorous in solving nonlinear optimization problems [4]. The stochastic optimization based on particle swarm is biologically as well as, it modified from a family of evolutionary computation [5].

In references [6] and [7] the improved a PSO algorithm based on the conduct of each particle of a swarm has been given. To increase efficiency of the group used this algorithm to share information among members within a group In comparison of PSO to the GA, it can be obtained that PSO has a low computational time and it gives a good performance that's because of its simplicity and in addition to these features, the realization in digital controllers and stability [4]. Moreover, the standard and improved PSO are illustrated in Ref. [4].

In order to select the PSO factors, the velocity v and weight factor w□□are concerned to be able of escaping from the local optimization and reach the goal (global optimization) [8]. The controller performance is evaluated from various control qualities that will be shown in following sections.

That's objective function of efficiently will be search the solution space by PSO [10, 11].

In case of a multidimensional problem, the velocity $v_i(n+1)$ of the next particle as well as the position $x_i(n+1)$ this equations are using for update of each particle in the swarm:

$$v_i(n+1) = \omega . v_i(n) + c_1 .rand.(pbest(n) - x_i(n)) +$$
$$c_2 .rand.(gbest(n) - x_i(n)) \ldots\ldots\ldots\ldots\ldots\ldots\ldots(3)$$
$$x_i(n+1) = x_i + v_i(n+1) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

where

$v_i(n+1)$ is the velocity of the ith particle at $(n+1)$ iteration,

$x_i(n+1)$ is the position of the ith particle at $(n+1)$ iteration,

ω is the weighting function,

c_1 is the cognitive acceleration constants learning rate

c_2 is the acceleration constants social learning rate,

The random function in the range [0, 1], while the each particle has best position is *pbest*, finally the global best position of the individuals is the *gbest*.

The *gbest* version implement the best position in terms of number of repetitions to converge. While, the most resistant to local minima is still two in the *pbest* version with neighborhoods.

For adjusting dynamically the velocity $v_i(n+1)$ of the particles the weight factor *w* is responsible, local and global search indicates the responsibility limits between these searches. The PSO is decay the inertia weight form the large value to small value when the start of the algorithm, thereafter execution process makes the algorithm at the beginning search globally and at the end of the execution of the algorithm search locally.

In other words, the weight factor *w* will affect the repetition number in order to find the optimal solution. The convergence will be fast when the value of weight factor *w* is low, but the solution will fall into the local minimum. In addition to increasing of the repetition number regarding the increasing of the value of weight factor *w*. After that the convergence will be slow. When the PSO algorithm is running to adjust the value of weight factor *w* in the training process.

The weighting function *w* is calculated as:

$$\omega = \omega_{\max} - \frac{(\omega_{\max} - \omega_{\min}).iter}{iter_{\max}} \qquad (5)$$

where,

$\omega_{\max}$ and $\omega_{\min}$ are the initial and final weights

iter is the current iteration time

$iter_{\max}$ is the maximum number of iterations.

To find the fitness function F(s) of the optimization of parameters of PID controller as [5]:

$F(s) = \omega_{\max} (M p + ISE) + \omega_{\min} (T p + T s)$ (6)

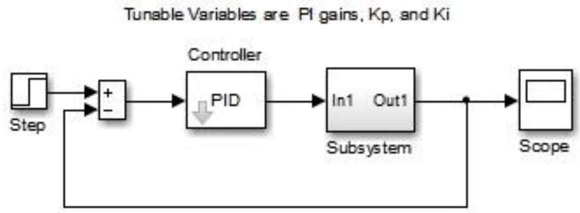

Figure 3 PSO controller

Although the P, and PI are more conventional for optimization aims, to obtain more accuracy and more optimal solutions of fitness function using PSO.

In result, a PSO algorithm is further enhanced with using a time lessening weight factor *w*, which leads to a reduction in the number of repetitions [11].

Equation (3) has two terms as,

$c_1.rand.(pbest(n) - x_i(n))$ this term represents the individual movement and,

$c_2.rand.(gbest(n) - x_i(n))$ this term represents the social behavior in finding the global best solution.

