International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 28527-28538

http://www.ripublication.com

Dc Motor Speed Control Using Fuzzy PID Controller

M. Kathamuthu, G. Balasubramanian, K. Ramkumar
School of Electrical and Electronics Engineering
SASTRA University, Thanjavur.
Email: muthuraaj.raga@gmail.com

Abstract

Proportional-Integral-Derivative (PID) control is the universally accepted and commonly used algorithm in industry. One of the applications used here is the speed control of DC motor. DC motors are used in industry extensively due to their low cost,low energy consumption and their easy adaptation with digital systems. Controlling the speed of a DC motors is very important as any small change can lead to instability of the closed loop system. This paper presents the fuzzy PID controller (FPID) implementation for DC Motor speed control. The algorithms of FPID and conventional PID controller are implemented in simulation tool kits of the MATLAB Simulink environment. The simulation result demonstrates that the proposed fuzzy PID controller realize a perfect performance compared to conventional controller.

Keywords: DC motor, PID, and Fuzzy logic, PSO.

Introduction

Due to the simple structure and good performance, PID controllers are mostly used for DC motor speed control. In addition, with technology advancements, PID controller parameters can be easily changed without changing any hardware. However, the PID controller performance depends on the accuracy of the system model and parameters. In practice, the controlled systems are usually nonlinear and hence their accurate mathematical models are not available. In addition, the system parameters can vary with the time and the operating conditions. Therefore, the tuning methods of the controller parameters are of great importance. The majority of control systems are operated by PID controllers. Because of the simplicity, applicability and easy usage, more than 95% of the controllers are of PID type used in the industrial process control applications [1-2].The PID controllers provide stable and good performance if it is tuned accurately.

Among the conventional tuning methods, the Ziegler-Nichols (ZN) technique is the most well-known method [3-4]. It is robust against the system model uncertainty and good for many industrial applications. However, it does not provide an optimal tuning since it produces a high overshoot in the system response. To improve the performance of the conventional tuning methods, various intelligent methods have been presented [4-9].

The major issues during applying a traditional algorithm in a speed controller is the results of non-linearity in a DC motor. Generally, a correct nonlinear model of an actual DC motor is tough to seek out and parameter obtained from systems identification could also be solely approximated values. The field of fuzzy implementation has been creating fast progress in recent years. Fuzzy logic Controller (FLC) has been a particularly active and fruitful analysis with several industrial applications are reported [10]

In the last three decades, an alternative to the traditional control methods in various applications are the evolution of FLC.FLC design technique does not need any prior knowledge about the system. Imitating the method of human learning, the rate change of the error and the tracking error are considered as the two inputs for designing a fuzzy control system [11-12].

In this paper, the PID controller parameter tuning is based on Particle Swarm Optimization (PSO) and fuzzy PID is proposed. The system is modelled using MAT-LAB/SIMULINK and the simulation results obtained are compared with those of conventional PID controller. The comparison indicates the effectiveness of the proposed tuning method as it gives a better performance and satisfies the specified control characteristics.

Dc Motor Modeling

Although their maintenance costs are higher than other motors, DC motors are most suitable motors for adjustable speed control applications that need high control requirements. The equivalent circuit of a separately excited DC motor is shown in figure 1.

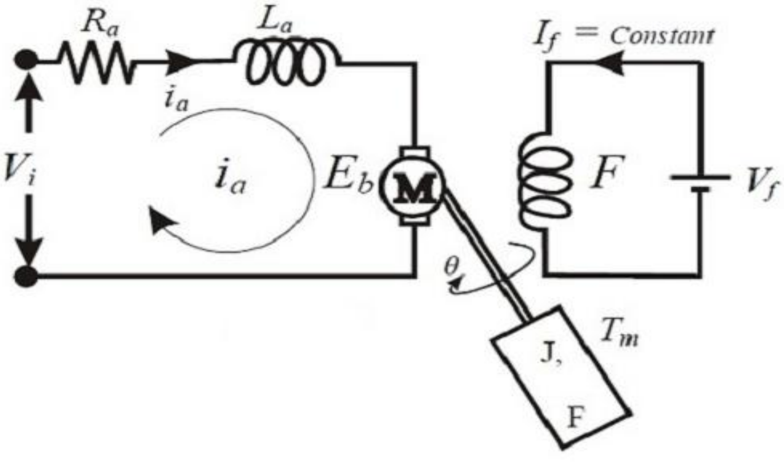

Figure 1: Schematic Diagram of DC motor

The torque T generated is on the motor shaft and it is proportional to armature current,

