Speed Control Of Dc Motor Using Particle Swarm Optimization Technique

[1]Akhilesh Kumar Mishra* and [2] Anirudha Narain

[1]M.Tech. Scholar (Control & Instrumentation), Department of Electrical Engineering, Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh, INDIA.

[2] Associate Professor, Department of Electrical Engineering, Department of Electrical Engineering, Motilal Nehru National Institute of Technology, Allahabad, Uttar Pradesh, INDIA.

ABSTRACT

The aim of this work is to design a speed controller of a DC motor by selection of PID parameters using bio-inspired optimization technique i.e. Particle Swarm Optimization (PSO). Here, model of a DC motor is considered as a second order system for armature voltage control method of speed control. In this work bio-inspired optimization technique in controllers and their advantages over conventional methods is discussed using MATLAB/Simulink. This proposed optimization methods could be applied for higher order system also to provide better system performance with minimum errors. The main aim is to apply PSO technique to design and tune parameters of PID controller to get an output with better dynamic and static performance. The application of PSO to the PID controller imparts it the ability of tuning itself automatically in an on-line process while the application of optimization algorithm to the PID controller makes it to give an optimum output by searching for the best set of solutions for the PID parameters.

Keywords: -Ziegler Nicolas, Cohen-Coon, Particle Swarm Optimization, PID controller, Parameter tuning.

1. INTRODUCTION

DC motor drives are widely used in applications requiring adjustable speed, good speed regulations and frequent starting, braking and reversing. Some important applications are rolling mills, paper mills, mine winders, hoists, machine tools, traction, printing presses, textile mills, excavators and cranes. Fractional horsepower DC motors are widely used as servo motors for positioning and tracking. Although, it is being predicted that AC drives will replace DC drives, however, even today the variable speed applications are dominated by DC drives because of lower cost, reliability and simple control. As per the control of DC motor, there are lot of methods to control the speed and position of the motor. The purpose of a motor speed controller is to take a signal representing the demanded speed and to drive a motor at that speed.

PID(proportional-integral-derivative)control is one of the earlier control strategies. It has a simple control structure which was understood by plant operators and which they found relatively easy to tune. Since many control systems using. PID control have proved satisfactory, it still has a wide range of applications in industrial control. PID control is a control strategy that has been successfully used over many years. Simplicity, robustness, a wide range of applicability and near-optimal performance are some of the reasons that have made PID controller so popular in the academic and industry sectors. Recently, it has been noticed that PID controllers are often poorly tuned and some efforts have been made to systematically resolve this matter. PID control has been an active research topic for many years; since many process plants controlled by PID controllers have similar dynamics it has been found possible to set satisfactory controller parameters from less plant information than a complete mathematical model. These techniques came about because of the desire to adjust controller parameters with a minimum of effort, and also because of the possible difficulty and poor cost benefit of obtaining mathematical models.

The PID controller calculation (algorithm) involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the 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 a control element. By tuning the three constants in the PID controller algorithm, the controller can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation.

Particle swarm optimization (PSO) is an evolutionary meta-heuristic algorithm based on the collective behavior emerging from the interaction of the different search threads that has proved effective in solving combinatorial optimization problems. The

PSO was inspired from natural behavior of the bird flock on how they find the optimal path and bring back to their nest by building the unique trail formation. The first algorithm which can be classified within this framework was presented in 1994 and, since then, many diverse variants of the basic principle have been reported in the literature. The objective of this algorithm is to optimize the gains of the PID controller for the given plant. In proposed PSO-PID controller, PSO algorithm is used to optimize the gains and the values are applied into the controller of the plant. The objective of this algorithm is to optimize the gains of the PID controller for the given plant. The proportional gain makes the controller respond to the error while the integral derivative gain help to eliminate steady state error and prevent overshoot respectively.

2. MATHEMATICAL ANALYSIS OF DC MOTOR

In armature control of separately excited DC motors, the voltage applied to the armature of the motor is adjusted without changing the voltage applied to the field. Figure 1 shows a [1]DC motor equivalent model.

