

PSO-BASED TUNNING OF PID CONTROLLER FOR SPEED CONTROL OF DC MOTOR

Ibrahim S. Fatah

Assistant Lecturer, Collage of Engineering, University of Diyala.

E-mail: ibrahems@yahoo.com

(Received: 12/6/2013; Accepted 6/10/2013)

ABSTRACT: - In this paper, a Proportional-Integral-Derivative (PID) controller of DC motor is designed by using particle swarm optimization (PSO) strategy for formative optimal PID controller tuning parameters. The proposed approach has superior feature, including easy implementation, stable convergence characteristics and very good computational performances efficiency. The DC Motor Scheduling PID-PSO controller is modeled in MATLAB environment. Comparing with conventional PID controller using Genetic Algorithm, the planned method is more proficient in improving the speed loop response stability, the steady state error is reduced, the rising time is perfected and the change of the required input do not affect the performances of driving motor with no overtaking.

Keyword: Control Tuning Parameters; Particle Swarm Optimization Strategy; PID-PSO Controller; Speed Control; Optimal Control.

1- INTRODUCTION

PID controllers are widely used in industrial plants because it is simple and robust. Industrial processes are subjected to variation in parameters and parameter perturbations, which when significant makes the system unstable. So the control engineers are on look for automatic tuning procedures.

From the control point of view, DC motor exhibit excellent control characteristics because of the decoupled nature of the field. Recently, many modern control methodologies such as nonlinear control, optimal control, variable structure control and adaptive control have been extensively proposed for DC motor. However, these approaches are either complex in theoretical bases or difficult to implement. PID control with its three term functionality covering treatment to both transient and steady-states response, offers the simplest and yet most efficient solution too many real world control problems. In spite of the simple structure

and robustness of this method, optimally tuning gains of PID controllers have been quite difficult.

The particle Swarm Optimization (PSO) methods have been employed successfully to solve complex optimization problems. PSO first introduced by Kennedy and Eberhart is one of the modern heuristic algorithms; it has been motivated by the behavior of organisms, such as fish schooling and bird flocking. Generally, PSO is characterized as a simple concept, easy to implement, and computationally efficient. Unlike the other heuristic techniques, PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities.

In this paper, a scheduling PID tuning parameters using particle swarm optimization strategy for a DC motor speed control is proposed. This paper has been organized as follows: in section 2 the mathematical model DC motor is described. In section 3, the PID controller is described. Section 4 the fitness function, section 5 the particle swarm optimization method is reviewed. A comparison between the results obtained by the proposed method and GA method via simulation the DC motor speed control is presented in section 6. The paper is concluded in section 7.

2. MATHEMATICAL MODELING OF DC MOTOR

The permanent magnetic DC motor equivalent model is shown in Fig. 1. The motor-load system consists of a DC motor coupling with load. Fig.1 shows schematic diagram of a DC motor. For constant field, the armature voltage needs to be considered. We now consider the dynamic model of DC motor with load given by the following differential equations:

$$V_a(t) = R_a \cdot I_a(t) + L_a \cdot \frac{di_a}{dt} + E_b(t) \quad \dots\dots\dots (1)$$

$$E_{b(t)} = K_b \cdot w(t) \quad \dots\dots\dots (2)$$

$$T_m(t) = K_T \cdot I_a(t) \quad \dots\dots\dots (3)$$

$$T_m(t) = J_m \cdot \frac{dw(t)}{dt} + B_m \cdot w(t) \quad \dots\dots\dots (4)$$

Where V_a is armature voltage (v), R_a is armature resistance (Ω), L_a is armature inductance (H), I_a is armature current (A), E_b is back emf (V), $w(t)$ is angular speed (rad/s), T_m is motor torque (Nm), J_m is rotor inertia (Kgm^2), B_m is viscous friction coefficient (Nms/rad), K_T is torque constant(Nm/A) and K_b is back emf constant (Vs/rad).

The block diagram of the DC motor is shown in Fig.2. The transfer functions between shaft speed and armature voltage at no-load is:

$$\frac{w(s)}{V_a(s)} = \frac{K_T}{L_a J_m s^2 + (R_a J_m + L_a B_m) s + (K_T K_b + R_a B_m)} \dots\dots\dots (5)$$

The model built in Simulink of the DC motor is shown in Fig. (3). Motor model was converted to a 2-input 1-output subsystem. Input ports are armature voltage (V_a) and load torque (T_{load}) and the output ports are angular speed in (w).

For applications where the load is to be rapidly accelerated or decelerated frequently, the electrical and mechanical time constants of the motor plays an important role. The mechanical time constants in these motors are reduced by reducing the rotor inertia. [1]

3. PID CONTROLLER

The time-domain PID controller is

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \dots\dots\dots (6)$$

where K_p is the proportional gain, K_i is the integral gain, and K_d is the derivative gain.

The variable $e(t)$ represents the tracking error, which is the difference between the desired input value $r(t)$ and the $y(t)$ actual output .

A proportional controller K_p has the effect on reducing the rise time, but not for the steady-state error. An integral control K_i has the effect on eliminating the steady-state error, but it may cause poor transient response. A derivative control K_d has the effect on reducing the overshoot, and improving the transient response. To obtain an optimal set (K_p, K_i, K_d) of the PID gains.

A PID controller does not "know" the correct output to bring the system to the set point. It moves the output in the direction which should move the process toward the set point and needs to have feedback (measurements) to perform.

