Implementation of Fuzzy Tuned PID Controller Using PIC18F2520 for Speed Control of BLDC Motor

Angeti Jival[1*] Iliya Tizhe Thuku[2] Allo Iliya Alhassan[3]

[1.] Department of Electrical and Electronic Engineering, Modibbo Adama University of Technology Yola, Nigeria

[2.] Department of Electrical and Electronic Engineering, Modibbo Adama University of Technology Yola, Nigeria

[3.] Department of Electrical and Electronic Engineering, Federal Polytechnic Mubi, Adamawa state, Nigeria

* E-mail of the corresponding author : angetijival@rocketmail.com

Abstract

Brushless direct current (BLDC) Motor finds widespread application in Engineering such as robotics, aerospace, biomedical systems, conveyor systems, electric vehicles etc. This is due to their superior electrical and mechanical characteristics. This research work focus on implementation of a Fuzzy-Tuned PID controller around PIC18F2520 microcontroller, for the speed control of a Brushless direct current (BLDC) motor. Fuzzy logic is used to update the gains of the PID controller online. The results prove the superiority of the Fuzzy-tuned PID controller over its conventional counterpart as it has lower steady state errors and more robust to parameter variations and load disturbances.

Keywords: BLDC motor, Fuzzy Tuned PID Controller, PIC18F252 Microcontroller, Speed Control

DOI: 10.7176/CTI/8-08

1.0 Introduction

Brushless Direct Current Motors (BLDC) are gradually taken over the usage of DC Motors [1]. Brushless, as the name implies, they have neither brushes nor commutators. These motors find application in Medical Equipment, Electric Vehicles, Air craft, Home appliances, computers, Robotics, and so on [2]. One unique feature that make BLDC motors robust over the conventional DC motors is that switching of current in the armature coil is done with the help of electronic circuit, which reduces mechanical losses and improves efficiency [3]. Whereas conventional DC motors uses brushes and commutators to performed commutation. In fact, these motors have several advantages over conventional DC motors which include: better torque versus speed characteristic, higher efficiency, better dynamic response due to low rotor inertia, lesser maintenance requirement, low electric noise and high speed range (up to 10,000 rpm) [4]. Additionally, the ratio of torque delivered to the size of the motor is higher, making it suitable in application where space and weight are critical factors.

2.0 Hardware System Description

The system shown in figure1: comprises of keyboard, Display unit, Microcontroller unit, driver's unit and BLDC motor. The candidate controller was design and implemented around PIC18F2520 microcontroller using fuzzy rule specified in the Mamdani type of fuzzy inference system. Desired speed is inputted through the keyboard, and the microcontroller displays both the desired and the actual speed on Liquid Crystal Display (LDC). Fuzzy logic block shown in Figure 1: Simply takes in the error "e" and rate of change-in-error "ec" as the input variable and makes use of the fuzzy set to modify PID parameters on-line. To get the actual speed of the motor, one of the Hall Effect Sensor embedded inside the Motor is used as a feedback to the microcontroller. With the designed fuzzy rules, the PID controller is tune online and the pulse width modulated (PWM) signals are generated based on the Hall sensor inputs. The microcontroller provides the necessary gate signals for the switching of the driver unit, which in turn energies the respective windings of the three phase BLDC motor. In this case the speed regulation is achieved by varying the duty cycles. When the duty cycle of PWM reduces within the sequences, the average voltage supplied to the stator winding reduces, thus reducing the speed of the motor and when the duty cycle increases the average voltage supply increases thereby increasing the speed of the motor hence, the speed of the motor is controlled as desired.