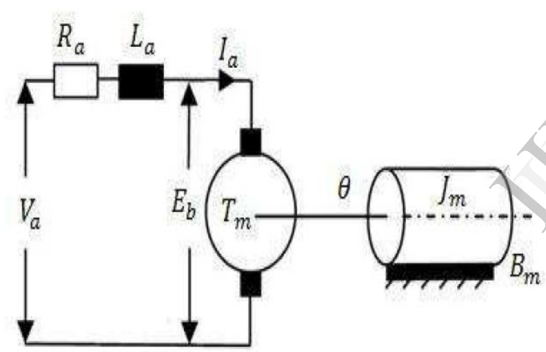

Fig.1 D.C. motor model

Some useful relations are:

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + E_b(t) \qquad \ldots(1)$$

$$E_b(t) = K_b \omega(t) \qquad \ldots(2)$$

$$T_m(t) = K_t i_a(t) \qquad \ldots(3)$$

$$T_m(t) - T_L(t) = J_m B_m \omega(t) \qquad \ldots(4)$$

where V_a = armature voltage (V), R_a = armature resistance (Ω), L_a = armature inductance (H), I_a = armature current (A), E_b = Back emf (V), ω = angular speed (rad/sec), T_m = motor torque (Nm), T_L = load torque (Nm), θ = angular position of rotor shaft (rad), J_m = rotor inertia (kgm2), B_m = viscous friction coefficient (Nms/rad), K_t = torque constant (Nm/A), K_b = Back emf constant (Vs/rad).

Figure 2 showing the basic block diagram of DC motor model including their transfer functions. V_a is the input supply, T_L is load torque and ω is angular speed.

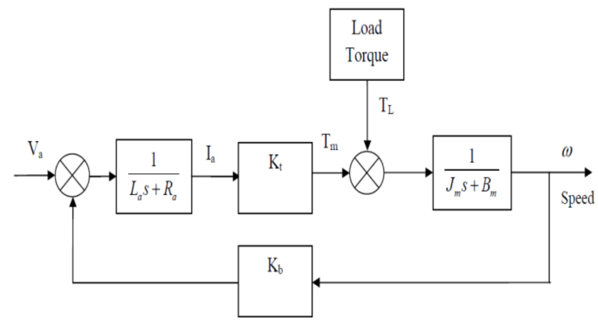

Fig.2 Block diagram of D.C. motor model

2.1 Speed Control of DC Motor

Substitute (3) in (2) and (4) in (5), we get

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_b \omega(t) \ldots (5)$$

$$K_t i_a(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \qquad \ldots(6)$$

Taking Laplace transform of equation (5) and (6),

$$V_a(s) = R_a i_a(s) + s\, L_a I_a(s) + K_b \omega(s) \ldots (7)$$

$$K_t I_a(s) = s\, J_m \omega(s) + B_m \omega(s) \qquad \ldots(8)$$

There are two possible conditions:

When $T_L = 0$

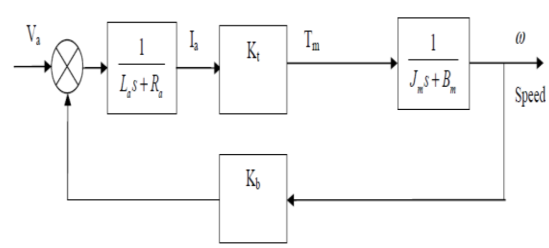

Fig.3 Block diagram D.C. motor model when $T_L = 0$

Figure 3 shows that the DC motor is running under no-load condition (ideal) i.e. $T_L = 0$. Now find the transfer function of ω(s) with respect to Va(s).