4. FITNESS FUNCTION

In PID controller design methods, the most common performance criteria are integrated absolute error (IAE), the integrated of time weight square error (ITSE) and integrated of squared error (ISE) that can be evaluated analytically in the frequency domain [2], [3].

These three integral performance criteria in the frequency domain have their own advantage and disadvantages. For example, disadvantage of the IAE and ISE criteria is that its minimization can result in a response with relatively small overshoot but a long settling time because the ISE performance criterion weights all errors equally independent of time. Although the ITSE performance criterion can overcome the disadvantage of the ISE criterion, the derivation processes of the analytical formula are complex and time-consuming. The IAE, ISE, and ITSE performance criterion formulas are as follows:

$$IAE = \int_0^{\infty} |r(t) - y(t)| dt = \int_0^{\infty} |e(t)| dt$$

..... (7)

$$ISE = \int_0^{\infty} e^2(t) dt$$

..... (8)

$$ISTE = \int_0^{\infty} te^2(t) dt$$

..... (9)

In this paper a time domain criterion is used for evaluating the PID controller. A set of good control parameters P, I and D can yield a good step response that will result in performance criteria minimization in the time domain. These performance criteria in the time domain include the overshoot, rise time, settling time, and steady-state error. Therefore, the main objective function that is selected to be minimized will be considered in this study: Minimize J, where:

$$J = (1 - e^{-\beta})(M_p + E_{ss}) + e^{-\beta}(t_s - t_r) \quad \text{..... (10)}$$

Where K is [P, I, D], and β is the weighting factor.

The performance criterion (fitness function) J can satisfy the designer requirement using the weighting factor β value. β can set to be larger than 0.7 to reduce the overshoot and steady states error, also can set smaller than 0.7 to reduce the rise time and settling time. The optimum selection of β depends on the designer's requirement and the characteristics of the plant under control. In BLDC motor speed control system the lower β would lead to more optimum responses. In this paper, due to trials, β is set to 0.5 to optimum the step response of speed control system. [4]

5. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart developed a PSO algorithm based on the behavior of individuals (i.e., particles or agents) of a swarm [5]. Its roots are in zoologist's modeling of

the movement of individuals (i.e., fishes, birds, and insects) within a group. It has been noticed that members of the group seem to share information among them. This communication fact leads to increase efficiency of the group. The PSO algorithm searches in parallel using a group of individuals similar to other AI-based heuristic optimization techniques [6]. Each individual corresponds to a candidate solution to the problem. Individuals in a swarm approach to the optimum through its present velocity, previous experience, and the experience of its neighbours.

Let x and v denote a particle coordinates (position) and its corresponding flight speed (velocity) in a search space, respectively. Therefore, the i 'th particle is represented as $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{id})$ in the d -dimensional space. The best previous position of the i -th particle is recorded and represented as $pbest_i = (pbest_{i1}, pbest_{i2}, pbest_{i3}, \dots, pbest_{id})$. The index of the best particle among all the particles in the group is represented by the $gbest_d$. The velocity for the particle i is represented as $v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{id})$. The modified velocity and position of each particle can be calculated using the current velocity and the distance from $pbest_{id}$ to $gbest_{id}$ as shown in the following formulas:

$$v_{id}^{t+1} = w \cdot v_{id}^t + c_1 \text{rand}(\cdot)(pbest_{id} - x_{id}^t) + c_2 \text{rand}(\cdot)(gbest_d - x_{id}^t) \quad \dots\dots\dots (11)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, \quad i = 1, 2, \dots, n \text{ and } d = 1, 2, \dots, m \quad \dots\dots\dots (12)$$

where n is number of particles in a group; m is number of members in a particle; t is the pointer of iterations (generations); w is inertia weight factor; c_1, c_2 are acceleration constants; $\text{rand}(\cdot)$ is uniform random value in the range $[0,1]$; v_i^t is velocity of particle i at iteration t , $V_d^{min} \leq V_{id}^t \leq V_d^{max}$; x_i^t is current position of particle at iteration.

In the above procedures, the parameter V^{max} determines the resolution, or fitness, with which regions are to be searched between the present position and the target position. In many experiences, V^{max} was often set at 10-20% of the dynamic range of the variable on each dimension.

The constants c_1 and c_2 and represent the weights of the stochastic acceleration terms that pull each particle toward the $pbest$ and $gbest$ positions. Here, the acceleration constants c_1 and c_2 were often set to be 2.0 according to experiences. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run. In general, the inertia weight is set according to the following equation:

$$w = w_{max} - \frac{w_{max} - w_{min}}{\text{iter}_{max}} \times \text{iter} \quad \dots\dots\dots (13)$$

where $iter_{max}$ is the maximum number of iterations and $iter$ is the current number of iterations.[7]

The main steps in the particle swarm optimization and selection process are described as follows:

- (a) Initialize a population of particles with random positions and velocities in d dimensions of the problem space and fly them.
- (b) Evaluate the fitness of each particle in the swarm.
- (c) For every iteration, compare each particle's fitness with its previous best fitness ($pbest$) obtained. If the current value is better than $pbest$, then set $pbest$ equal to the current value and the $pbest$ location equal to the current location in the d -dimensional space.
- (d) Compare $pbest$ of particles with each other and update the swarm global best location with the greatest fitness ($gbest$).
- (e) Change the velocity and position of the particle According to equations (11) and (12) respectively.
- (f) Repeat steps (a) to (e) until convergence is reached based on some desired single or multiple criteria.

Fig. (4) shows the general flow chart of the PSO algorithm.