$T \propto i_a$

$T = k_T i_a$

The back emf developed is proportional to angular velocity,

$V_b \, \alpha \, \omega$

$V_b = k_E \, \omega$

Take the Kirchhoffs voltage law of the given circuit,

$$R_a i_a + \frac{L_a \, di_a}{dt} + V_b = V_a \quad \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots .. \qquad (1)$$

We assumed that no saturation occurs in the magnetic circuit of the machine provided the magnitude of the input signal is kept smaller than the rated voltage of the machine. The output of the motor is mechanical; armature current produces the torque in the mechanical part. $J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} + T_L = T$ here, θ = angular displacement

so above equation becomes,

$$J\frac{d\omega}{dt} + B\omega + T_L = T$$

Torque can be written as, $T = K_T i_a$

ie,

$$J\frac{d^2\theta}{dt^2} + B\frac{d\theta}{dt} + T_L = k_T i_a \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \qquad (2)$$

take Laplace transform of (1) equation

$$R_a I_a(S) + L_a S I_a(S) + V_b = V_a$$

$$I_a(S)[L_a S + R_a] = V_a - V_b$$

$$I_a = V_a - V_b / L_a\,(S) R_a$$

Then take Laplace transform of (2) equation,

$$JS_\omega(S) + B_\omega\,(S) + T_L = K_T i_a$$

$$\omega(S) = K_T i_a - \frac{T_L}{J_s + B}$$

After simplification, we get the transfer function as, For loaded condition,

$$\frac{\omega}{V_a} = \frac{K_T - \frac{T_L(LaS+Ra)}{Va}}{K_T K_E + (J_S + B)(LaS + Ra)}$$

For no load condition

$$\frac{\omega}{Va} = K_T / K_T K_E + (J_S + B)(LaS + Ra)$$

The parameters of DC motors are tabulated in table1.

Table 1: Dc Motor Parameters

Description of the parameter	Parameter Value
Armature resistance (Ra)	1.2 ohm
Armature inductance (La)	2.7 mH
Armature voltage (Va)	220V
Mechanical inertia (Jm)	0.00025Nm/rad/sec
Friction Coefficient (Bm)	0.001 Nm/rad/sec
Back emf constant (K)	0.05 V/rad/sec
Rated speed (N)	1500 Rpm
Motor torque constant	0.9688

PID Controller Design

The PID controller parameters are proportional gain, integral gain and derivative gain. The proportional gain provides a control action proportional to the error. The integral action reduces the steady state error. While, the derivative action improves the transient response. The PID controller is made by the summation of these three actions as shown in Fig. 2.

