© September 2019 | IJIRT | Volume 6 Issue 4 | ISSN: 2349-6002

Speed Control of Brushless Dc Motor by Using PID and Fuzzy Logic Controller

N.Arjuna Rao[1], Dr.Ch.Ravi Kumar[2]
[1]*PG Scholar, Dept. Of EEE, University college of Engineering & Technology, A.N.U, Guntur, A.P*
[2]*Asst.Professor, Dept. of EEE, University College of Engineering & Technology, A.N.U, Guntur, A.P*

Abstract- **Brushless DC (BLDC) motors are widely used for many industrial applications because of their reliability, high efficiency, high starting torque, and less electrical noise. During the past decades, the process control techniques in industry have made great advances. Many control methods such as conventional control, neural control, and fuzzy control have been studied. In this the performances of BLDC motor have been evaluated with conventional PID control and fuzzy logic control methods. PID controller has been widely used in the industry due to its simple structure and robust performance. But it has been quite difficult to tune properly the gains of high order, time delay and non linearties. Over the years many heuristic methods have been proposed to tune the PID controller. The first method used to find optimal or near optimal PID parameters with ZIGLER AND NICHOLAS and TYREUS-LUYBEN. It is often difficult to find optimal or near optimal PID parameters with ZIGLER AND NICHOLAS, TYREUS-LUYBEN formulas in many industrial plants. Due to new features, PSO and FUZZY LOGIC techniques have been included to improve the controller performance characteristic than other methods. ZIGLER-NICHOLAS, TYREUS-LUYBEN and PSO and FUZZY LOGIC CONTROLLERS are implemented for the tuning methods to control the BLDC motor in MATLAB/SIMULINK software. The performance characteristics of the BLDC motor are compared with respective these four tuning methods**

Index terms- Bldc Motor, Pid Controller, Tuning of Pid Controller, Z-N Method, T-L Method Pso Method, Fuzzy Control Method, Simulink Model

I. INTRODUCTION

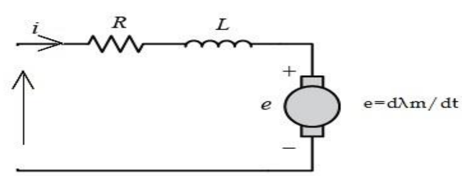
Brushless motors are widely applied in areas which needs high performance drive such as high efficacy, long operating life, noise-less operation, high speed range and high torque. The controlling of this motor is done by using PID Controller.

In this thesis ZIGLER- NICHOLAS, TYREUS-LUYBEN, PSO & FUZZY LOGIC CONTROL methods are implemented for the tuning of PID controller to control the BLDC motor. MATLAB/SIMULINK software is used for the implementation of this project. The performance characteristics of the BLDC motor are compared with respective these four tuning methods.

II. BRUSHLESS DC MOTOR

BLDC MOTOR: Permanent magnet DC motors use mechanical commutators and brushes to achieve the commutation. But BLDC motors adopt Hall Effect sensors in place of mechanical commutators and brushes.

The stators of BLDC motors are the coils and the rotors are the permanent magnets. The stators develop the magnetic fields to make the rotor rotating. Hall effect sensors detect the rotor position as the commutating signals. Hence, BLDC motor use permanent magnets instead of coils in the armature and so don’t need brushes. Here a three phase and eight pole BLDC motor is studied.