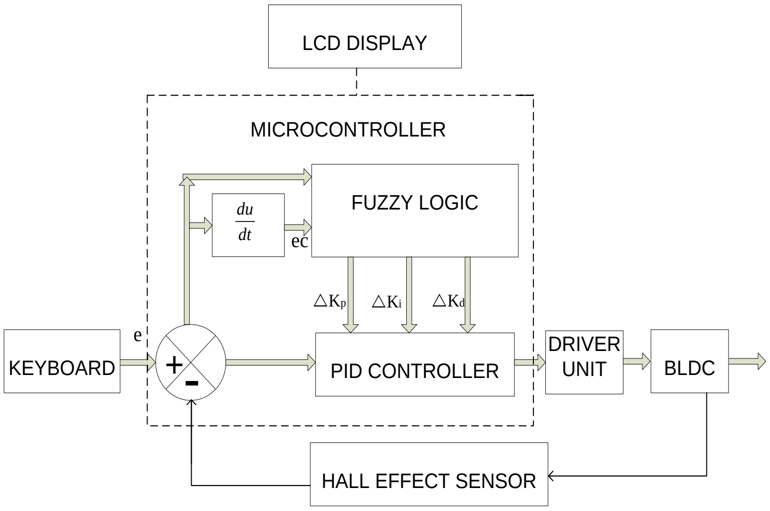


Figure 1: system block diagram of BLDC motor controller

3.0 Design of PID controller

The transfer function of the analogue PID controller is given as:

$$G(s) = kp + \frac{ki}{S} + kdS = \frac{kpS + Ki + KdS^2}{S} \tag{1}$$

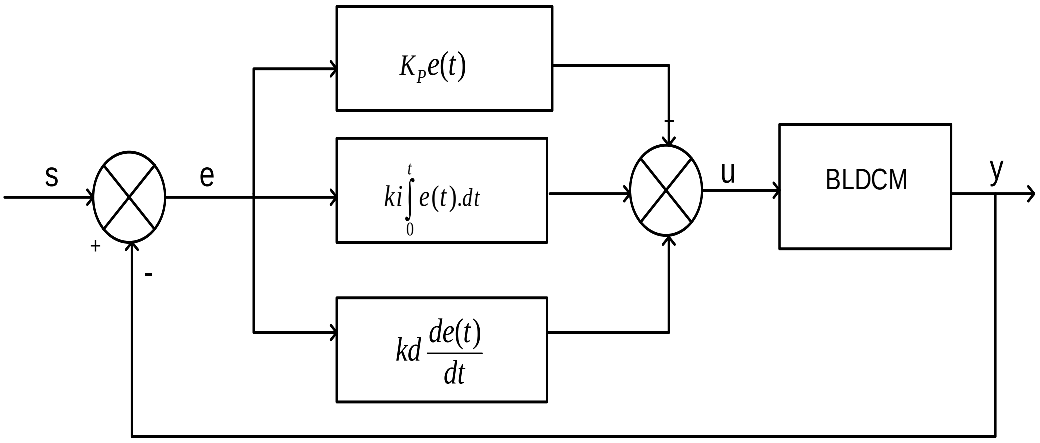


Figure 2: Block diagram of a conventional PID controller.

For the PID control presented in Figure 2: The output of the controller to the motor u is [5]:

$$u(t) = kpe(t) + ki\int_0^t e(t)dt + kd\frac{de(t)}{dt} \tag{2}$$

$$\frac{de(t)}{dt} = \frac{e(k) - e(k-1)}{Ts} \tag{3}$$

$$\int_0^t e(t)dt = \sum_{i=0}^{t} \mathrm{e(i)} \tag{4}$$

$$u(k) = kpe(k) + kiTs\sum_{i=0}^{k} e(i) + \frac{kd}{Ts}\left[e(k) - e(k-1)\right] \tag{5}$$

