14th International Conference on
*AEROSPACE SCIENCES & AVIATION TECHNOLOGY*,
*ASAT - 14* – May 24 - 26, 2011, Email: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel: +(202) 24025292 –24036138, Fax: +(202) 22621908

# Optimal Position Control of Permanent Magnet DC Motor (PMDC)

T.S. Abd El Ghaffar[*], A.S. El-Wakeel[*], A. Eliwa[*], R.M. Mostafa[†]

**Abstract:** This paper presents a comparative study of position control for Permanent Magnet DC motor (PMDC). A comparison between the Proportional-Integral-Derivative (PID) controller tuned using Particle Swarm Optimization technique (PID-PSO), and using Genetic Algorithm technique (PID-GA) has been introduced. The proposed strategy is intended to improve the performance of conventional PID controller by using different optimization techniques and selecting the best one. The MATLAB/SIMULINK toolbox is used to model and simulate the PMDC Motor accurately, since the model of modest accuracy cannot be expected to give fair comparison of different controllers. The performance of both controllers is evaluated to explain the main characteristics of each one.

**Keywords:** Permanent magnet DC Motor (PMDC), Proportional-Integral-Derivative (PID) controller, Genetic Algorithm (GA), Particle Swarm Optimization Technique (PSO).

## 1. Introduction

In spite of the development of power electronics resources, the direct current machines are becoming more and more useful. The PMDC motor is one of the most widely used prime movers in industry today. PMDC motors used in many applications such as steel rolling mills, electric tracking systems, textile mills including weaving and spinning, robotic manipulators, electric vehicle, etc[1, 2].

The position control of PMDC motor can be adjusted to a great extent so as to provide easy control and high performance. There are several conventional and numerical controller types intended for controlling the position of PMDC motor. This includes: PID Controller, Fuzzy Logic Controller; or the combination between them, in addition to optimized PID controllers including PID-PSO and PID-GA.

Conventional proportional-integral-derivative controller (PID) is widely used in industrial applications due to its simplicity in structure and ease to design. However it is difficult to achieve the desired control performance without controller tuning.

Tuning is important parameter for the best performance of PID controllers. PID controllers can be tuned in a variety of ways including hand tuning Ziegler Nichols tuning; Cohen–Coon tuning and Z–N step response, but these have their own limitations. Soft computing techniques like Genetic Algorithm and Particle Swarm Optimization have proved their excellence in giving better results by improving the steady state characteristics and performance indices[3].

[*] Egyptian Armed Force, Egypt.
[†] Beni–Suif University, Egypt.

The PID controller calculation involves three separate parameters: proportional, integral, and derivative values. The proportional value determines the reaction of the current error. The integral value determines the reaction based on the sum of recent errors, and derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via the final control element[3, 4]. The typical PID control law in its standard form is given by equation (1).

$$y(t) = \left[ k_p e(t) + k_d \frac{d(e)}{d(t)} + k_i \int_0^t e(t) d(t) \right] \qquad (1)$$

where, *e (t)* is the system error (difference between the reference input and the system output) $k_p$ is the proportional gain, $k_d$ is the derivative gain, and $k_i$ is the integral gain.

The general effects of control parameters are summarized in Table (1).

**Table (1) Effect of PID gains on the response**

| Parameter | Rise Time | Overshoot | Settling Time | Steady State Error |
|---|---|---|---|---|
| $k_p$ | Decrease | Increase | Small Change | Decrease |
| $k_i$ | Decrease | Increase | Increase | Eliminate |
| $k_d$ | Small Change | Decrease | Decrease | Small Change |

## 2. Mathematical Model of PMDC Motor

The goal in the development of the mathematical model is to relate the voltage applied to the armature to the angel of the motor. Two balance equations can be developed by considering the electrical and mechanical characteristics of the system. The following steps are to be made step by step;

- Represent the DC motor circuit diagram.
- Represent the system equations.
- Calculate the Transfer function of the system.
- Convert to model block
- Create a suitable Matlab/Simulink program to simulate the model.