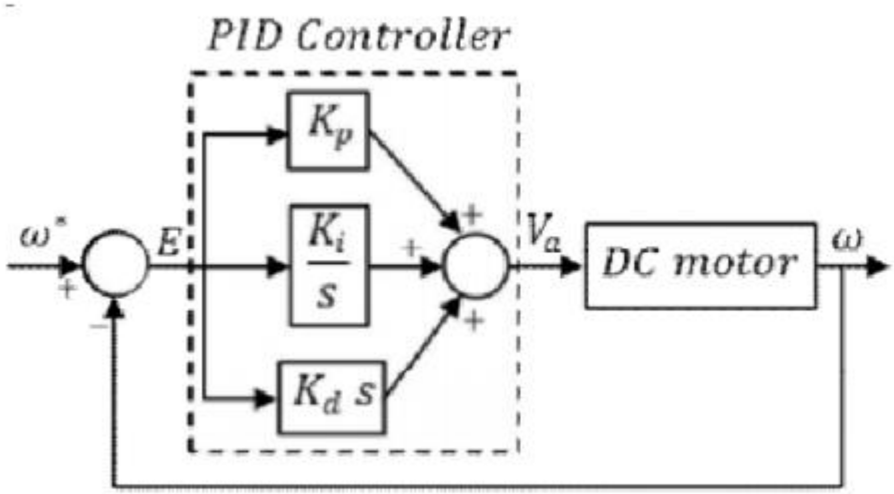

Figure 2: Block diagram of PID controller with DC motor

The PID controller transfer function is given by

$$\frac{Va(S)}{E(S)} = K_P + \frac{Ki}{S} + K_D s = K_p[1 + \frac{1}{Tis} + T_d s] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots . \quad 3$$

PID Controller tuning

The PID controllers are widely used for different control applications. The most critical step for applying the PID controller is the tuning of its parameters. The tuning process needs a lot of time and effort. In the worst case, the bad tuning leads to a poor performance of the controlled system. The goal of the tuning process is to determine the PID controller parameters that satisfy the performance specifications of the controlled system, such as the rising time, the maximum overshot, the settling time and the steady state error. However, it is difficult to obtain the desirable values of these requirements simultaneously. As we know that, larger values of proportional gain results in faster response while overshoot is increased. Therefore, an optimum tuning technique is of great importance.

Particle Swarm Optimization (PSO)

Particle swarm optimization is well suited for finding nonlinear problem. The tactic is predicated on the swarm behaviour such as birds finding food by flocking. Swarm and particles are mainly used to get the optimal values. Using PSO formula, particles are moved around within the search-space. The arrangements of the particles are directed by their individual best illustrious position within the search-space still as the whole swarms best illustrious position. Once higher locations are being discovered these can then come back to guide the movements of the swarm. The method is perennial and by doing therefore it's hoped, however not secure, that a suitable solution can eventually be exposed. Here during this technique a set of particles are place in n-dimensional search space with randomly selecting speed and position. The initial position of the particle is taken because the best position for the beginning and then the speed of the particle is updated supported the experience of different particles of the swarming population.

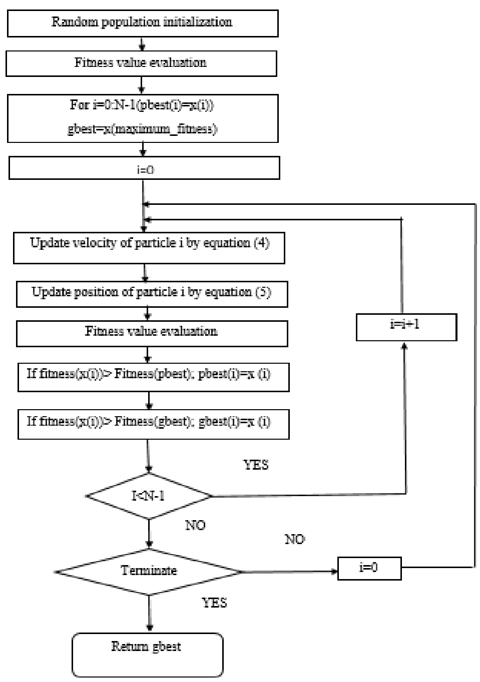

Figure 3: Flowchart of Particle swarm optimization

The rate of particle, adjusted per their individual flying expertise and therefore the alternative particles winged expertise. In PSO k-th particle is drawn as $(X_k = X_k;\ 1;\ X_k;\ 2;\ : X_k;\ d)$ within the n-dimensional house. the most effective previous worth of the k-th particle is recorded and drawn as:

$Pbest_k$ = ($Pbest_k$, 1, $Pbest_k$, 2, $Pbest_k$, d)