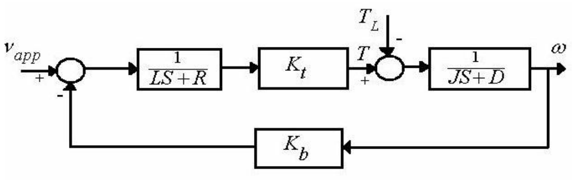


$$v_{app}(t) = L\frac{di(t)}{dt} + R \cdot i(t) + v_{emf}$$

$$v_{emf} = k_b \cdot \omega(t)$$

$$T(t) = k_t \cdot i(t)$$

$$T(t) = J\frac{d\omega(t)}{dt} + D \cdot \omega(t)$$



Working of BLDC Motor

The trick of operation in BLDC motors is the **Hall sensor** that is attached to the stator. It faces the magnets perpendicularly and can distinguish if the North or South pole is in front of it. The following image shows this Hall senor. The photo is taken from a PC fan (yes, PC fans do have BLDCs!): The Hall sensor is this little component under the right electromagnet. When it senses the South pole, it keeps the coils turned off. When the sensor senses no magnetic field (or could be also the South Pole), then it turns on the coils. The coils have both the same magnetic polarity which is north. So they pull the opposite pole and torque is then created.

If you put a probe to the Hall sensor and watch the signal, then you will discover that during a full rotation of the rotor, the Hall sensor is two times HIGH and two times LOW. The waveform on oscilloscope would be like this one: Yet another great advantage for the brushless motors. This very signal that is used to control the coils, can be used as is for measuring the speed of the motor! It can also be used to see if the motor is functional or not! Actually, this signal is exactly the one that comes out from the third wire from the PC fans that have 3 (or 4 wires)! These fans do not have any extra circuitry to measure the speed of the motor. They use the signal from the Hall sensor. Each revolution will generate 2 pulses. With a simple frequency measuring circuitry, someone can measure precisely the rpm of the motor.

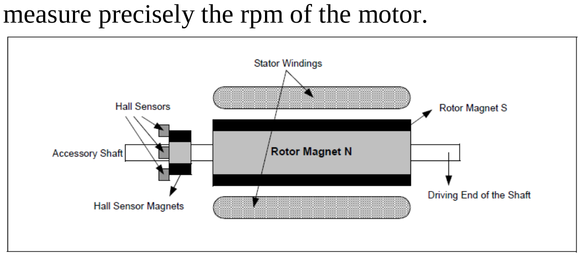


Fig: 2.1 Transverse Section of BLDC Motor

III.PID CONTROLLER

This introduction will show you the characteristics of the each of proportional (P), the integral (I), and the derivative (D) controls, and how to use them to obtain a desired response. Consider the following unity feedback system

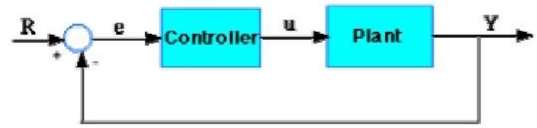


Fig. 3.1 Simple block diagram of PID controller

Plant: A system to be controlled Controller: Provides the excitation for the plant; Designed to control the overall system behavior

The three-term controller

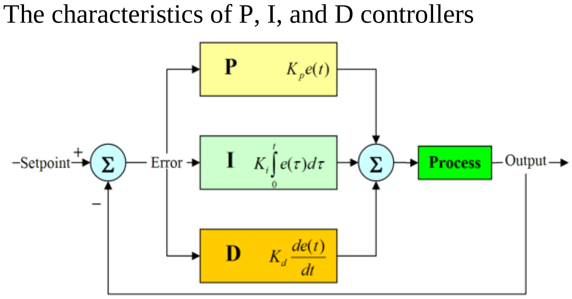
The transfer function of the PID controller looks like the following:

- Kp = Proportional gain
- KI = Integral gain
- Kd = Derivative gain

First, let's take a look at how the PID controller works in a closed-loop system using the schematic shown above. The variable (e) represents the tracking error, the difference between the desired input value (R) and the actual output (Y). This error signal (e) will be sent to the PID controller, and the controller computes both the derivative and the integral of this error signal. The signal (u) just past the controller is now equal to the proportional gain (Kp) times the magnitude of the error plus the integral gain (Ki) times the integral of the error plus the derivative gain (Kd) times the derivative of the error

This signal (u) will be sent to the plant, and the new output (Y) will be obtained. This new output (Y) will be sent back to the sensor again to find the new error signal (e). The controller takes this new error signal and computes its derivative and its integral again. This process goes on and on.