$$kp = KP \quad kiTs = KI \quad \text{and} \quad \frac{kd}{Ts} = KD \tag{6}$$

$$U(k) = KPe(k) + KI\sum_{i=0}^{k} e(i) + KD\left[e(k) - e(k-1)\right] \tag{7}$$

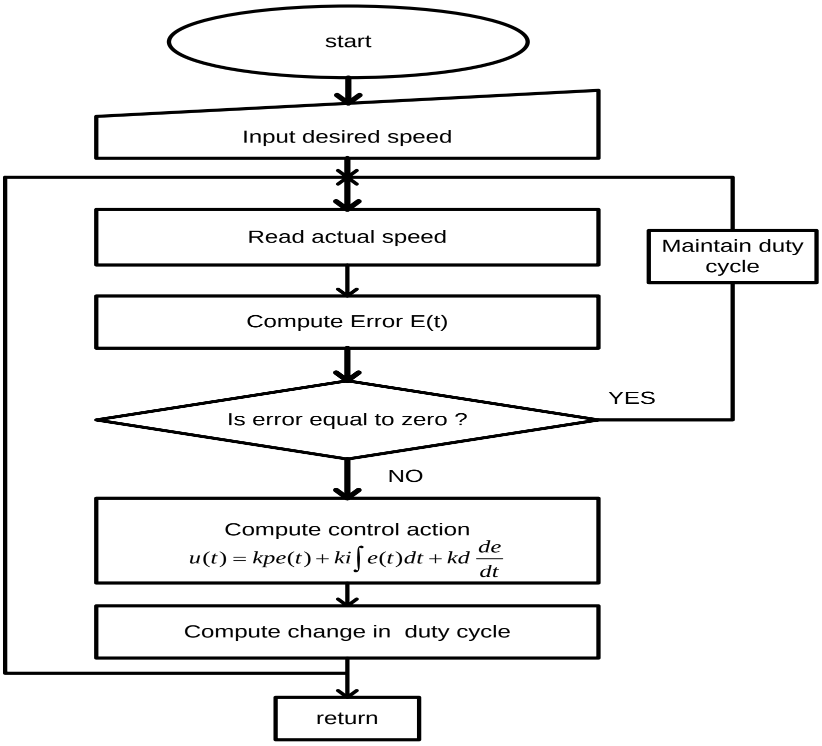


Figure 3: Conventional PID controller flow chart

The flow chart of figure 3 shows the control algorithms of a conventional PID controller used in this research work. The control signal u computes the control action and change the duty cycle based on the input signal obtained from the reference point.

4.0 Fuzzy-PID Controller Design

4.1 *Fuzzification*

This performs the conversion of the point-wise (Crip) value of the process variable into fuzzy set in order to make it compatible with the fuzzy set representation of the process variable in the rule antecedent [6]. This conversion is based on the membership function so assigned and shown in figure 4

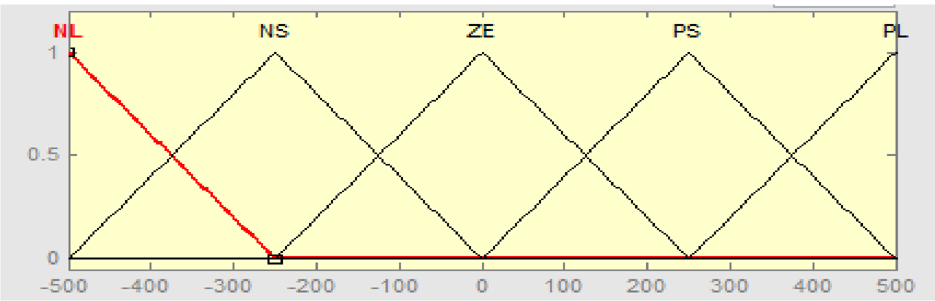
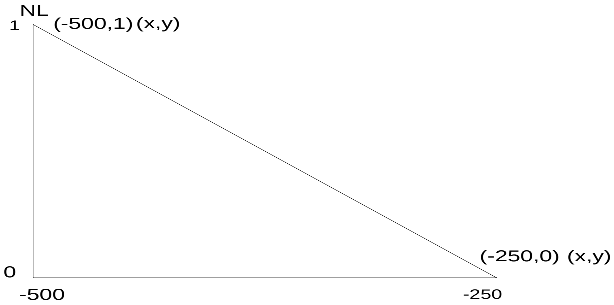


Figure 4: Input membership function for change in error ('CE')



From figure 4 above each membership function was resolved using the equation of a straight line.

$$For -500 \le x \le -250$$

$$\frac{y-y_1}{x-x_1} = \frac{y_2-y_1}{x_2-x_1}$$

$$\frac{y-1}{x-(-500)} = \frac{1-0}{-500-(-250)}$$

$$\frac{y-1}{x+500} = \frac{1}{-250}$$

$$y = \frac{-x}{250} - 1$$

$$NL = \begin{cases} \frac{-x}{250} - 1; -500 \le ce \le -250 \end{cases}$$

4.1.1 Data base

Fuzzy Tune PID controller is designed having [-3000 3000] universe of discourse to cover the maximum error that could occur in the system. The input variable (error and change in error) is partitioned into fuzzy sets called 'Negative large' (NL), 'Negative Small' (NS), 'Zero' (ZE), 'Positive Small' (PS) and 'Positive Large' (PL) with domains [-3000 -2500] [-3000 0] [-1500 1500] [0 3000] [1500 3000] respectively. Triangular membership function is adopted in the procedure. For the output variable, [0.1 0.7] universe of discourse is used with seven triangular membership functions called 'Positive Very Small', 'Positive Small', 'Positive Medium Small', 'Positive Medium', 'Positive Medium Large', 'Positive Large' and 'Positive Very Large' ranged at [0.1 0.2] [0.1 0.3] [0.2 0.4] [0.3 0.5] [0.4 0.6] [0.5 0.7] [0.6 0.7] respectively as shown in figure 5.