The index of best particle among all of the particles within the cluster is gbestn. The velocity for k-th particle is drawn as, Vk = (Vk. 1, Vk, 2, Vk, n).The changed position and velocity of every particle will be calculated using distance and current velocity and distance from Pbestk,n to gbest,n shown within the following formulas:

$$V_{k,b}(t+1) = W.V_{k,b}\,t + C1 * rand() * \left(pbest_{k,b} - X_{k,b}(t)\right) + C2 * rand() * (gbest_b - X_{k,b}(t))$$

$$X_{k,b}(t+1) = X_{k,b}t + V_{k,b}(t+1)\ \ k = 1,2 \ldots \ldots n;\ b = 1,2, \ldots \ldots d$$

In this study, for initialization purpose, particles considered are 50 with 110 iteration and 0 to 110 initial particles are bounded. C1 and C2 are the default values and set as 1.5. The initial value of ω is 0.9 and linearly decreased to 0.5 at some stage in iteration. Table III shows optimal PID parameters obtained using the PSO algorithm.

Table 3: Pid Parameters Based on Optimization

PID Gains	Parameters
K_P	2.6300
K_I	4.0504
K_D	0.9688

Fuzzy Logic Controller

Fuzzy logic is a control procedure established on a linguistic control plan, which is resulting from skilled information into an automatic control plan. The operation of a FLC is based on qualitative knowledge about the system being controlled. It doesn't need any challenging scientific design. While the others control scheme having very difficult calculation to produce a controlled plant, it only uses simple scientific calculation to simulate the skilled information. The requirement for the application of a FLC ascends mainly in situations where:(i)The explanation of the scientific process is existing only in word form, not in analytical form. (ii)It is not possible to recognize the parameters of the process with accuracy. (iii)The report of the process is too complex and it is more sensible to express its report in plain language words. (iv)The controlled scientific process has a fuzzy character. It is not possible to surely define these conditions. The block diagram of fuzzy logic system is shown in figure 4.

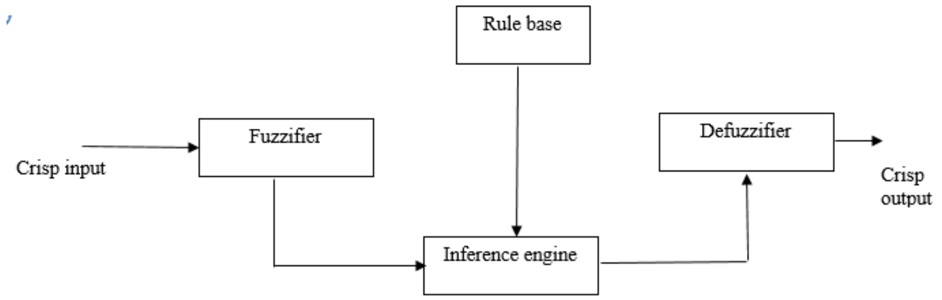

Figure 4: Fuzzy logic system

1. FUZZIFIER: The Fuzzifier module change the crisp inputs into fuzzy sets. A Fuzzy membership function has values which are defined by crisp variables such as negative small, negative big, positive big, positive small and zero etc. Where both one is defined by a gradually varying membership function. In Fuzzy set semantic all the imaginable values that a adjustable can assumed named motor rpm and Fuzzy set change the whole motor rpm. The forms of Fuzzy sets can be triangular, trapezoidal, singleton, Gaussian etc.
2. RULE BASE: The group of rules are called rule base. The rules are created using if then structure. If side is called situation and then side is called decision.
3. INFERENCE ENGINE: The inference engine are used to combine rules and fuzzifier output that is fuzzy sets and execute the output of inference engine that is degree of firing is directly send to defuzzifier.
4. DEFUZZIFIER: The defuzzifier collects fuzzy output sets from the inference engine and convert crisp output.