### 2.1 Electrical Characteristics of PMDC Motor:

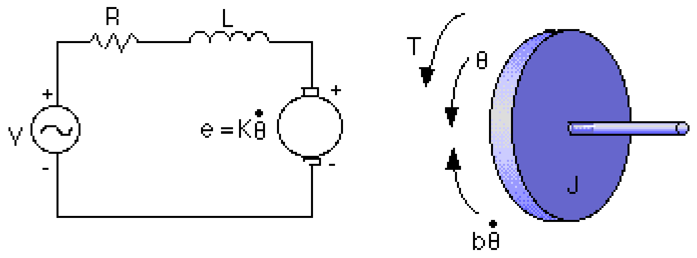

**Fig. (1) Equivalent electric circuit and mechanical system of permanent magnet DC motor.**

A differential equation for the equivalent circuit can be derived by using Kirchhoff's voltage law around the electrical loop. Kirchhoff's voltage law states that the sum of all voltages around a loop must equal zero as [5]:

$$V - V_R - V_L - V_e = 0 \tag{2}$$

$$V_R = iR \tag{3}$$

$$V_L = L\frac{d}{dt}i \tag{4}$$

$$V_e = K\dot{\theta} \tag{5}$$

where, $V$ is the voltage source, $V_R$ is the voltage across the armature resistance $R$, $V_L$ is the voltage across the armature inductance $L$, $V_e$ is the induced voltage, $i$ is armature current and $K$ is induced voltage (torque) constant.

## 2.2 Mechanical Characteristics of PMDC Motor:

Performing an energy balance on the system, the sum of the torques of the motor must equal zero. Therefore,

$$T_e - T_{\omega'} - T_\omega - T_L = 0 \tag{6}$$

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

$$T_{\omega'} = J\ddot{\theta} \tag{8}$$

$$T_\omega = B\dot{\theta} \tag{9}$$

where, $T_e$ is the electromagnetic torque, $T_{\omega'}$ is the torque due to rotational acceleration $\ddot{\theta}$ of the rotor and the motor moment of inertia $J$, $T_\omega$ is the torque produced from the angular velocity of the rotor $\dot{\theta}$ and its friction coefficient $B$, and $T_L$ is the mechanical load torque.
From the above electrical and mechanical system characteristics, the two balance equations of the system can be written as:

$$iR + L\frac{d}{dt}i = V - K\dot{\theta} \tag{10}$$

$$J\ddot{\theta} + B\dot{\theta} = Ki \tag{11}$$

## 2.3Transfer Function:

Using Laplace Transform, the system equations can be written as:

$$(R + Ls)i(s) = V - Ks\theta(s) \tag{12}$$

$$(Js + B)s\theta(s) = Ki(s) \tag{13}$$

where, $S$ denotes the Laplace operator.

From equations (12) and (13), the current $i(s)$ can be expressed as:

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

From equation (14), the following transfer function can be obtained, where the rotational speed is the output and the voltage is the input [6].

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

This transfer function can be represented and created as a model easily by using Simulink Library.

Before any consideration of the above equations, the constant values of the model data must be known. The data of the PMDC motor used in this paper is given in Table (2).

**Table (2): Parameters of PMDC motor**

| | | |
|---|---|---|
| Armature Resistance (*R*) | 2.7 | Ω |
| Inductance (*L*) | 4.3 | mH |
| Armature Voltage (*V*) | 24 | Volt |
| Rotor inertia (*J*) | 8 | Kg.cm$^2$ |
| Armature Current (*i*) | 1.5 | Amp |
| Damping coefficient (*B*) | 0.01 | N.m/rad/sec |
| Rated Speed (*n*) | 180 | rpm |

## 3. Tuning of PID Controller

Two stochastic optimization techniques will be used to search efficiently the optimal proportional-integral-derivative (PID) controller gains. These are genetic algorithm (GA) and particle swarm optimization (PSO). In this section the two techniques will be outlined to demonstrate their operation.

### 3.1 Genetic Algorithm (GA)

Genetic Algorithm GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems[7].

Basically, GA consists of three main stages: Selection, Crossover and Mutation. The application of these three basic operations allows the creation of new individuals which may be better than their parents.