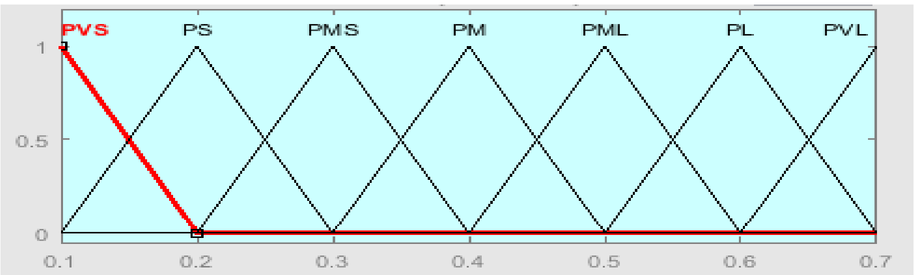


Figure 5: Output membership functions Kp, Ki and Kd

4.1.2 Rule base

This is the stage where fuzzy control rules are framed to attain the finest performance of fuzzy Tuned PID controller. The IF part of the rule refers to the antecedent and the THEN part refers to the consequent Since the output of the fuzzy logic is to tuned the gains of the PID KP, KI and KD through the output membership function called 'Positive Very Small', 'Positive Small', 'Positive Medium Small', 'Positive Medium', 'Positive Medium Large', 'Positive Large' and 'Positive Very Large' as error and change in error varies between fuzzy sets : 'Positive Very Small', 'Positive Small', 'Positive Medium Small', 'Positive Medium', 'Positive Medium Large', 'Positive Large', and 'Positive Very Large' respectively. Few among the golden rules are formulated for the fuzzy logic are as follows:

If E is NL and CE is NL Then KP is PVL, KI is PM and KD is PVS

If E is NL and CE is NS Then Kp is PVL, KI is PM and KD is PMS

If E is NL and CE is ZE Then Kp is PVL, KI is PM and KD is PM

If E is NL and CE is PS Then Kp is PVL, KI is PM and KD is PL

If E is NL and CE is PL Then Kp is PVL, KI is PM and KD is PVL

If E is NS and CE is NL Then Kp is PML, KI is PMS and KD is PMS

If E is NS and CE is NS Then Kp is PML, KI is PMS and KD is PML

If E is NS and CE is ZE Then Kp is PS, KI is PMS and KD is PL

If E is NS and CE is PS Then Kp is PL, KI is PML and KD is PVL

If E is NS and CE is PL Then Kp is PVL, KI is PMS and KD is PVL

4.1.3 Defuzzification

This process involves the conversion of all the aggregated fuzzy set into a single Crip value corresponding to the require control action.

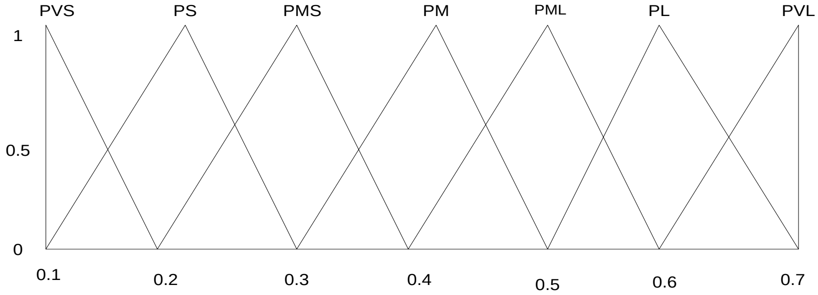


Figure 6: Output membership function of KP, KI and KD

Before the Defuzzification process, the implication and aggregation need to be done [7].

Figure 7 shows implication and aggregation of the output membership function ‘positive small’.