Fuzzy Logic Controller Design

Membership function: A membership function (MF) is normally expressed graphically and tends to illustrate how completely a crisp variable belongs to a fuzzy set. The membership function of error, change in error and output are shown in fig 5,6 and 7 respectively. The fuzzy rule base is as shown in Table 4..

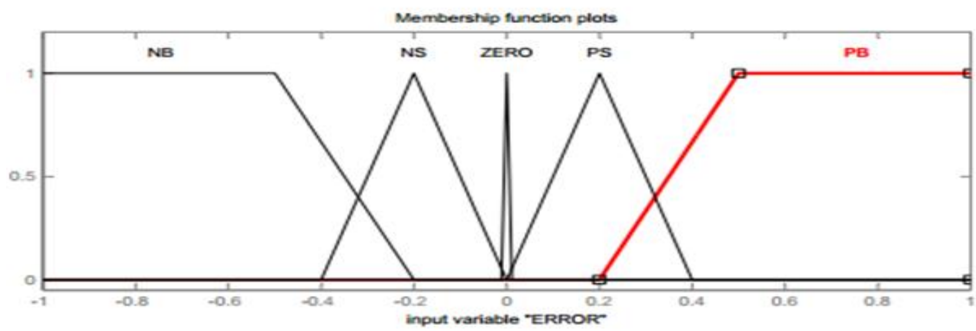

Figure 5: Error

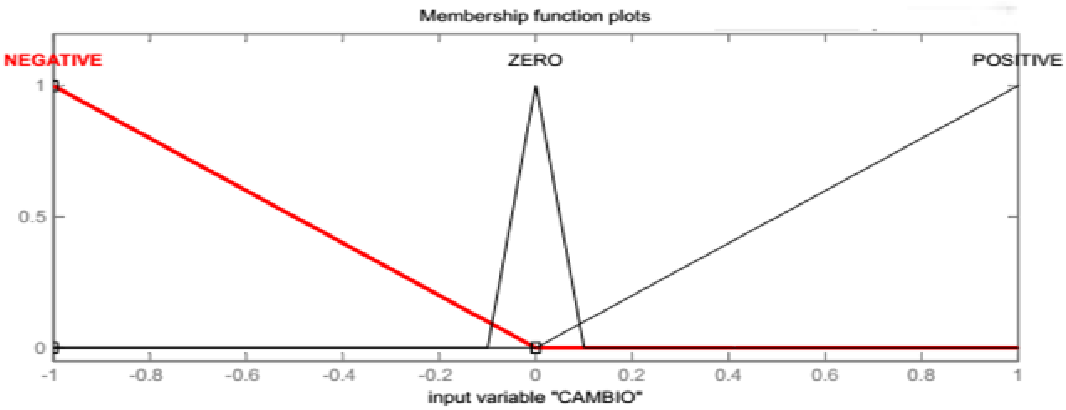

Figure 6: Change in error

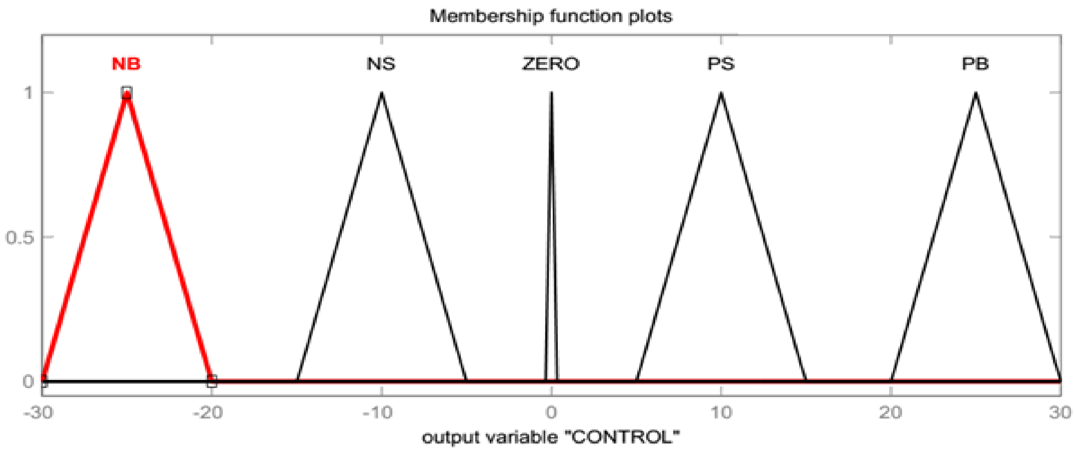

Figure 7: Output

Table 4: Fuzzy Logic Rules