So, the relation between motor speed and applied voltage is given by the transfer function,

$$\frac{\omega(s)}{V_a(s)} = \frac{K_t}{L_a J_m s^2 + (R_a J_m + L_a B_m)s + (R_a B_m + K_b K_t)}$$

Motor Parameter

Power P = 8 watts, Speed N = 5000 rpm (max), rotor inertia $\mathbf{J_m}$ is assumed to be 0.01 and Supply voltage Vt = 12 volts. Therefore for the max speed rpm of 5000, it can be calculate the torque constant K;

$$\frac{V_t}{K} = \frac{2\pi N}{60} = \omega_m \qquad \text{.... (10)}$$

$\mathbf{K_t}$ = 0.023 and ω_m = 524 radsec^{-1}

From equation no.7 as $\frac{d\theta}{dt} = \omega$

$$K_t i_a(t) = J_m \frac{d\omega(t)}{dt} + B_m \omega(t) \quad \text{...(11)}$$

At the steady state (used as analyzed data), both **I** and **ω** are stabilized: $\left(\frac{d\theta}{dt} = \omega = 0\right)$

$\frac{P}{\omega} = T$; Where **W** mentioned as the minimum possible speed to rotate the DC motor, 1200 rpm;

T = 15.27 Nm, Therefore, the total equivalent damping $\mathbf{B_m}$ can be chosen the value of;
(0.023*0.663) – $\mathbf{B_m}$ (524) = 0
$\mathbf{B_m}$ = 0.00003

By calculating and assuming the require data as above, the Motor Model. **Va**=12V; $\mathbf{J_m}$=0.01; $\mathbf{B_m}$=0.00003; **Kt** =0.023; **R** =1ohm; and **L** =0.5H;

The Matlab code for the motor is as follows

```
Kt =.023;
Kb =.023;
R = 1;
L = .5;
J = .01;
b = .00003;
num = Kt;
den=[(J*L)((J*R)+(L*b))((b*R)+Kt*Kb)];
Dcmotor=tf (num, den)
```

3.SPEED CONTROL USING CLASSICAL PID TUNING METHODS

[2]The PID controller is the most common general purpose controller in the today's industries. It can be used as a single unit or it can be a part of a distributed computer control system.

After implementing the PID controller, now we have to tune the controller; and there are different approaches to tune the PID parameters like P, I and D. The Proportional (P) part is responsible for following the desired set-point while the Integral (I) and Derivative (D) part account for the accumulation of past errors and the rate of change of error in the process or plant, respectively.

PID controller consists of three types of control i.e. Proportional, Integral and Derivative control

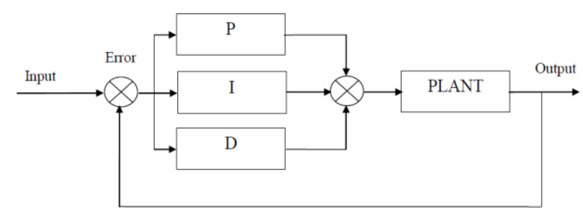

Fig.4 Schematic of PID controller

The system transfer function in continuous s-domain are given as

For $P = K_p, I = K_i / s$ and $D = K_d s$

$$G_c(s) = P + I + D = K_p + \frac{K_i}{s} + K_d s \quad \text{...(12)}$$

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \quad \text{..(13)}$$

Where K_p is the proportional gain, K_i is the integration coefficient and K_d is the derivative coefficient.
T_i is known as the integral action time or reset time and T_d is the derivative action time or rate time

There are various tuning strategies based on an open-loop step response. While they all follow the same basic idea, they differ in slightly in how they extract the model parameters from the recorded response, and also differ slightly as to relate appropriate tuning constants to the model parameters. There are different methods, the classic Ziegler-Nichols test, and Cohen- Coon test. Naturally if the response is not sigmoid or 'S' shaped and exhibits overshoot, or an integrator, then this tuning method is not applicable.

This method implicitly assumes the plant can be adequately approximated by a first order transfer function with time delay.

$$Gp = \frac{Ke^{-\theta s}}{Ts+1} \qquad \dots (14)$$

Where K is gain, θ is the dead time or time delay, and T is the open loop process time constant. Once we have recorded the open loop input/output data, and subsequently measured the times T and θ, the PID tuning parameters can be obtained directly from the given tables for different classical methods.