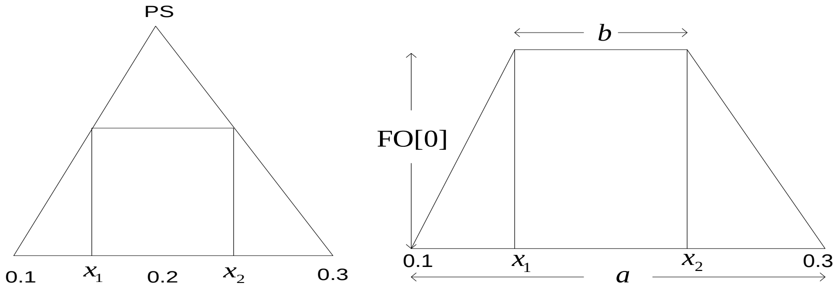


Figure 7: implication and aggregation output membership function ‘positive small’

For the ‘positive small’ MF the area is given as: $A_{ps} = \frac{1}{2}(a+b)\,\text{Fo}[1]$

$$PS = \begin{cases} 10x_1 - 1; 0.1 \le x_1 \le 0.2 \\ -10x_2 + 3; 0.2 \le x_2 \le 0.3 \end{cases}$$

From rule (8) above

If E is NS and CE is ZE Then KP is PS

$PS = 10x - 1$ For $0.1 \le x \le 0.2$

$PS = -10x + 3$ For $0.2 \le x \le 0.3$

Therefore $10x_1 - 1 = Fo[1]$

$$10x_1 = Fo[1] + 1$$

$$x_1 = \frac{Fo[1]+1}{10}$$

$$x_2 = \frac{Fo[1]+3}{-10}$$

$Dfz\_ps = \dfrac{A_{ps} \times c_{ps}}{A_{ps}}$ Where: C_{ps} Is the Centre of 'positive Small' and $A_{ps} = \dfrac{1}{2}(a+b)\,\mathrm{Fo}[1]$ is the area of 'positive Small'

5.0 Hardware Implementation

The implementation of this research work was carried out in two major phases the hardware and the software aspect respectively. The hardware section comprised of implementation of sub units making up the overall functionality of the system. These sub units include: driver units, control and display unit.

5.1 *Driver Unit*

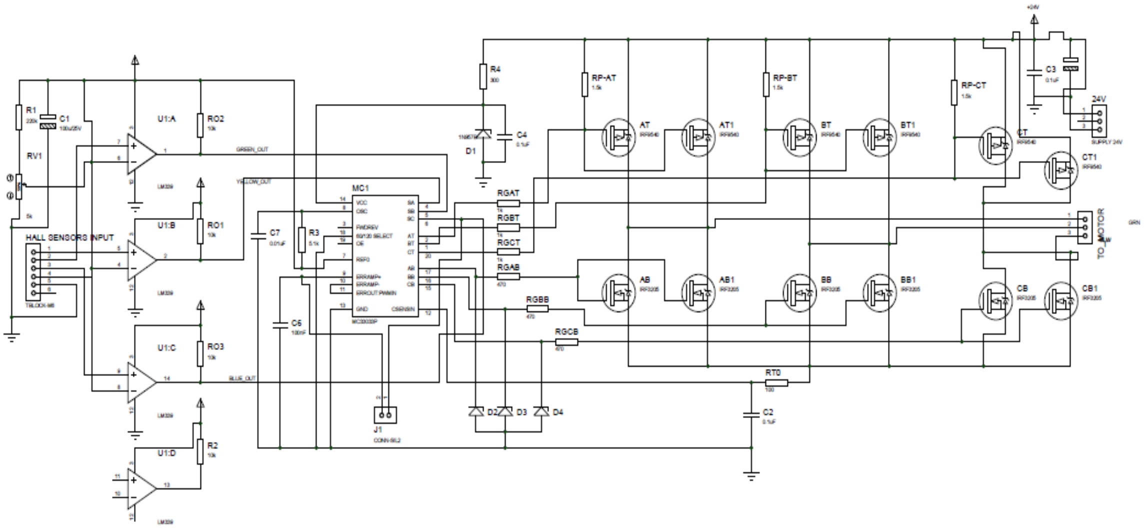


Figure 8: Complete driver circuit of the system.