Error /Change in error	Negative	Zero	Positive	None
PB	-	-	-	PB
NB	-	-	-	NB
ZERO	NS	-	PS	ZERO
PS	-	-	-	PS
NS	-	-	-	NS

Results

The Simulink diagram for conventional PID and Fuzzy PID controller is shown in figure 8. The response of the DC motor for various inputs are shown in figure 9 and 10 and figure 13 and 14 shows the pulsating load conditions.

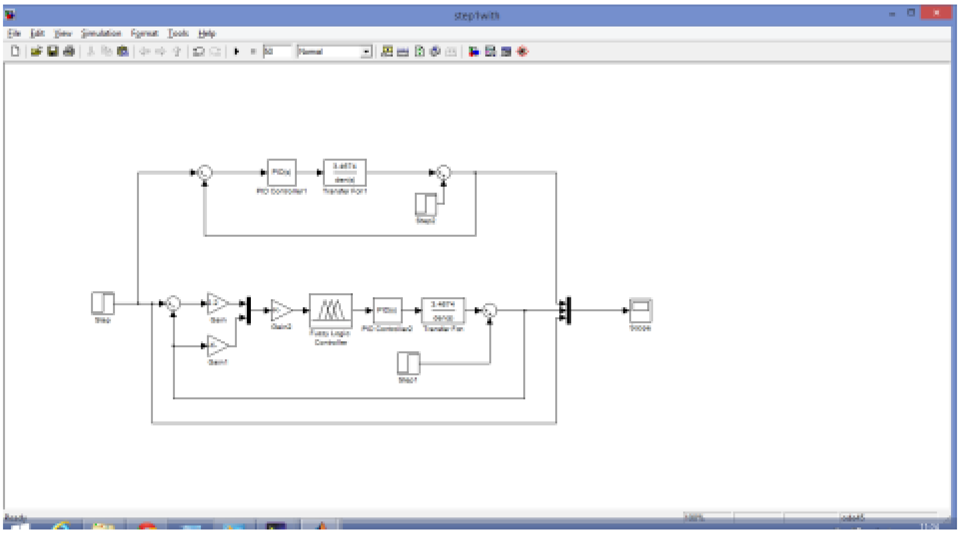

Figure 8: Simulink Diagram For PID And Fuzzy PID Controller

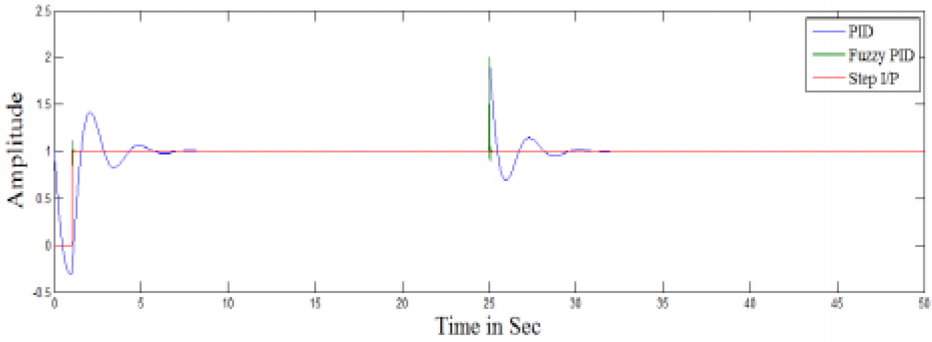

Figure 9: Response of The DC Motor (With Disturbance)