The steps involved in creating and implementing a genetic algorithm are as follows:

1- Generate an initial, random population of individuals.
2- Evaluate their fitness (to minimize integral square error).
3- Select the fittest members of the population.
4- Implement crossover operation on the reproduced chromosomes (choosing probabilistically both the crossover site and the mates).
5- Execute mutation operation with low probability.
6- If the termination criteria reached then the process ends. If the termination criteria is not reached search for another best chromosome.

The GA architecture is shown in Figure (2).

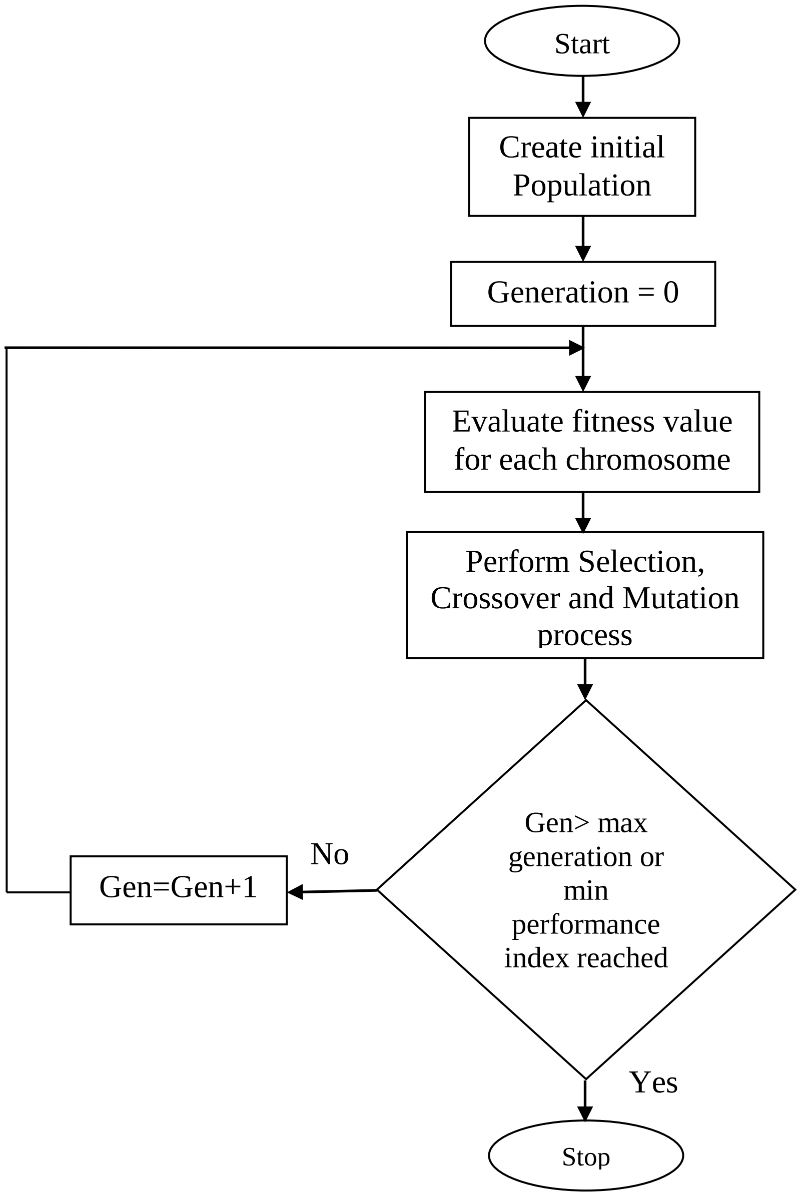
Start
Create initial Population
Generation = 0
Evaluate fitness value for each chromosome
Perform Selection, Crossover and Mutation process
Gen> max generation or min performance index reached
No
Gen=Gen+1
Yes
Stop

**Fig. (2) Genetic algorithm architecture.**

### 3.2 Particle Swarm Optimization(PSO)

Particle Swarm Optimization [PSO] is a population-based optimization method first proposed by Eberhart and Colleagues. Some of the attractive features of PSO include the ease of implementation and the fact that no gradient information is required. It can be used to solve a wide array of different optimization problems[8-10].