The driver unit is responsible for providing proper commutation sequence for the BLDC motor to run. Figure 9 show the complete driver units of the system. The most essential component used to design this unit is MC33033 Motor Driver. The MC33033 play an important role in the operation of this unit in the sense that it has three open collector top drive output (PIN 1,2 and 20) and three totem pole bottom drive output (PIN 15, 16 and17) which are suited for direct drive of N-Channel MOSFET transistors. Included in the MC33033 is the internal rotor position decoder used for monitoring the three sensor inputs from the motor for providing commutation sequencing to the top and bottom driver outputs which is used to drive the MOSFET transistors in the circuit. The MC33033 has 120^{o} select pin which configures the rotor position decoder for sensing electrical phasing inputs. The three Hall effects sensors are used to detect the position of the rotor. One of the Hall Effect sensor (Blue Phase) is configured to measure speed of the motor by measuring the time it takes for the sensor to switch. The microcontroller timer is used to measure the length of the pulse from the Hall Effect sensor. Measured time is then converted into revolution per minute (RPM). The converted speed is the actual speed used in calculating error for control action. A comparator (LM339) was used in the circuit to improve the strength of the sensor outputs from the BLDC motor for proper detection of rotor position. Noise from the sensor output is filtered out using capacitor C1

5.1.1 Control unit

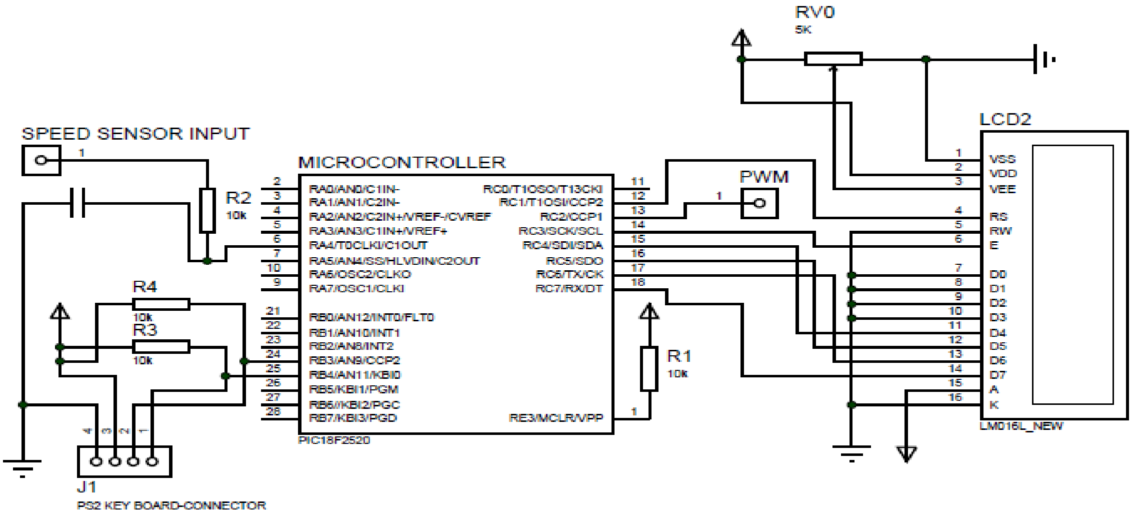


Figure 9: Control and display unit

The candidate controller was implemented around PIC18F2520 microcontroller as shown in Figure 9. Desired speed is input through the keyboard, and the microcontroller displays both the desired and the actual speed on Liquid Crystal Display (LCD). The controller measured the actual speed and compare it against the desired value to obtain the error signal. Next, this error is processed as per the control algorithms resulting in an output signal to the driver interface. The microcontroller ports are utilized as follows: Port C connects the LCD display, while the PWM needed for speed control of the BLDC motor is generated through pin 13 (RC2/CCP1). The actual speed of the motor is sensed through port A (RA4). This serves as the feedback to the controller. The whole control unit runs on a 32MHZ internal oscillator for the execution of every instruction.

6.0 Fuzzy Tuned PID Software Flowchart

The flowchart in figure10 above present the programming algorithm of fuzzy tuned PID controller. If the desired input is captured through the keyboard, microcontroller will read both the desired and the actual speed and display it on the Liquid Crystal Display (LCD display). At the same time the controller computes the error and change in error, if the error is equal to zero and change of error is equal to zero, the controller Maintain duty cycle and return to initial condition. But if the error is not equal to zero, the controller will determine the change in PID gains using fuzzy logic and update the duty cycle, then return to the initial condition.