Figure 5. Block diagram of plant with variable output

The method is based on computing the times t1 and t2 at which the 35.3% and 85.3% of the system response is obtained as shown in the figure:

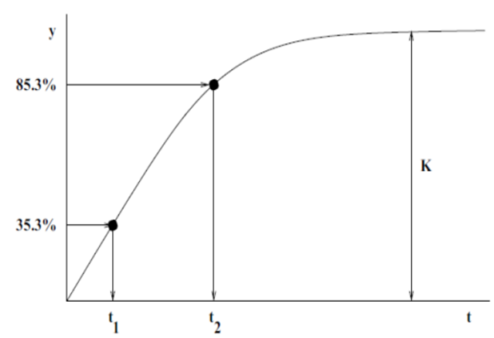

Figure 6 System response for first order time delay transfer function

After computing the t_1 and t_2 times, the time delay (θ) and process time constant (T) can be obtained from the following equations:

$$\theta = 1.3\, t_1 - 0.29 t_2$$

$$T = 0.67(t_2 - t_1) \qquad \dots (15)$$

Ziegler-Nichols Tuning Method

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (14) as derived by Ziegler-Nichols.

They often form the basis for tuning procedures used by controller manufacturers and process industry. The methods are based on determination of some features of process dynamics. The controller parameters are then expressed in terms of the features by simple formulas. The method presented by Ziegler and Nichols is based on a registration of the open-loop step response of the system, which is characterized by two parameters. First determined, and the tangent at this point is drawn. The intersections between the tangent and the coordinate axes give the parameters *T* andθ. A model of the process to be controlled was derived from these parameters. This corresponds to modeling a process by an integrator and a time delay. Ziegler and Nichols have given PID parameters directly as functions of *T* andθ. The behavior of the controller is as can be expected. The decay ratio for the step response is close to one quarter. It is smaller for the load disturbance. The overshoot in the set point response is too large.

Table 1 Ziegler Nichols method

Controller		K_p	T_i	T_d
Ziegler-Nichols Method	P	T/Kθ	-	-
	PI	0.9T/Kθ	θ/0.3	-
	PID	1.2T/Kθ	2θ	0.5θ

Cohen-Coon Tuning Method

Cohen and Coon based the controller settings on the three parametersθ, *T* and *K* of the open loop step response. The main design criterion is rejection of load disturbances. The method attempts to position closed loop poles such that a quarter decay ration is achieved.

The PID tuning parameters as a function of the open loop model parameters K, T and θ from equation (14) as derived by Cohen-Coon:

Table 2 Cohen Coon method

Controller		K_p	T_i	T_d
Cohen-Coon Method (Open Loop)	P	$\frac{T}{K\theta}\left(1+\frac{\theta}{3T}\right)$	-	-
	PI	$\frac{T}{K\theta}\left(0.9+\frac{\theta}{12T}\right)$	$\theta\left(\frac{30+3\theta/T}{9+20\theta/T}\right)$	-
	PID	$\frac{T}{K\theta}\left(\frac{4}{3}+\frac{\theta}{4T}\right)$	$\theta\left(\frac{32+6\theta/T}{13+8\theta/T}\right)$	$\theta\left(\frac{4}{11+2\theta/T}\right)$

4.PARTICLE SWARM OPTIMIZATION

[3]James Kennedy an American Social Psychologist along with Russell C.Eberhart innovated a new evolutionary computational technique termed as Particle Swarm Optimization in 1995.The approach is suitable for solving nonlinear problem. The approach

is based on the swarm behavior such as birds finding food by flocking. A basic variant of the PSO algorithm works by having population (called a swarm) of candidate solution (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Here in this technique a set of particles are put in d-dimensional search space with randomly choosing velocity and position. The initial position of the particle is taken as the best position for the start and then the velocity of the particle is updated based on the experience of other particles of the swarming population.

Algorithm of PSO

The ith particle in the swarm is represented as

$Xi = (xi1, xi2, xi3, \dots\dots\dots xid)$ in the d-dimensional space.

The best previous positions of the ith particle are represented as:

$$Pbest = (Pbesti,1, Pbesti,2, Pbesti,3 \dots\dots Pbesti,d)$$

The index of the best particle among the group is Gbestd.