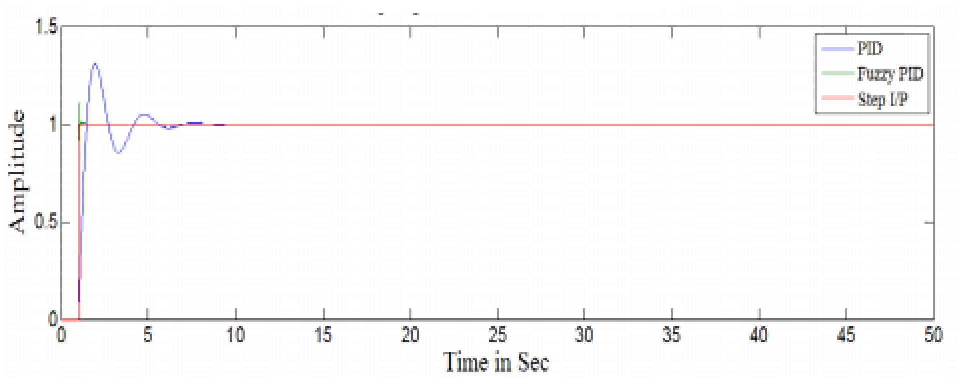

Figure 10: Response of The DC Motor (Without Disturbance)

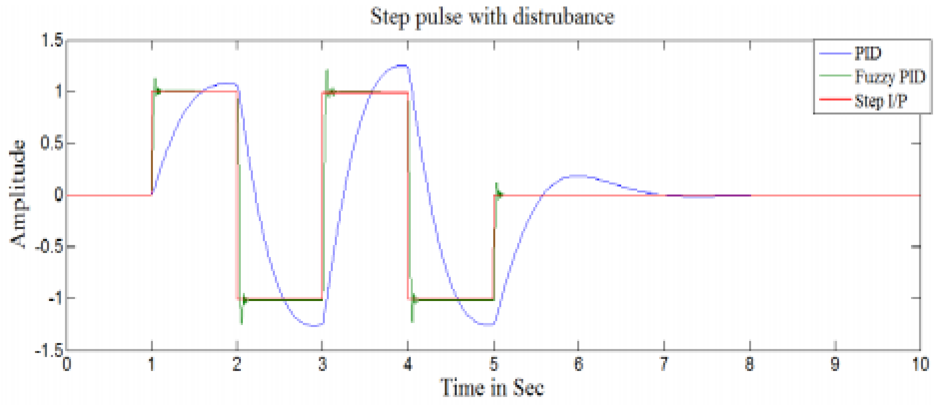

Figure 13: Response of the DC Motor at with disturbance

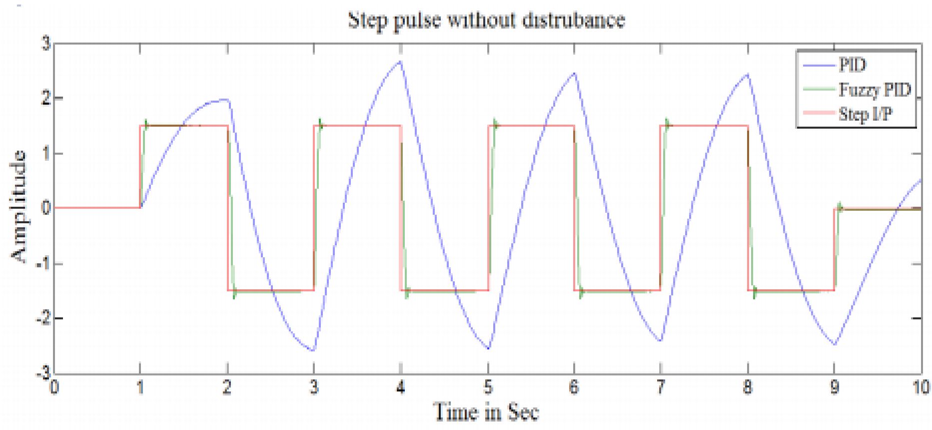

Figure 14: Response of the DC Motor for pulsating load condition

From the response it can be inferred that the FUZZY PID controller gives the better response then PID controller. The set point tracking response is quicker in FUZZY PID then the PID controller and the time domain specifications are very less in fuzzy PID controller design. Table 5 shows the time domain specifications of both controllers.