6. SIMULATION AND RESULTS

The value of K_p, K_i , and K_d corresponding to the best solution obtained is used in the block diagram shown in Fig.2. Even though the design of K_p, K_i , and K_d is carried out assuming that there is no change in reference speed, the values of K_p, K_i , and K_d obtained through PSO are now used to study the system's response when there is a step change in reference speed as a first track. It can be inferred from this figure that settling time and maximum overshoot is well within the stipulated limits.

To control the speed of the DC motor at 1000 rpm which is represented as 1 volt signal, according to the trials, the following PSO parameters are used to verify the performance of the PSO-PID controller parameters:

- Population size: 50;
- $w = 0.6$;
- $c_1 = c_2 = 1.2$;
- Iteration :100;

Comparison of PSO-PID Method with GA Methods to show the effectiveness of the proposed method, a comparison is made with the designed PID controller with GA method. At first method, the PID controller is designed using GA method is used to tune the PID controller. The following GA parameters which are used to verify the performance of the GA-PID controller parameters:

- Population size: 100
- Crossover rate: 0.9
- Mutation rate: 0.005
- Number of iterations :100

Fig. (5) shows the convergence graph in the GA method, Fig.6 shown the change of the PID controller gains according to GA method.

Fig. (7) shows the convergence graph in the PSO method, Fig. (8) shown the change of the PID controller gains according to PSO method.

The Step response DC motor in GA based PID speed controller is shown fig. (9). Fig. (10) shown the response of the system in case of change the required input where the input signal is change to the new value at (5 sec), after that retune to the first value.

The Step response DC motor in PSO based PID speed controller is shown fig. (11). Fig. (12) has shown the response of the system in case of change the required input where the input signal is change to the new value at (5 sec), after that retune to the first value.

The values of designed PID Controller are mounted in the table I.

7- CONCLUSION

This paper has explained application of Particle Swarm Optimization PSO for the controller design of a separately excited DC motor. The main objective functions to be minimized, that will be studied separately, are: Minimize the steady state error, minimize the rise time, minimize the maximum overshoot, minimize the settling time, and minimize the ISE which is represented the cost function of the system. The Computer simulation results show that an optimized speed response is always obtained with change reference input speed to double, as well demonstrate that the excellent performance of the optimal PID controller. The size of the population of the GA is greater than the size of the particles swarm that are used with PSO method, though the results that get from the PSO are better than the GA method.

In this paper a new design method to determine PID controller parameters using the PSO method is presented. Obtained through simulation of DC motor, the results show that the proposed controller can perform an efficient search for the optimal PID controller gains. By

comparison with GA methods, it shows that this method can improve the dynamic performance of the system in a better way.

8- REFERENCES

1. Nader Jamali Sufi Amlashi, "Design and Implementation of Fuzzy Position Control System for Tracking Applications and Performance Comparison with Conventional PID", IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 1, No. 1, pp. 31-44, March 2012
2. R. A. Krohling and J. P. Rey, "Design of optimal disturbance rejection PID controllers using genetic algorithm," IEEE Trans. Evol. Comput, vol. 5, pp. 78–82, Feb. 2001.
3. Y. Mitsukura, T. Yamamoto, and M. Kaneda, "A design of self-tuning PID controllers using a genetic algorithm" in Proc. Amer. Contr. Conf., San Diego, CA, pp. 1361–1365, June 1999.
4. Z.-L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system" IEEE Trans. Energy Conversion, vol. 19, pp. 384-391, June 2004.
5. J. Kennedy and R. Eberhart, "Particle swarm optimization," Proc IEEE Int Conf on Neural Networks, pp. 1942-1948, 1995.
6. J. Kennedy and R. Eberhart, Swarm intelligence, San Francisco, CA: Morgan Kaufmann Publishers, 2001.
7. Poomyos Payakkawan, Kitdakorn Klomkarn and Pitikhate Sooraksa, "Dual-line PID Controller based on PSO for Speed Control DC Motor" IEEE, ISCIT, pp.134-139, 2009.

Table I: The values of designed PID Controller.

	PSO-PID	GA-PID	Units
Kp	4.6132	4.1569	
Ki	0.8654	0.78431	
Kd	-0.8773	0	
Tr	8.7	8.0952	msec
Ts	40.7	57.478	msec
Mp%	3.4223	20.2875	
PeakTime	20	20	msec
Ess	0	0	
ISE	0.40039	0.43086	
Computation Time	11.875	26.206	sec

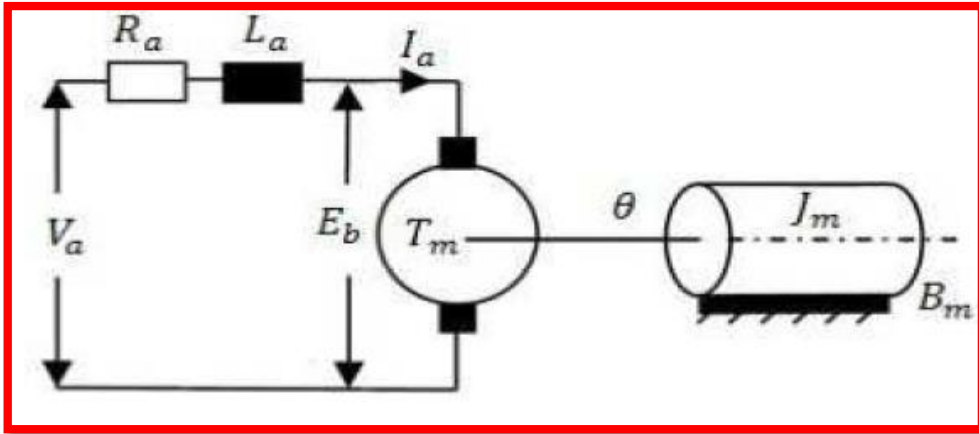


Fig. (1): Separately excited DC motor model.