Velocity of the ith particle is represented as

$$Vi = (Vi,1\ Vi,2\ Vi,3 \dots\dots Vi,d)$$

The updated velocity and the distance from Pbestid to Gbesti,d is given as ;

$$Vi,m^{t+1} = W * Vi,m^{t} + C1 * rand() * (Pbesti,m - Xi,m^{t}) + C2 * rand() * (Gbest\ m$$

$Xi,d^{(t+1)} = Xim^{(t)} + Vi,m^{(t+1)}$

For i=1, 2, 3.......n.

m = 1, 2, 3.....d.

where,

n:- Number of particles in the group.

d: - dimension index.

t: - Pointer of iteration.

$Vi,m^{(t)}$:- Velocity of particle at iteration i.

W: - Inertia weight factor.

C1, C2:- Acceleration Constant.

rand ():- Random number between 0 and 1.

$Xi,d^{(t)}$:- Current position of the particle 'i' at iteration.

Pbest: - Best previous position of the ith particle.

Gbest:- Best particle among all the particle in the swarming population.

Algorithmic Approach for the Specified Design:-

In our case, we cast the PID controller design problem in PSO framework as given. We consider the three dimensional search space. K_P, K_I and K_D are the three dimensions. We consider the fitness function based on time domain Characteristics for adaptation. We set the number of adaptation iterations based on expected parameters and time of computation.

Objective Function for Particle swarm optimization

```
function F= tightnes (kd, kp, ki)
T=tf([.023*kd .023*kp .023*ki],[.005 (.010015+.023*kd) (.000559+.023*kp).023*ki]);
S=stepinfo (T1);
tr=S.RiseTime;
ts=S.SettlingTime;
Mp=S.Overshoot;
Ess=1/(1+dcgain(T1));
F= (1-exp (-0.5))*(Mp+Ess) +exp(-0.5)*(ts-tr);
```

Table 3 Parameter For PSO

Parameter	Values
Acceleration Constant. C_1	1.2
Acceleration Constant.C_2	1.2
Inertia weight factor.	.9
No. of Particles	300
No. of Iterations	50

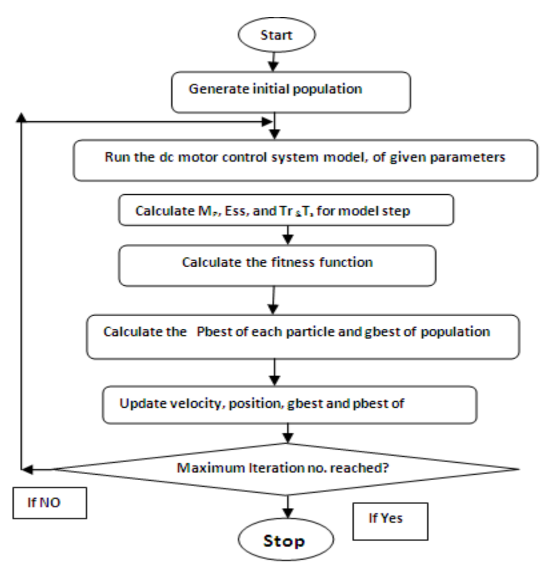

Fig. 7 Flowchart of Particle Swarm Optimization

4. SIMULINK MODEL OF DC MOTOR

The Simulink model of DC motor using is shown in Fig 8.

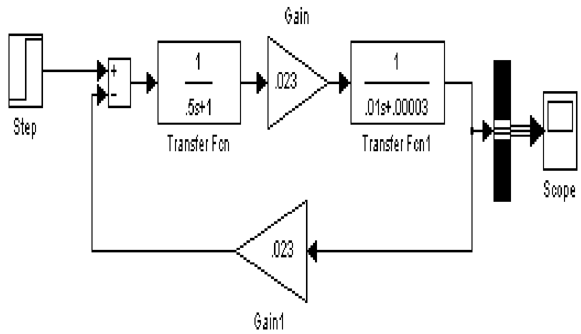

Fig.8 Simulink model of DC motor

The Simulink model of various tuning method for speed control of DC motor using PID controller is shown in Fig 9.