Table 5: Comparison of Time Domain Specifications

Parameter	Controller	
	PID	Fuzzy PID
Peak time (sec)	1.32	1.12
Rise time (sec)	1.4	1.01
Settling time (sec)	11	2.1
Peak overshoot (%)	13.2	11.2

Conclusion

In this paper we have studied different techniques for DC motor speed control. This paper introduces a fuzzy PID controller which consists of two inputs and one output and the controller is designed using MATLAB toolbox. The simulation results demonstrate that compared to PID controller, Fuzzy PID controller gives a better performance in both settling time and peak overshoot. It can easily be deduced that Fuzzy PID can maintain the speed at desired values irrespective of change of load. It can also be concluded that a superior performance is guaranteed than conventional controller.

References

[1] Zhang, N. Wang and S. Wang, "A developed method of tuning PID controllers with fuzzy rules for integrating process," Proceedings of the American Control Conference, Boston, 2004, pp. 1109-1114.

[2] K.H. Ang, G. Chong and Y. Li, "PID control system analysis, design and technology," IEEE transaction on Control System Technology, Vol.13, No.4, 2005, pp. 559-576.

[3] J.G. Ziegler, N. B. Nichols, Optimum Settings for Automatic Controllers, Trans. ASME, Vol. 64, No. 8, pp. 759-768, 1942.

[4] P. M. Meshram, R. G. Kanojiya, Tuning of PID Controller Using Ziegler-Nichols Method for Speed Control of DC motor, in Proceedings book IEEE International Conference On Advances In Engineering, Science And Management (ICAESM), pp. 117-122, Mar.2012.

[5] N. P. Adhikari, M. Choubey, R. Singh, Dc Motor Control Using Ziegler Nichols and Genetic Algorithm Technique, International Journal of Electrical, Electronics and Computer Engineering, Vol. 1, pp.33-36, 2012.

[6] B. Allaoua, A. Laoufi, B. Gasbaoui, A. Nasri, A. Abderrahmani, Intelligent Controller Design for DC Motor Speed Control based on Fuzzy Logic-Genetic Algorithms Optimization, Leonardo Journal of Sciences, Issue 13, p. 90-102, July-Dec. 2008.

[7] A. R. Singh, V. K. Giri, Design and Analysis of DC Motor Speed Control by GA Based Tuning of Fuzzy Logic Controller, International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 5, July 2012.

[8] E. Gowthaman, C. D. Balaji, Self Tuned PID Based Speed Control of PMDC Drive, in Proceedings book International Multi-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), pp. 686-692, Mar. 2013.

[9] S. J. Bassi, M. K. Mishra, E. E. Omizegba, Automatic Tuning of Proportional–Integral–Derivative (PID) Controller Using Particle Swarm Optimization (PSO) Algorithm, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.4, pp. 24-34, Oct. 2011.

[10] H. X. Li and S. K. Tso, "Quantitative design and analysis of Fuzzy Proportional-Integral-Derivative Control- a Step towards Auto tuning", International journal of system science, Vol.31, No.5, 2000, pp.545-553.

[11] Thana Pattaradej, GuanrongChen and pitikhate Sooraksa, "Design and Implementation of Fuzzy PID Control of a bicycle robot" Integrated computer-aided engineering, Vol.9, No.4, 2002.

[12] Weiming Tang, Guanrong Chen and Rongde Lu, “A Modified Fuzzy PI Controller for a Flexible-joint Robot Arm with Uncertainties”, Fuzzy Set and System, 118 (2001) 109-119