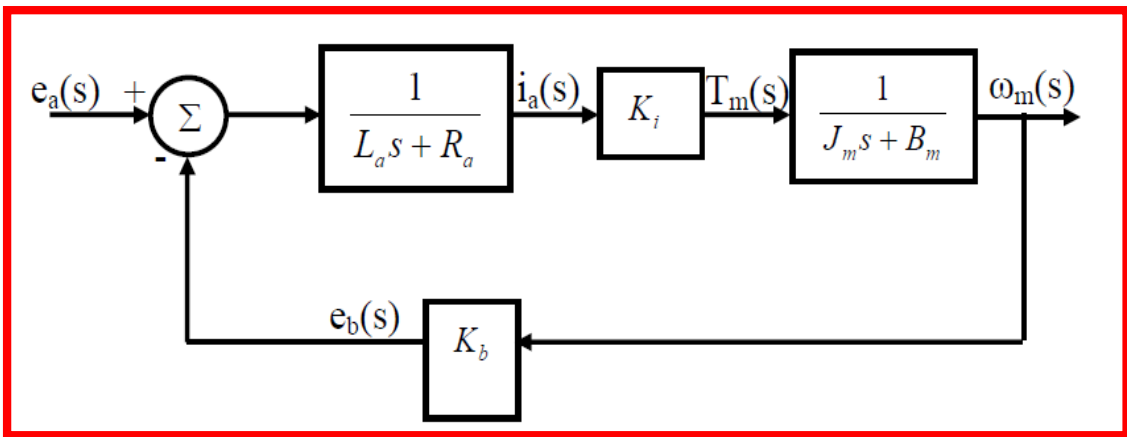


Fig. (2): Block Diagram of armature controlled DC motor.

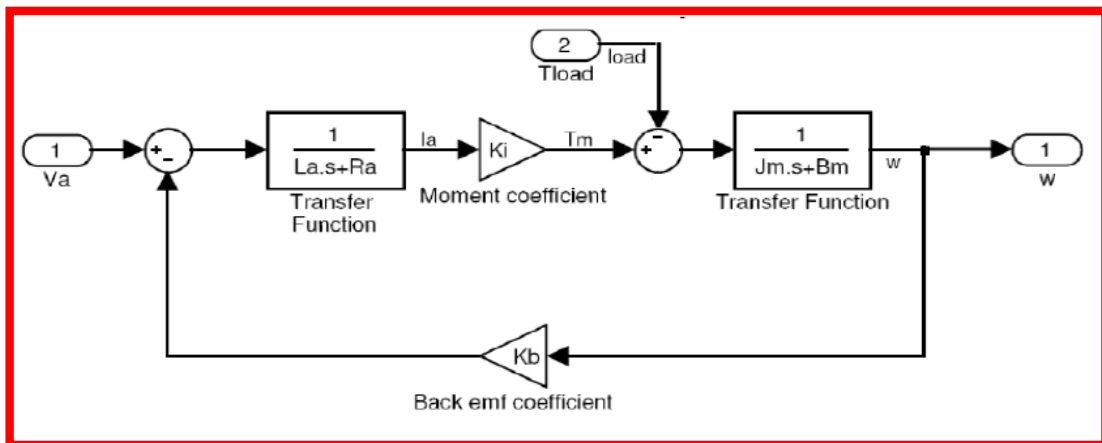


Fig. (3): Simulink DC motor model.