The technique is derived from research on swarms such as bird flocking and fish schooling. According to the research results for a flock of birds, birds find food by grouping or flocking and not individually. This observation leads to the assumption that every information is shared inside flocking [10]. The assumption is a basic concept of PSO.

In the PSO algorithm a flock of particles are put into the *d*-dimensional search space with randomly chosen velocities and positions knowing their best values so far (*Pbest*) and the position in the *d*-dimensional space. The velocity of each particle is adjusted according to its own flying experience and the other particles flying experience. For example, the *i*-th particle is represented as $x_i = (x_{i,1}, x_{i,2}, .., x_{i,d})$ in the *d*-dimensional space. The best previous position of the *i*-th particle is recorded and represented as [10]:

$$P_{besti} = (P_{besti,1}, P_{besti,2}, ..., P_{besti,d}) \tag{16}$$

The index of best particle among all of the particles in the group is $G_{best,d}$. The velocity for particle *i* is represented as $V_i = (V_{i,1}, V_{i,2}, \ldots, V_{i,d})$. The modified velocity and position of each particle can be calculated using the current velocity and the distance from $P_{besti,d}$ to $G_{best,d}$ as shown in the following formulas:

$$V_{i,m}^{(t+1)} = W \bullet V_{i,m}^{(t)} + C_1 \bullet rand() \bullet (Pbest_{i,m} - X_{i,m}^{(t)}) + C_2 \bullet rand() \bullet (Gbest_m - X_{i,m}^{(t)}) \tag{17}$$

$$X_{i,m}^{(t+1)} = X_{i,m}^{(t)} + V_{i,m}^{(t+1)} \quad ; i = 1, 2, ...n \quad ; m = 1, 2, ..d \tag{18}$$

where, *n* is number of particles in the group, *d* is the dimension, t is the pointer of iterations (generations), $V_{i,m}^{(t)}$ is the velocity of particle *i* at iteration *t* , $C_1$, $C_2$ are acceleration constants , rand() is random number between 0 and 1, $X_{i,d}^{(t)}$ is current position of particle *i* at iterations and *W* is the inertia weight factor, $P_{besti}$ Best previous position of the *i*-th particle, Gbest Best particle among all the particles in the population[10].
The structure of the PID controller with optimization algorithms is shown in Figure (3).