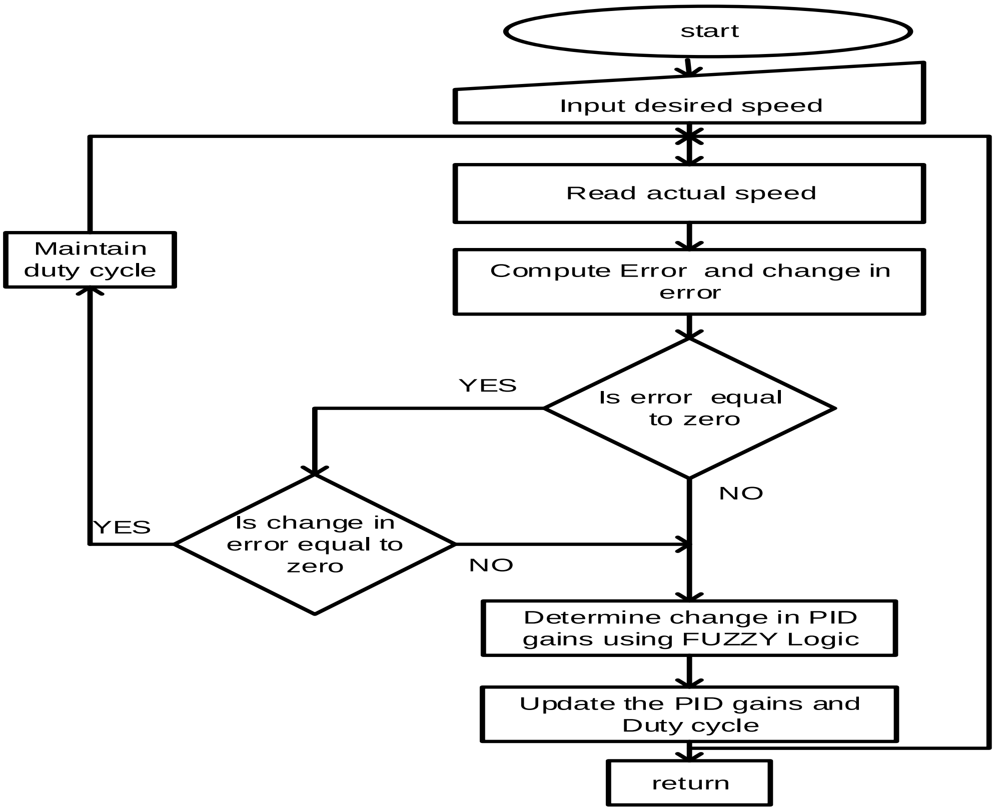

start
Input desired speed
Read actual speed
Compute Error and change in error
Is error equal to zero
YES
NO
Is change in error equal to zero
YES
NO
Maintain duty cycle
Determine change in PID gains using FUZZY Logic
Update the PID gains and Duty cycle
return

Figure 10: Fuzzy-tuned PID flow chart

7.0 Experimental Setup

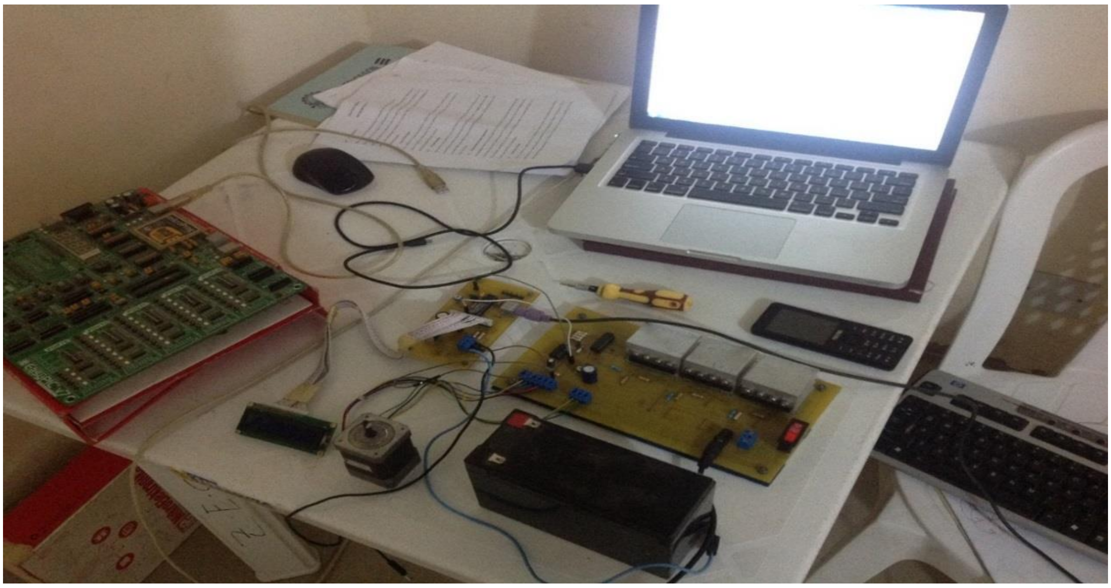


Plate 1: Fuzzy-tuned PID flow chart

The system units were interconnected to form a complete system setup as shown in plate 1 This

setup was used to carry out test at varying Speed ranging from 500rpm to 3000rpm.
The result for Convention PID and that of Fuzzy tuned PID was obtained and tabulated in the table 1and 2 respectively.