4. TUNING OF PI CONTROLLER BASED ON PSO

The values of the three parameters (K_p, K_i and K_d) must be adjusted. So that, the control input will provide possible accomplishment. These parameters have been included in a chromosome as illustrated in Figure (4). There are several controller design methods are implemented to get an acceptable results. The response with classical control methods needs retuning by the designer but these methods provide initial approximation.

K_p	K_i	K_d

Figure 4 Chromosome structure

Fitness Function F_{obj}:

It is important to be accurately specified. In this paper, the fitness function (F_{obj}) is defined as follows:

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

where,

t_r is Rise time;

t_s is Settling time;

M_P is Overshoot;

Ess is the steady state error.

The PID controller parameters could be evaluated approximately using conventional tuning method such as Ziegler-Nichols experimental method [6].

5. SIMULATION RESULTS

The simulation results of the input is unit step response and transfer function of DC motor using P, PI controllers, and PSO controller their performance parameters are described and compared. However, without controller, the DC motor in this case has a slow step response.

As shown in Figures (5)-(12) by using P, PI controllers and PSO controller, with two values of speed 1500 r.p.m and 2000 r.p.m under different load torque values (1.2 Nm and 1.9 Nm).

It can be illustrated from the figures that the improvement of the response under different dynamic operations. There are different cased have been considered to verified the proposed method (PSO). These cases are shown for speed responses under various dynamic operations (the load torque (T_L) are 1.2Nm and 1.9)