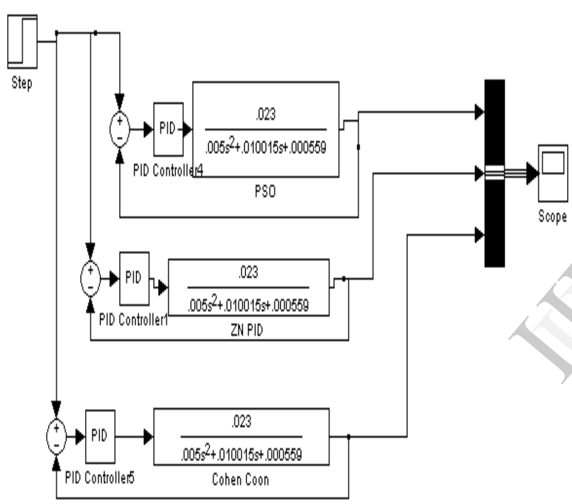

Fig.9 Simulink model of various tuning methods

The parameters used to describe the electrical and electromechanical systems are given below.

Table 3 Parameters of DC Motor

Parameter	Values & Unit
R	1Ω
K_b	.023Kg-m/A
K_t	.023Vs/rad
L	.0.5H
J_m	.01Kgms^2/rad
B_m	.00003Kgms/rad

5. RESULTS AND DISCUSSION

The Simulink model in Fig. 8 & 9 was simulated and the plots for various tuning method were observed. Fig. 10 and Fig. 11 shows the Speed versus Time plot for conventional and bio inspired optimization method (PSO) respectively .

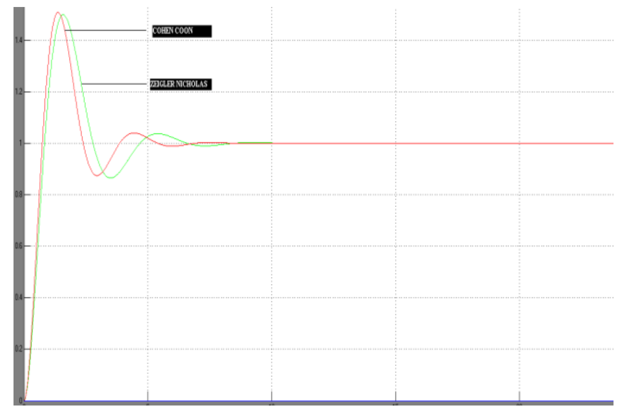

Fig. 10 Speed versus Time plot with reference speed for PID tuned with Zeigler Nicholas & Cohen Coon

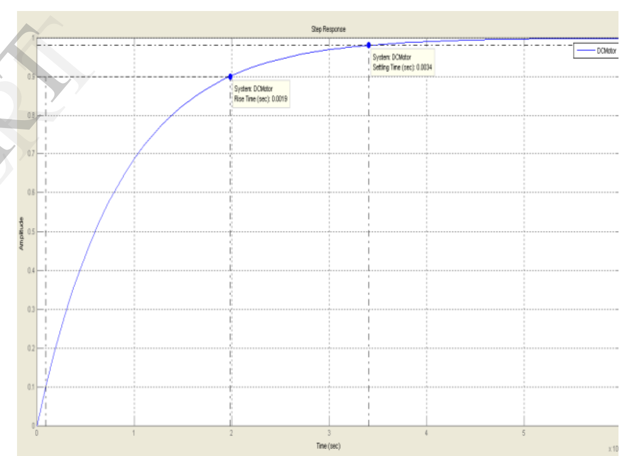

Fig. 11 Speed versus Time plot with reference speed for PID tuned with Particle swarm optimization

From the above two result it is clear that bio inspired optimization method is far better than the conventional tuning method. Their comparison is shown in figure 12 and detailed comparative analysis considering all the parameter is given in Table 4.