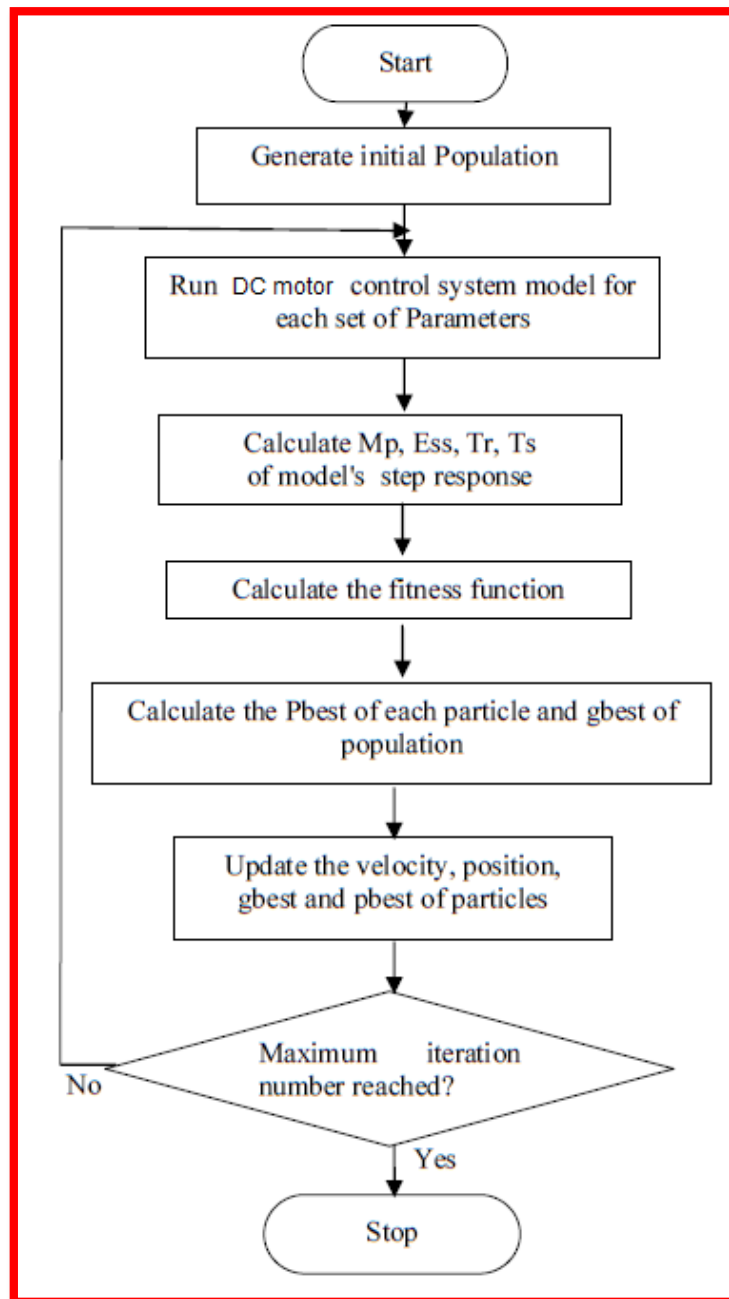


Fig. (4): The general flow chart of the PSO algorithm.

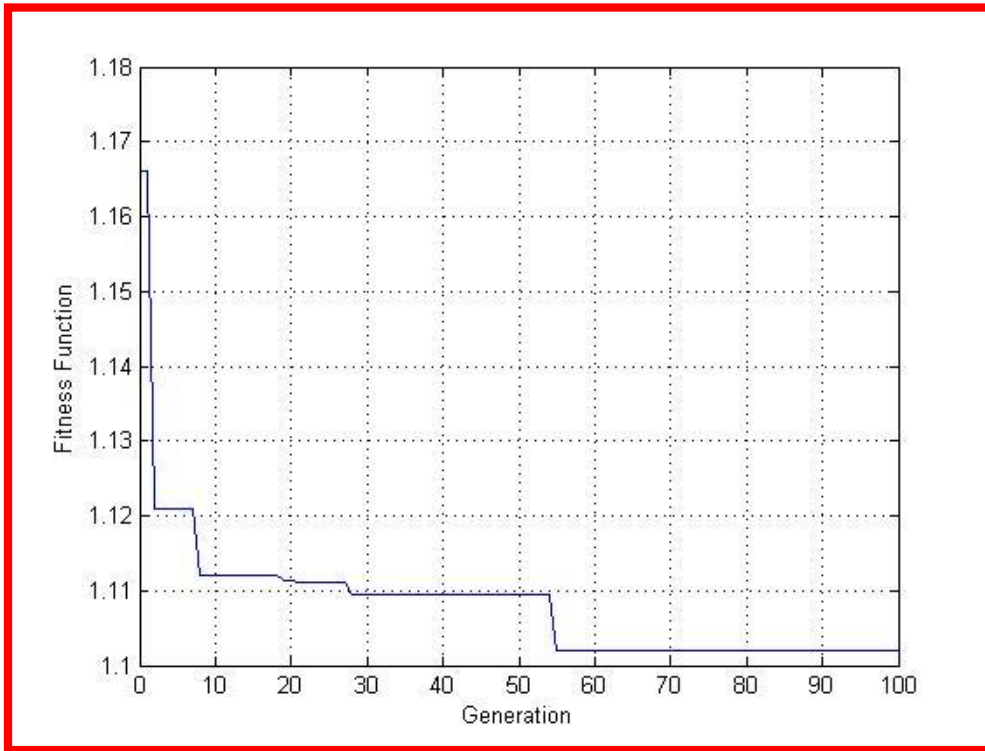


Fig. (5): Convergence graph in the GA based PID speed controller.