The characteristics of P, I, and D controllers



3.1 Tuning of Pid Controller Using Zigeler Nicholas Method

From the Frequency Response Test (or Relay Test), we obtained the ultimate gain *Ku* and *Tu.* From these values, we obtain the following table

Controller	K_p	K_i	K_d
PI	$\frac{2K_u}{5}$	$\frac{K_u}{2T_u}$	0
PID	$\frac{3K_u}{5}$	$\frac{6K_u}{5T_u}$	$\frac{3K_uT_u}{40}$

Table. 3.1 ZN method parameters table for Frequency response

These were obtained analytically because we actually know the system exactly. They are close enough to the approximations which don't require exact knowledge to obtain.

3.2. Tuning of Pid Controller Using Tyreus-Luyben Method

The Tyreus-Luyben procedure is quite similar to the Ziegler–Nichols method but the final controller settings are different. Also this method only proposes settings for PI and PID controllers. These settings that are based on ultimate gain and period are given in Table. Like Z-N method this method is time consuming and forces the system to margin if instability. Many other algorithms have been proposed to solve these problems by obtaining critical data (ultimate gain and frequency) under more acceptable conditions

Controller	k_c	τ_I	τ_D
PI	$k_{cu}/3.2$	$2.2P_u$	-
PID	$k_{cu}/3.2$	$2.2P_u$	$P_u/6.3$

Table 3.2:TL method parameters table

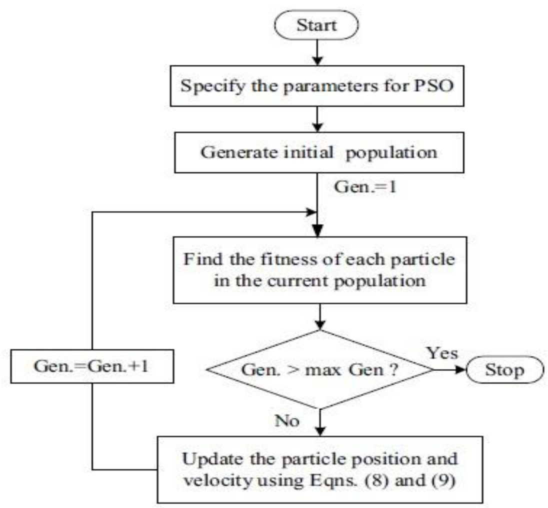
TUNING OF PID USING PSO-BASED OPTIMIZATION

PSO is optimization algorithm based on evolutionary computation technique. The basic PSO is developed from research on swarm such as fish schooling and bird flocking After it was firstly introduced in 1995 a modified PSO was then introduced in 1998 to improve the performance of the original PSO. A new parameter called inertia weight is added This is a commonly used PSO where inertia weight is linearly decreasing during iteration in addition to another common type of PSO which is reported by Clerc The later is the one used in this paper. In PSO, instead of using genetic operators, individuals called as particles are "evolved" by cooperation and competition among themselves through generations. A particle represents a potential solution to a problem. Each particle adjusts its flying according to its own flying experience and its companion flying experience. Each particle is treated as a point in a D-dimensional space. The ith particle is represented as XI= (xi1, xi2 ..xiD). The best previous position (giving the minimum fitness value) of any particle is recorded and represented as PI= (pi1,pi2,…,piD), this is called p best. The index of the best particle among all particles in the population is represented by the symbol g, called as g best. The velocity for the particle i is represented as VI= (vi1, vi2… viD). The particles are updated according to the following equations:

$$v_{id}^{n+1} = w.v_{id}^{n} + c_1.rand().(\ p_{id}^{n} - x_{id}^{n}\ ) + c_2.rand().(\ p_{gd}^{n} - x_{id}^{n}\ )$$

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1}$$

Flow Chart of PSO:



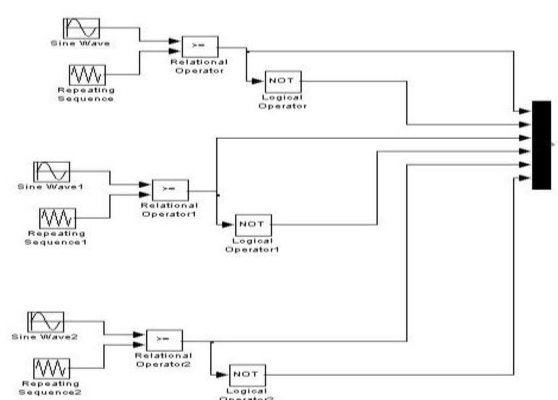
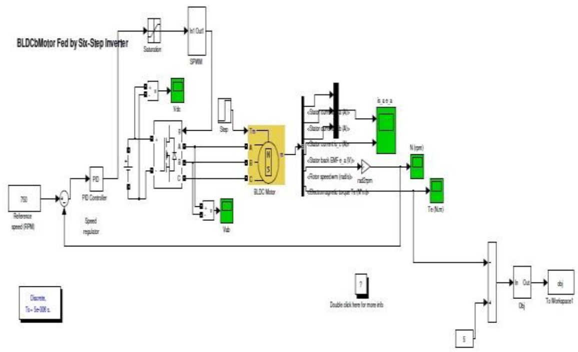


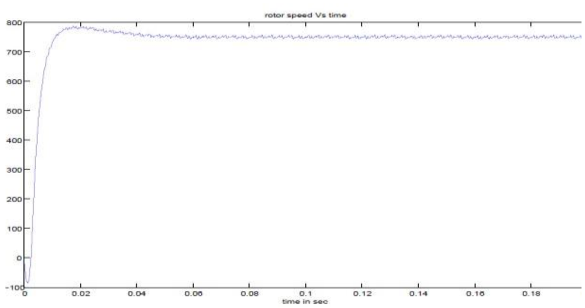
Fig.3.2 Simulink model of spwm

IV. SIMULINK MAIN MODEL

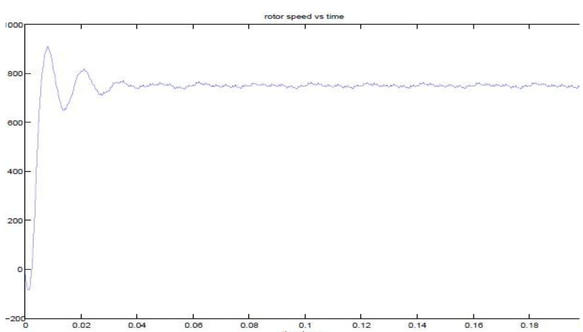
4.1 BLDC Motor Fed by Six Step Inverter



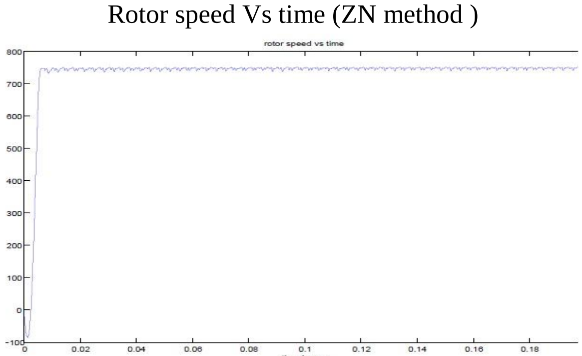
V. SIMULATION RESULTS



Rotator speed Vs time (TL-method)



Rotor speed Vs time (ZN method)



Rotor speed vs. time (PSO method)