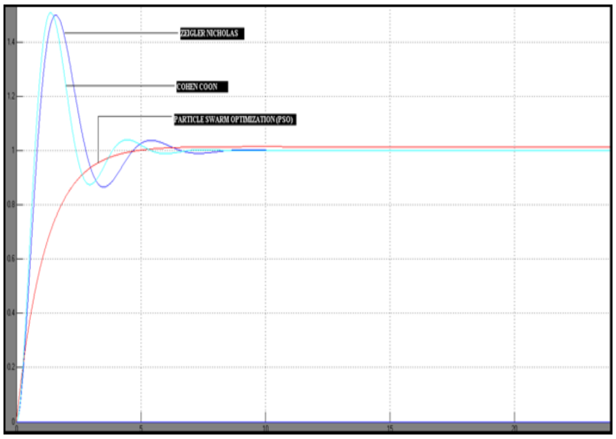

Fig. 12 Speed versus Time plot with reference speed for PID tuned with ZN, CC, and PSO

Table 4 Comparative analysis of various tuning methods

Method	Settling Time(T_s)	Rise Time(T_r)	Overshoot (%)	Steady state error(E_{ss})	Controller Parameter (K_p, K_i & K_d)
Without Controller	3.8279	0.7233	19.4273	024	
Z-Nicolas	12.3	0.457	75.1	0	[1.385, 1.6956, .391797]
Cohen Coon	9.17	0.444	68.7	0	[1.54502, 1.55074, 1.55074]
PSO	0.0034	0.0019	0	0	[220.538 3.04 252.842]

It can be seen from the above comparison table that while using the bio-inspired technique (Particle Swarm Optimization) the overshoots obtained is zero as compared to the case when the PID Controller is was tuned via conventional methods. The settling time is also lesser in case of the Particle Swarm Optimization, also the rise time is reduced. The Particle Swarm Optimization PID controller tends to approach the reference speed faster and has, comparatively, a zero overshoot. It can be observed from Fig 12 that the Conventional PID controller have overshoot from the reference speed and attain a steady state with larger settling time

6. CONCLUSION

Performance comparison of different controllers has been reviewed and it is found that Particle Swarm Optimization is best among the all methods which are used for tuning the parameter of PID controller for which settling time and rise is found to be less. The conventional controllers however are not recommended for higher order and complex systems as they can cause the system to become unstable. Hence, a heuristic approach is required for choice of the controller parameters which can be provided with the help of Bio inspired methods such as Particle Swarm Optimization, where we can define variables in a subjective way.

ACKNOWLEDGMENT

This work was supported by Mr. Amit Gupta. Special Thanks to Assistant Professor Anukaran Khanna (United College Of Engineering & Research, Allahabad India) for his co-operation.

7. REFERENCES

[1] Gopal K. Dubey, "Fundamentals of Electrical Drives", Narosa Publishing House Pvt. Ltd., 2001, chap. 6.

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

[3] J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge", Proceeding of the IEEE International Conference on Evolutionary Computation, ICEC1997, Indianapolis, pp. 303-308, 1997.

[4] Nasri M. I, pour H. N, and Maghfoori M., 2007 "A PSO-Based Optimum Design of PID Controller for a Linear Brushless DC Motor" Proceedings Of World Academy Of Science, Engineering And Technology Vol. 20 April

[5] Benjamin C. Kuo, Farid Golnaraghi, 2009. Automatic Control Systems, 9th ed., John Wiley & Sons

[6] Liu, Y., Zhang, J. and Wang, S. (2004). Optimization design based on PSO algorithm for PID controller. *Proceedings of 5th World Congress on Intelligent Control and Automation, Vol. 3, 2419-2422.*

[7] T. Hägglund and K. J. Åström,‖ Revisiting The Ziegler-Nichols Tuning Rule For Pi Control— Part II The Frequency Response Method‖, Asian Journal of Control, Vol.No. 4, December 2004,pp. 469-482

[8] I. J. Nagrath, M. Gopal, "Control Systems Engineering", New Age International Publishers, 2007, chap. 3

[9] D. H. Kim, J. H. Cho; '*A Biologically Inspired Intelligent PID Controller Tuning for AVR Systems*', International Journal of Control, Automation and Systems, 2006, Vol. 4, No. 5, pp. 624-636.