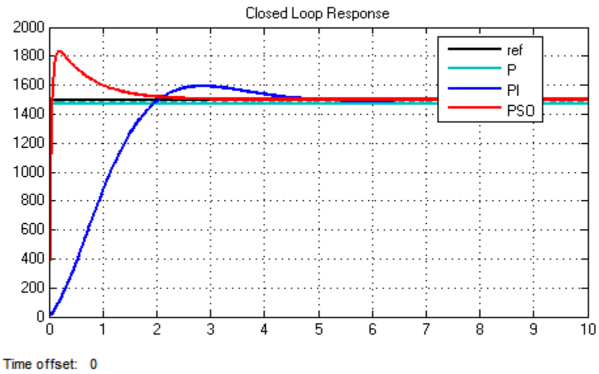

Figure 5 speed reference at 1500rpm, 5000KHz, T_L= 1.2Nm at 0.12sec

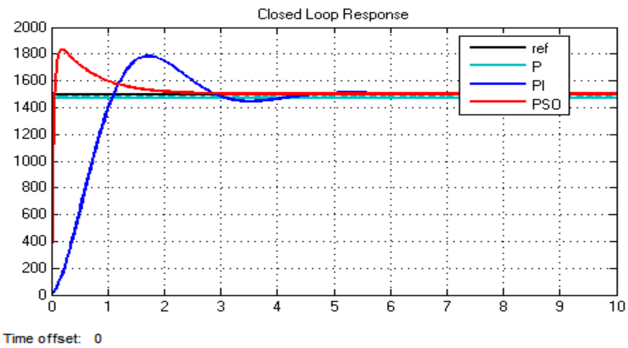

Figure 6 speed reference at 1500rpm, 10000KHz, T_L= 1.2Nm at 0.12sec

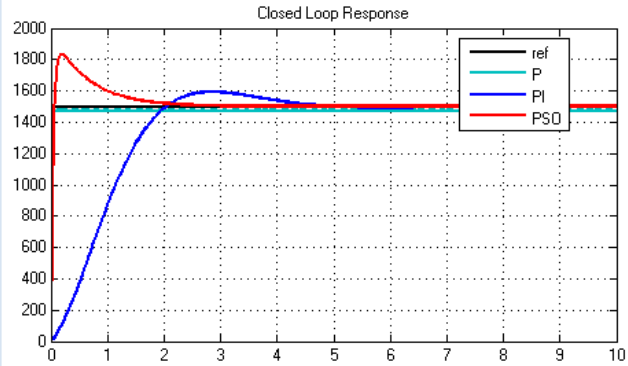

Figure 7 speed reference at 1500rpm, 5000KHz, T_L= 1.9Nm at 0.9sec

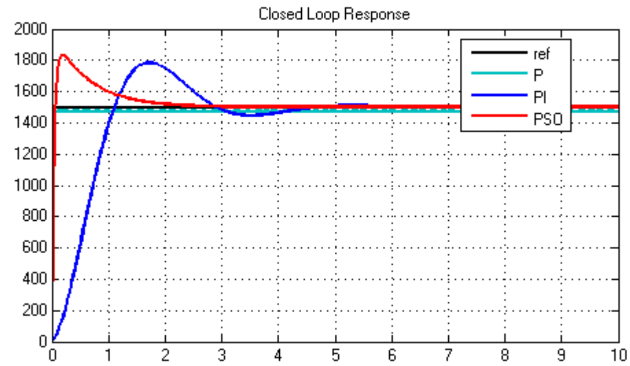

Figure 8 speed reference at 1500rpm, 10000 KHz, T_L= 1.2Nm at 0.12sec

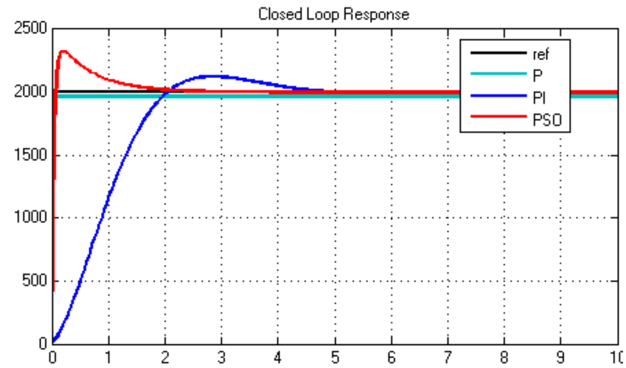

Figure 9 speed reference at 2000rpm, 5000 KHz, T_L= 1.2Nm at 0.12sec

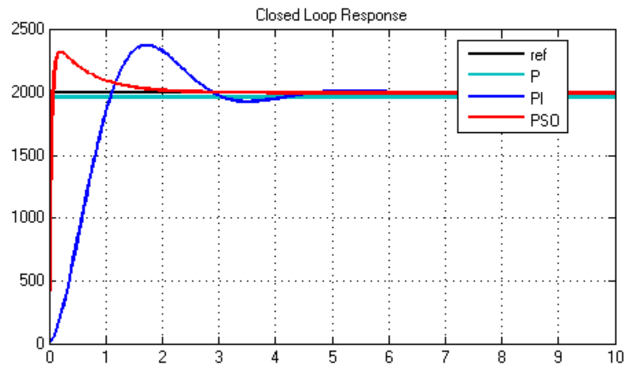

Figure 10 speed reference at 2000rpm, 10000 KHz, T_L= 1.2Nm at 0.12sec

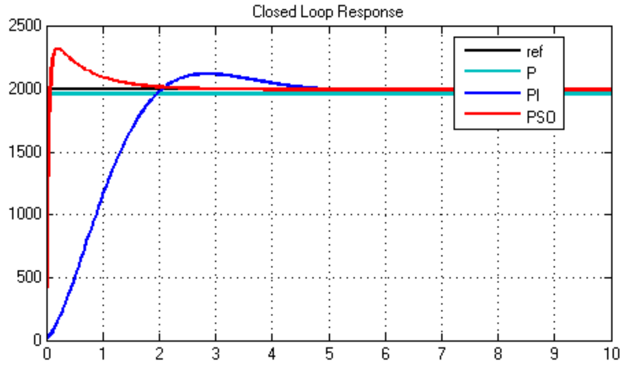

Figure 11 speed reference at 200rpm, 5000 KHz, T_L= 1.9Nm at 0.32sec

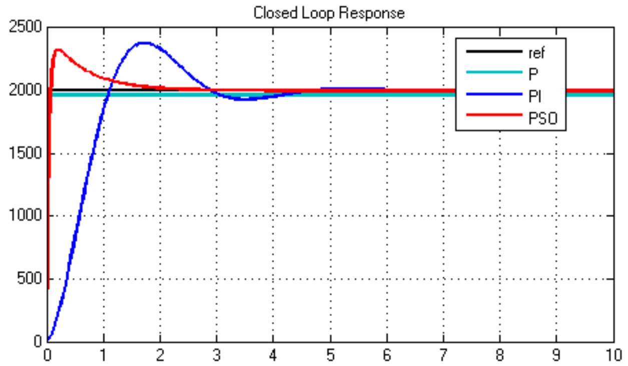

Figure 12 speed reference at 200rpm, 10000 KHz, T_L= 1.2Nm at 0.12sec

When running of PSO algorithm for different combination of c_1, c_2 and w that give the optimal speed response as shown in Table 2.

Table 2 PSO parameters values

Parameter	Values	Values	Values
Number of Particles	10	10	10
Maximum no. of Iterations	20	20	20
c_2	1.2	1.9	1.2
c_1	0.2	0.32	0.12
Ω	0.9	0.9	1.5

From simulation results, it was observed that under different values of speed the PI controller taken a long rise time while the PSO controller performed well in the case of sufficiently large reference input changes regarding a short settling time. It can be revealed also that the delay time is decreased in PSO controller with different dynamic operations.

6. CONCLUSIONS

In this paper, the P, and PI controller has been designed and optimized the parameters of the speed controller by a Particle Swarm Optimization (PSO) technique. The proposed PSO has a good accuracy and divergence speed comparing with based method of P, and PI speed controllers according to obtained evaluation results. In addition, design of PID controller using PSO is caused that the rate of rise time, delay time, and settling time in step response curve is reduced in comparison with P, and PI. It should be noted that PSO performance in design and optimization process can be more improved by increasing the number of iterations.