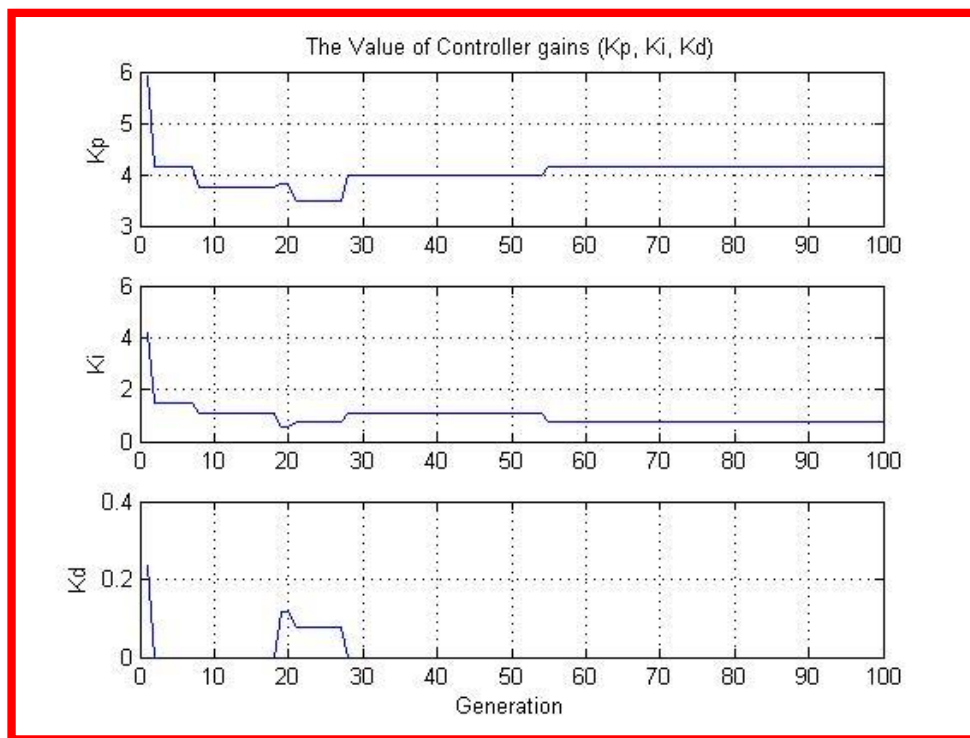


Fig. (6): Convergence graph of the PID gains value in the GA method.

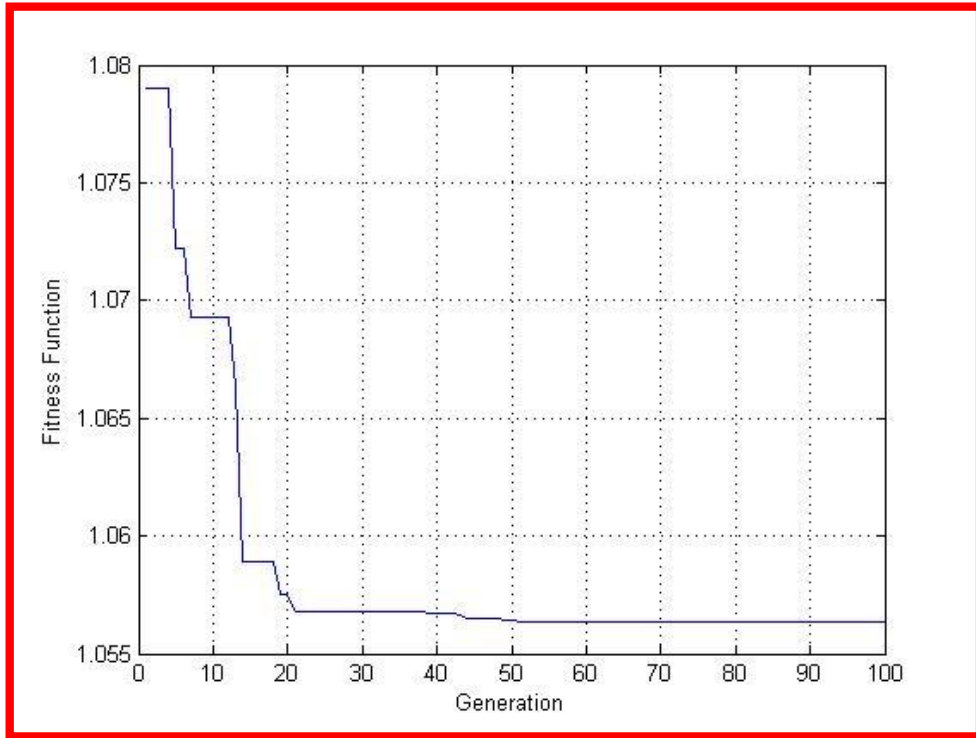


Fig. (7): Fitness function Convergence graph in the PSO based PID speed controller.

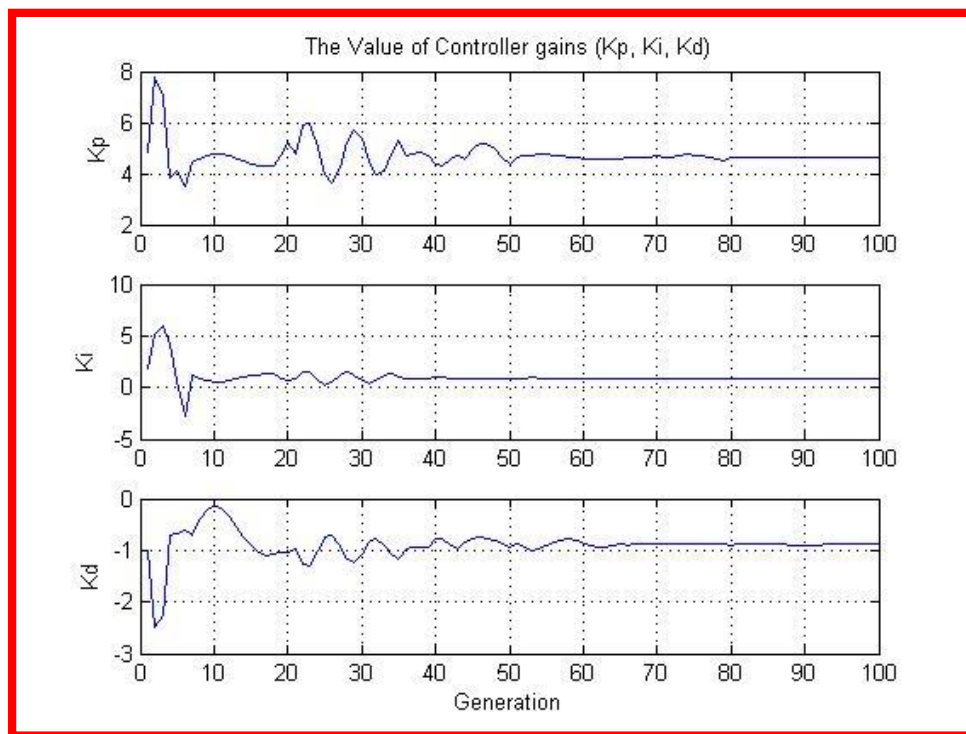


Fig. (8): Convergence graph of the PID gains value in the PSO method.