S	APPROACH TUNING	LUYBENT TUNING	ZIGLERS TUNING
[P I D]	{287.2521,190.0294,0.0876}	{7.506,200.695,0.02025}	{10.008,1177.417,0.02127}
RISE TIME(s) [Tr]	0.003	0.012	0.006
PEAK TIME(s) [Tp]	0.014	0.017	0.009
PEAK OVER SHOOT [Mp]	0.00038	0.01033	0.01867
SETTLING TIME(s) [Ts]	0.0141	0.051	0.036

fig Comparison of PSO,TYREUS-LUYBENT & NICHOLAS-ZIGLER

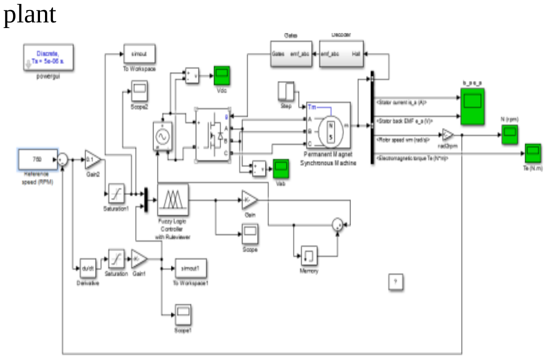
VI. FUZZY CONTROLLER METHOD

Fuzzification: its defined as the process of converting the input analog signals to values can be compared with rules in rule base.

The inference mechanism: it's defined as the mechanism which decides the control rules related at the current time and then make the convenient input to the system.

Rule base : it's defined as the linguistic logical rules using (IF-THEN) conditions to get the desired output.

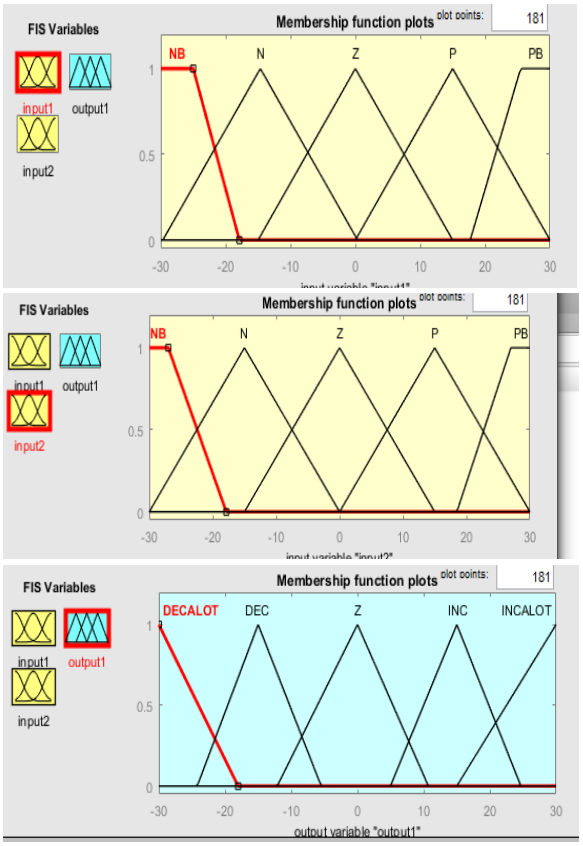
Defuzzification: its defined as the process of converting the result of applying rules from fuzzy logical form into analog form to be entered to the plant



VII.FUZZY SIMULATION RESULTS

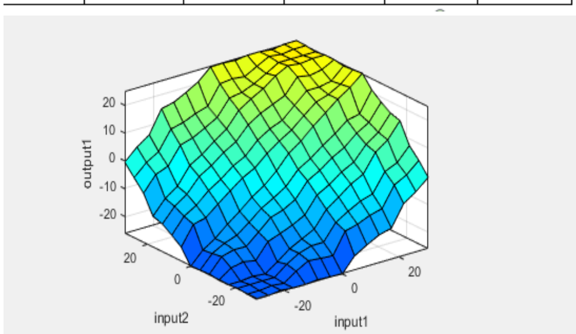
At the start, the speed was zero, and at 0.0123sec the speed increased to 750 RPM, at 0.0145 sec the speed increased to 750 rpm at 0.1 sec a load of
10 % is added which makes undershoot at speed 0.70% the speed comes back to its value 750 rpm after 0.0102 sec.