7. REFERENCES

[1] Santosh Kumar Suman, Vinod Kumar Giri, "Speed control of DC motor using optimization techniques based PID Controller", IEEE International Conference on Engineering and Technology (ICETECH) , Pages: 581 – 587, 2016

[2] Aziz Ahmed Yogesh Mohan Aasha Chauhan Pradeep Sharma Comparative Study of Speed Control of D.C.Motor Using PI, IP, and Fuzzy Controller", International Journal of Advanced Research in Computer and Communication EngineeringVol. 2, Issue 7, July 2013.

[3] Adel A. A. El-Gammal1, Adel A. El-Samahy," A Modified Design of PID Controller For DC Motor Drives Using Particle Swarm Optimization PSO", International Conference on Power Engineering, Energy and Electrical Drives, POWERENG '09.18-20 March 2009

[4] Jun-Yi Cao and Bing-Gang Cao, "Design of fractional order controller based on particle swarm optimization,"International Journal of Control, Automation and Systems, vol. 4, no. 6, 2006, pp. 775-781.

[5] Majid Zamani, Masoud Karimi-Ghartemani, Naseer Sadati, "FOPID controller design for robust performance using particle swarm optimization,"Fractional Calculus and Applied Analysis, vol. 10, no. 2, 2007, pp. 169-187.

[6] J. Kennedy, R. Eberhart, Y. Shi, "Swarm Intelligence", Morgan Kaufmann Publishers, 1st Edition, San Francisco, pp. 80-95, 2001.

[7] 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.

[8] Mohammad Reza Dastranj, Mojtdaba Rouhani, and Ahmad Hajipoor, "Design of Optimal Fractional Order PID Controller Using PSO Algorithm,"International Journal of Computer Theory and Engineering, Vol. 4, No. 3, June 2012.

[9] J.G Zigeler, N.B. Nichols, "Optimization Setting for Automatic Controller", Trans. ASME, Vol. 64, pp. 756-769, 1942.

[10] H. Shayeghi, A. Jalili, H.A. Shayanfar, "Multistage Fuzzy Load Frequency Control Using PSO", Energy Convers. Manage., Vol. 49, pp. 2570-2580, 2008.

[11] 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.