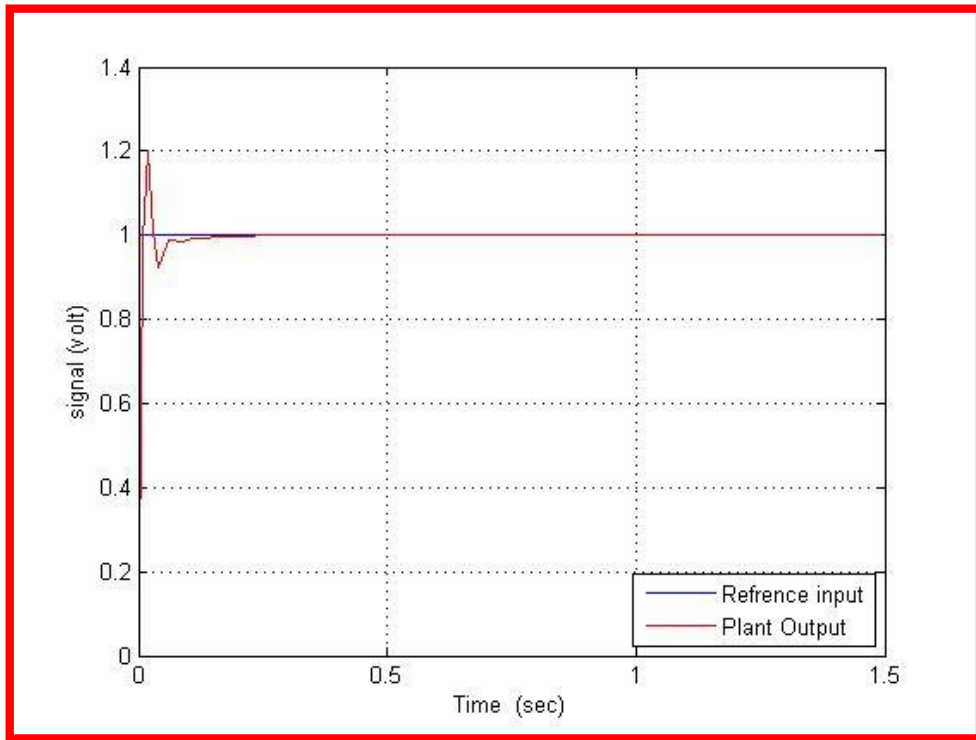


Fig. (9): Step response DC motor in GA based PID speed controller.

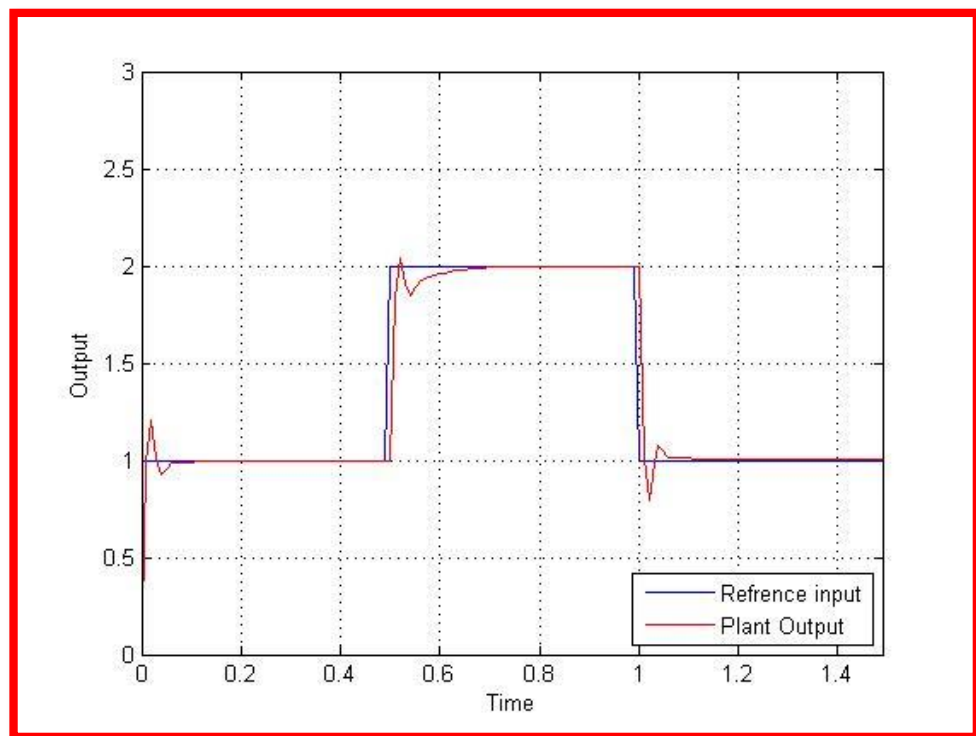


Fig. (10): Change the reference input response of the system by using the GA-PID.