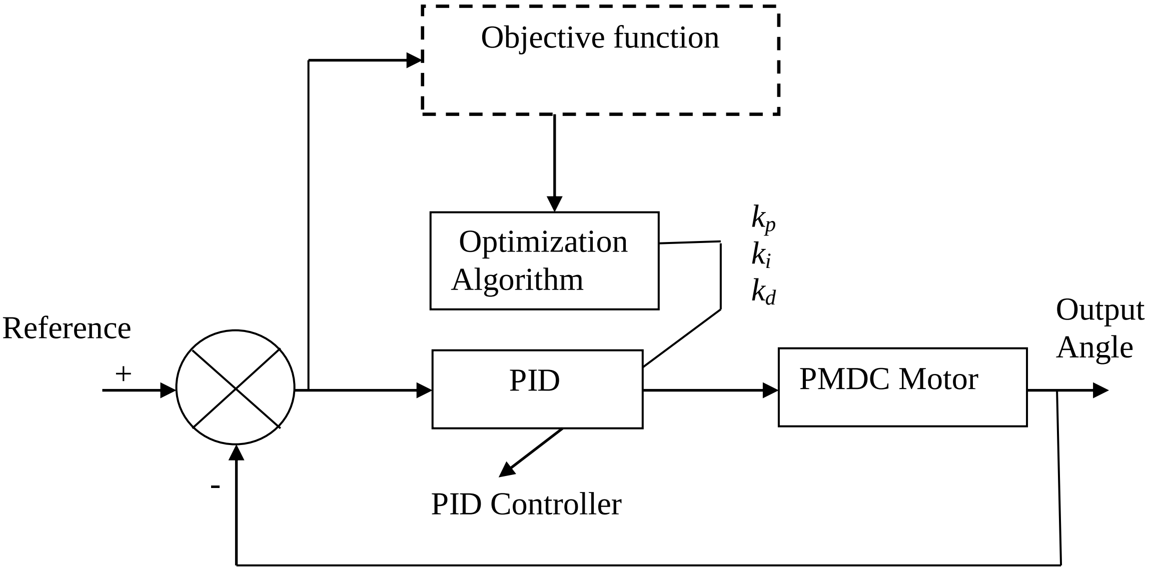

**Fig. (3) The block diagram of proposed PID controller with optimization Algorithms**

## 5. Simulation Results

In this section a comparison between PID controller using Particle Swarm Optimization technique (PID–PSO) and PID controller using Genetic Algorithm Optimization technique (PID–GA) is performed.
To make a fair comparison between the two optimization techniques, the two techniques are tried and run for 50 times and the optimized results are stored for evaluation. The comparison is done based on the worst optimized performance (case1) and the best optimized performance (case2) that can be obtained.

Table (3) shows the parameters of Genetic and PSO algorithms.

**Table (3): Parameters of Genetic Algorithm and PSO**

| Genetic Algorithm | | Particle Swarm Optimization | |
|---|---|---|---|
| GA property | Value | PSO parameters | Value |
| Population Size | 500 | Number of birds | 50 |
| Maximum Number of Generations | 20 | Number of iteration | 100 |
| Performance index/fitness function | Mean square error | *Max weight factor W max* | 0.9 |
| Crossover Method | Arithmetic Crossover | *Min weight factor W min* | 0.4 |
| Mutation Method | Uniform Mutation | Acceleration $C_1$ | 1.5 |
| Mutation Probability | 0.01 | Acceleration $C_2$ | 0.8 |

### 5.1 Case 1 (Worst Performance)

In this case the comparison between the worst performance obtained from (PID – PSO) and (PID –GA) controller techniques is explained as shown in Figures 4 and 5. The two techniques are performed 50 times.

It can be noticed from examining the response curves of these Figures, that the transient behaviour of the system when equipped with PID-PSO controller is slightly better than that with PID - GA controller during the starting period. This is because a minimum rise time obtained with PID-PSO controller. The worst performance parameters are clear in Table (4).