As we say before we can use fuzzy logic controller to improve the performance of the system, and in figure7, we can see the response of the motor to the controller the rise time is 0.0025



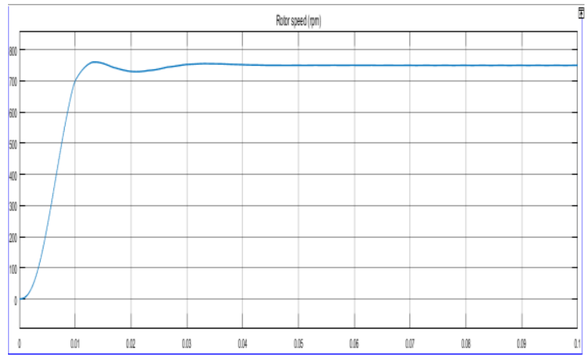
FUZZY CONTROLLER RULE BASE:

E / CE	NB	N	ZERO	P	PB
NB	BIGDEC	BIGDEC	BIGDEC	DEC	ZERO
N	BIGDEC	BIGDEC	DEC	ZERO	INC
Z	BIGDEC	DEC	ZERO	INC	BIGINC
P	DEC	ZERO	INC	BIGINC	BIGINC
PB	ZERO	INC	BIGINC	BIGINC	BIGINC



FUZZY SURFACE

SPEED OF BLDC MOTOR WITH FUZZY CONTROLLER



Comparison of pid and fuzzy controller

PARAMETERS	PSO APPROACH TUNING	TYREUS-LUYBENT TUNING	NICHOLAS-ZIGLERS TUNING	FUZZY LOGIC CONTROLLER
[P I D]	**{287.2521,190.0294,0.0876}**	**{7.506,200.695,0.02025}**	**{10.008,1177.417,0.02127}**	
RISE TIME(s) [Tr]	0.003	0.012	0.006	0.002
PEAK TIME(s) [Tp]	0.014	0.017	0.009	0.012
PEAK OVER SHOOT [Mp]	0.00038	0.01033	0.01867	0.00036
SETTLING TIME(s) [Ts]	0.0141	0.051	0.036	0.0139

VIII.CONCLUSIONS

In this thesis conventional methods of tuning the PID controller and FUZZY controller have been used. These methods are Particle swarm optimization method, Tyreus-Luyben method and ziglers-Nicholas method and fuzzy logic control method

By comparing these four techniques the dynamic response of the BLDC motor due to PSO technique is found better than that of Tyreus- luyben and Ziglers-Nicholas tuning of PID controller. Briefly speaking we can say that PID controller is a simple controller with a simple tuning method but with a moderate response and performance. Fuzzy controller is more complicated controller but with a good and more stable performance

REFERENCES

[1] Ahmed, A. M., Ali-Eldin, A., Elksasy, M. S., & Areed, F. F. (2015). Brushless DC motor speed control using both PI controller and fuzzy PI controller. International Journal of Computer Applications, 109(10), 29-35.

[2] Mustafa, G. Y., Ali, A. T., Bashier, E., & Elrahman, M. F. (2013). Neuro-fuzzy controller

design for a dc motor drive. University Of Khartoum Engineering Journal, 3(1).

[3] Tiwari, N., RITEE, R. C., & Diwan, R. Speed Control of Brushless DC Motor using Fuzzy and Neuro Fuzzy.

[4] Mosavi, M. R., Rahmati, A., Khoshsaadat, A., & Elektrotechniczny, P. (2012). Design of efficient adaptive neuro-fuzzy controller based on supervisory learning capable for speed and torque control of BLDC motor. PRZEGLĄD ELEKTROTECHNICZNY (Electrical Review), R, 88.

[5] Yashoda, M., & Sekhar, O. C. (2016). Design and Analysis of ANFIS based BLDC Motor. Indian Journal of Science and Technology, 9(35).

[6] Arulmozhiyal, R., & Kandiban, R. (2012, July). An intelligent speed controller for Brushless DC motor. In Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on (pp. 16-21). IEEE.

[7] Navaneethakkannan, C., & Sudha, M. (2016). Analysis and Implementation of ANFIS-based Rotor Position Controller for BLDC Motors. Journal of Power Electronics, 16(2), 564-571.