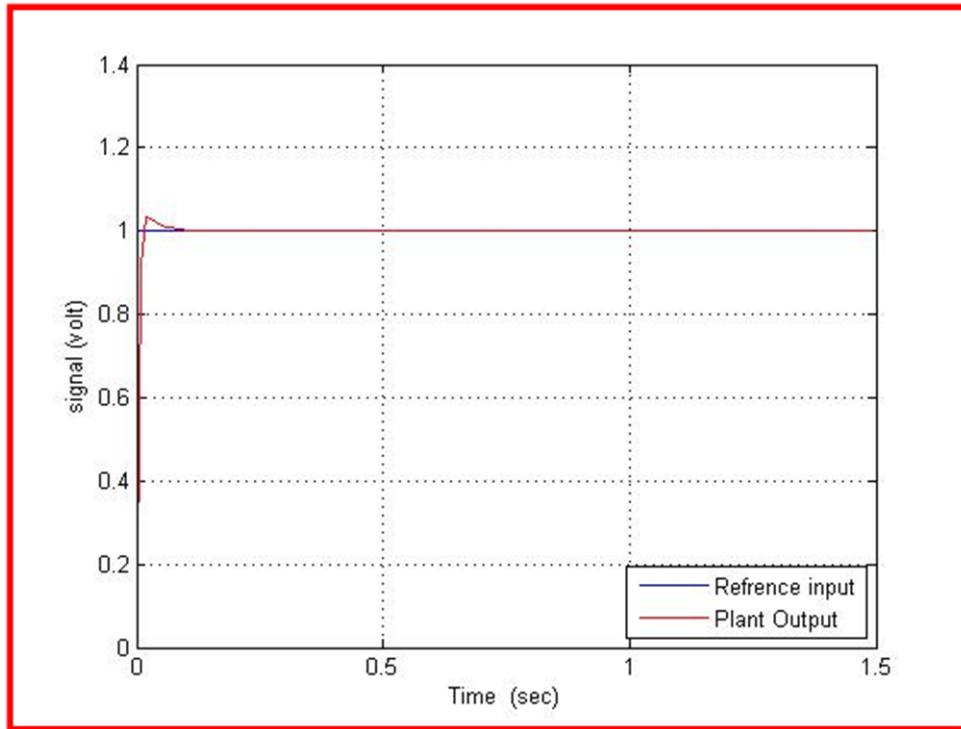


Fig. (11): Step response DC motor in PSO based PID speed controller.

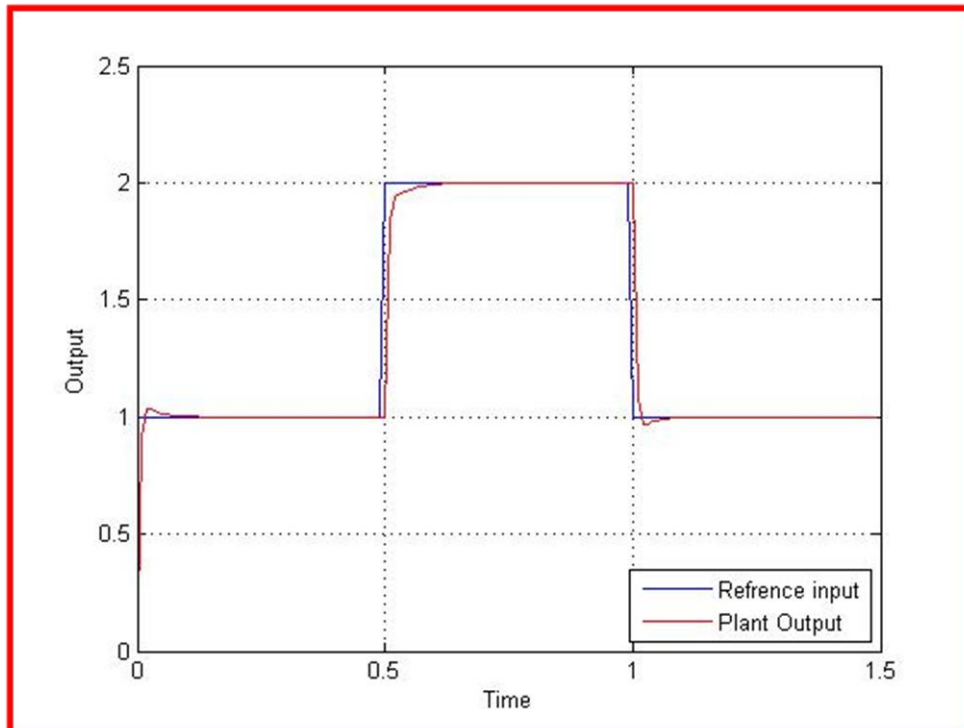


Fig. (12): change the reference input response of the system by using the GA-PID.

تنعيم لمسيطر نوع PID مستند الى خوارزمية PSO للسيطرة على سرعة محرك DC

ابراهيم سعدون فتاح

قسم الالكترونىك / كلية الهندسة / جامعة ديالى

الخلاصة:

في هذا البحث تم تصميم مسيطر نوع (تناسب-تكامل-مشتق) (PID) لمحرك تيار مستمر (DC) باستخدام خوارزمية اسراب الجسيمات لتنعيم قيم المسيطر بافضل قيم. ان الطريقة المقترحة هي صورة متفوقة سهلة التنفيذ وخواص اقتراب مستقرة وكفاءة للحسابات جيدة جدا. قيم المسيطر (PID-PSO) مجدولة باستخدام بيئة MATLAB. تمت المقارنة مع المسيطر الاعتيادي باستخدام الخوارزمية الجينية الطريقة المقترحة كانت اكثر براعة في تحسين استقرار استجابة حلقة السرعة، تقليل خطأ الحالة النهائية، وقت الصعود اكمل بالاضافة الى ان تغيير الدخل المطلوب لا يؤثر على سلوك سواقة المحرك.