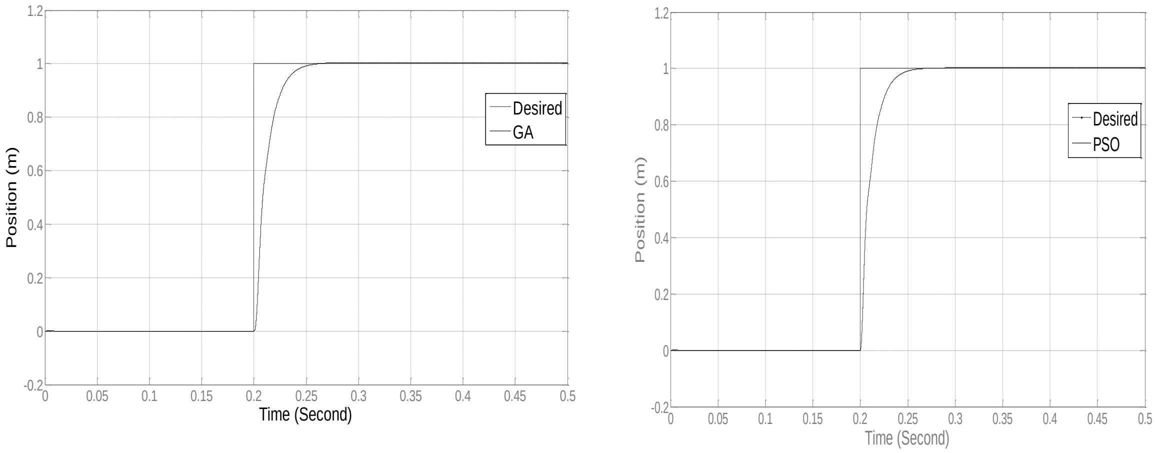

**Fig. (4) The worst response of PMDC motor using GA and PSO**

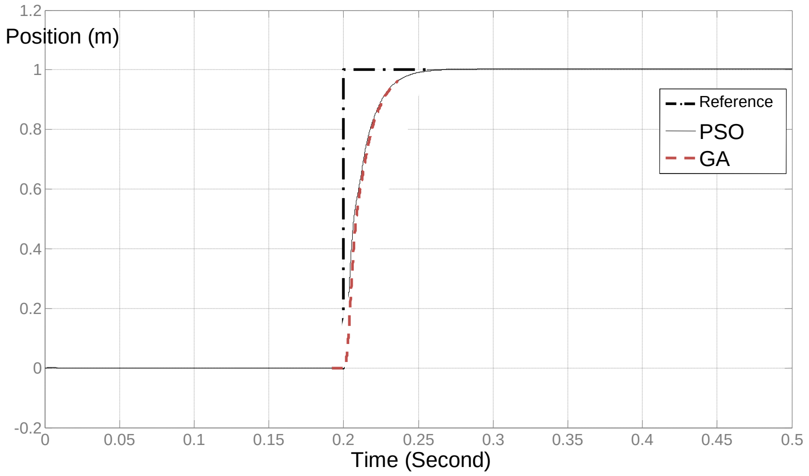

**Fig. (5) Comparison between PSO and GA in the worst response**

**Table (4): Parameters of the worst performance**

| Results | PID – GA- Controller | PID – PSO - Controller |
|---|---|---|
| Rising time | 0.0345 sec | 0.0256 sec |
| Overshoot | 0 | 0 |
| Steady-State- Error | 0.0045 m | 0.003 m |

## 5.2 Case 2

This case study examines the best performance that is obtained from (PID–PSO) and (PID-GA) techniques as shown in Figures 6 and 7.