8.0 Experimental Result

Table 1: PID controller without load at varying speed of 500 to 3000 rpm

PID CONTROLLER WITHOUTLOAD				
S/NO	DESIRED SPEED (rpm)	ACTUAL SPEED (rpm)		STEADY STATE ERROR (rpm)
		L C D	SHARFT	
1	500	540	540	40
2	1000	999	999	-1
3	1500	1500	1505	0
4	2000	2020	2015	20
5	2500	2550	2550	50
6	3000	3030	3030	0

Table 2: Fuzzy-Tune PID controller without load at varying speed of 500 to 3000rpm

FUZZY-TUNE PID CONTROLLER WITHOUTLOAD				
S/NO	DESIRED SPEED (rpm)	ACTUAL SPEED (rpm)		STEADY STATE ERROR (rpm)
		L C D	SHARFT	
1	500	510	510	10
2	1000	1005	1000	5
3	1500	1500	1500	0
4	2000	2010	2010	10
5	2500	1520	1520	20
6	3000	3000	3000	0

The implementation of this research work was carried out using hardware components, the results are presented in tables 1and 2. The shaft speed which is displayed on an LCD, is also determined separately using a tachometer. The measured speed ranging from 500rpm to 3000rpm is compared with the desired speed on no-load. The steady state error for Fuzzy-tuned PID controller is comparatively less than that observed for conventional PID controller as can be seen clearly in table 1 and table 2.

9.0 Conclusion

In this research work fuzzy-tuned PID controller for the control of the BLDC Motor Speed was developed and implemented on a PIC18F2520 microcontroller. Experimental evaluations using different speed ranges (500rpm-3000rpm) reveals that the fuzzy-tuned PID controller offers a significantly better steady state response than that of conventional PID controller.

REFERENCES:

[1] D. Karthikeyan, K. Vijayakumar, T. Saptarshi, M. Kheria, and S. Kumari, "Implementation of Improved BLDC based Ceiling Fan Control Using Zeta Converter with Fuzzy Controller," *Int. J. Recent Technol. Eng.*, no. 1, pp. 551–559, 2019.

[2] P. Madasamy, K. Ramadas, and S. Nagapriya, "Fuzzy Logic Controller Based Direct Torque Control of PMBLDC Motor," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 5, no. X, pp. 487–496, 2017.

[3] A. Eid, M. A. Shamseldin, R. R. Darwish, and W. M. Refaey, "IMPLEMENTATION OF SELF-TUNING FUZZY PID CONTROL APPLIED ON BRUSHLESS DC MOTOR," *J. Al Azhar Univ. Eng. Sect.*, vol. 11, no. 41, pp. 1311–1320, 2016.

[4] J. Angeti, A. S.Kadalla, G. Dubukumah, and B. K. Zirata, "Speed Control of BLDC Motor using Fuzzy : Tuned PID Controller," *Commun. Appl. Electron. -Foundation Comput. Sci. FCS, New York, USA*, vol. 7, no. 29, pp. 6–12, 2019.

[5] A. Maghsadhagh, "Advances in Robotics & Automation Implementation of PID Controller by Microcontroller of PIC (18 Series) and Controlling the Height of Liquid in Sources," *Adv. Robot. Autom.*, vol. 5, no. 3, pp. 3–6, 2016.

[6] H. Mohan *et al.*, "Speed Control of Brushless DC Motor Using Fuzzy Based Controllers," *Int. Res. J. Eng. Technol.*, vol. 2, no. 4, pp. 875–881, 2015.

[7] R Arulmozhiyal, "Desing and Implementation of Fuzzy PID Controller for BLDC Motor Using FPGA," *2012 IEEE Int. Conf. Power Electron. Drives Energy Syst.*, vol. 12, 2012.