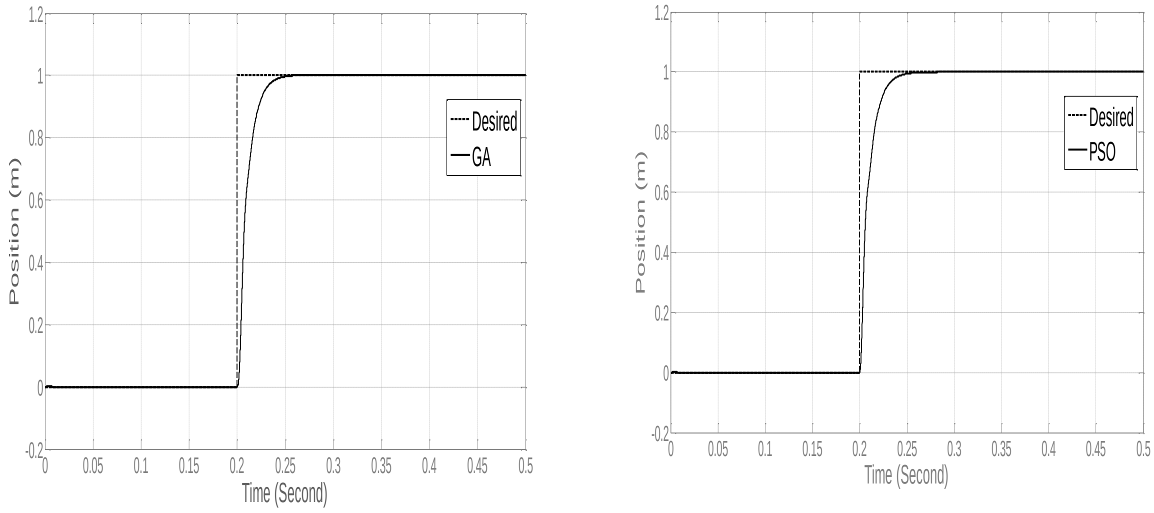

**Fig. (6) The best response of PMDC motor using GA and PSO**

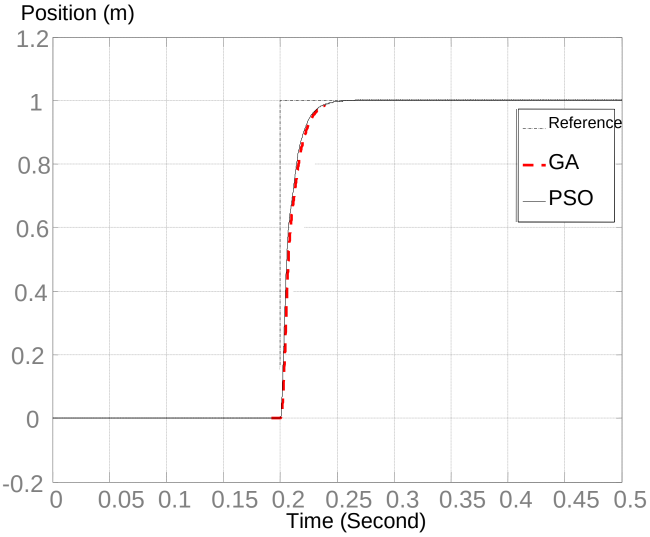

**Fig. (7) Comparison between PSO and GA in the best response**

It is clear that the transient behaviour of the PID-PSO controller in this case is also better than that of PID-GA controller. This can be explained from performance parameters in Table 5, where the PID-PSO controller achieves less rising time.

**Table (5): Performance of PID – PSO- Controller**

| Results | PID – Genetic-Controller | PID – PSO - Controller |
|---|---|---|
| Rising time | 0.03 sec | 0.02 sec |
| Overshoot | 0 | 0 |
| Steady-State- Error | 0 | 0 |

From the above case studies, it is clear that the PSO response slightly supersedes that of the GA optimization technique.

## 6. Conclusion:

In this paper Genetic and PSO algorithms for PID tuning are presented. The angle of a PMDC Motor drive is controlled by PID-PSO and PID-GA controllers. The simulation results show that the proposed controllers can perform an efficient search for the optimal PID controller parameters. By trying a fair comparison, the results show that PID-PSO controller can improve the dynamic performance of the system in a better way compared to PID-GA., however both systems offer excellent steady state response and performance indices.

## References:

[1] A. E. Fitzegrald. C. Kingsley and Stephen D. Umans, Electric Machinery, 6th ed. ed. New York: The McGraw-Hill Companies, 2003.

[2] S. S. Patil and P. Bhaskar, "Design and Real Time Implementation of Integrated Fuzzy Logic Controller for a High Speed PMDC Motor," International Journal of Electronic Engineering Research, vol. 1, pp. 13–25, 2009.

[3] J. Jantzen, "Tuning of Fuzzy PID Controllers," in Technical University of Denmark, Department of Automation. Bldg 326, DK-2800 Lyngby, DENMARK, Sep 1998.

[4] M. J. Willis, Proportional-Integral-Derivative Control, 6th October, 1999.

[5] M. E. Fisher, A. Ghosh and A. M. Sharaf, "IntellIgent Control Strategies for Permanent Magnet DC Motor Drives," presented at Power Electronics,Drives and Energy systems for Industrial Growth, New Delhi, 1996.

[6] W. P. Aung, "Analysis on Modeling and Simulink of DC Motor and its Driving System Used for Wheeled Mobile Robot," 2007.

[7] R. Malhotra, N. Singh and Y. Singh, "Design of Embedded Hybrid Fuzzy-GA Control Strategy for Speed Control of DC Motor: A Servo Control Case Study," International Journal of Computer Applications (0975 – 8887), vol. 6, September 2010.

[8] K. Y. Lee and M. A. El-Sharkawi, Modern Heuristic Optimization Techniques. Canada: John Wiley and Sons, 2008.

[9] B. ALLAOUA, B. GASBAOUI and B. MEBARKI, "Setting Up PID DC Motor Speed Control Alteration Parameters Using Particle Swarm Optimization Strategy," Leonardo Electronic Journal of Practices and Technologies, pp. p. 19-32, January-June 2009.

[10] Mehdi Nasri, Hossein Nezamabadi-pour, and Malihe Maghfoori, "A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor," World Academy of Science, Engineering and Technology, 2007.