Nigerian Journal of Technology (NIJOTECH)
Vol. 36, No. 2, April 2017, pp. **594 – 602**
Copyright© Faculty of Engineering, University of Nigeria, Nsukka,
Print ISSN: 0331-8443, Electronic ISSN: 2467-8821
www.nijotech.com
http://dx.doi.org/10.4314/njt.v36i2.35

CONTROL OF A DC MOTOR USING FUZZY LOGIC CONTROL ALGORITHM

I. H. Usoro[1], U. T. Itaketo[2] and M. A. Umoren[3,*]
[1, 2, 3] Dept. of Electrical/Electronic and Computer Engineering, Univ. of Uyo, Akwa Ibom State, NIGERIA.
E-mail addresses: *[1] marconi40@yahoo.com, [2] engr1easy@yahoo.com, [3] mfon4gigis@yahoo.com*

ABSTRACT

This study sought to establish the impact of a fuzzy logic controller (FLC) and a Proportional–Integral–Derivative (PID) controller in the control performance of an industrial type DC motor using MATLAB. The fuzzy logic controller was developed on the basis of Mamdani type fuzzy inference system (FIS). The centroid method of defuzzification was also adopted. A choice of seven membership functions was designed for the error and change in error inputs alongside the output, hence a resultant of 49 rules was achieved for the fuzzy controller. Simulations for the PID controller and with the fuzzy controller were obtained at no load and for a selected reference speed of 1200 rpm. The simulations were further repeated for a load torque of 7Nm with the same reference speed. The simulated results were compared. Based on the findings, it was observed that the fuzzy speed controlled DC motors perform better than the PID speed controlled DC motors in terms of its faster settling time, absence of an overshoot and its sensitivity to applied load. Sequel to the foregoing findings, the researchers concluded that fuzzy logic controllers should be readily implemented in our local industries to enhance precision and improve performance in operations involving DC motors.

Keywords: *DC motor, Fuzzy logic, Rule base, PID Controller, Fuzzification*

1. INTRODUCTION

Over the years, most of the advancements in science and technology were as a result of developments made in control systems technology [14]. One of such advances is in the development of high performance direct current (DC) motor drives for industrial, as well as other applications. A DC motor can be defined as an actuator that converts electrical energy into rotational mechanical energy [5].

The need to attain optimal performance of dynamic systems, improve product quality, lessen the cost of production and relieve repetitiveness in manual industrial control operations, led to the design of a number of conventional control mechanisms such as proportional (P), proportional – integral (PI), proportional – derivative (PD) and proportional – integral – derivative (PID) controllers [13]. However, these conventional control mechanisms developed for the control of dynamic systems were faced with challenges arising from non-linearity and uncertainties that are characteristic of real-life control problems [12].

The era of artificial intelligence (AI), led to the creation of intelligent machines. An intelligent machine, according to [4], is a system that is able to achieve a goal or sustained behaviour under conditions of uncertainty. These machines were designed to act independently through the use of intelligent controllers. Some of the intelligent controllers in frequent use today are adaptive controllers, artificial neural networks (ANNs), and fuzzy logic controllers (FLCs). The ability of fuzzy logic controller to handle imprecise and inconsistent real-world problems has made it suitable for a wide variety of applications [10].

Efficient control is tightly related to improvements in the quality of industrial production processes. In industries employing DC motors for their operations, various conditions such as changes in motor load demand, non-linearity, disturbances, etc. have made it quite difficult for their efficient performance [1]. PID controllers amongst other conventional controllers have been used to tackle some of these problems. However, PID controllers still show some lag in its speed response to motor load demands. The potential of FLCs in resolving imprecision and non-linearity by heuristic knowledge of real life dynamic systems through user friendly interfacing has hereby made it paramount in its adoption towards addressing speed related control issues in DC motors [2].

2. REVIEW OF RELATED LITERATURE

Fuzzy logic theory was introduced by Professor Lofti A. Zadeh in 1965 and is based on the concept of fuzzy sets [9, 12]. Fuzzy logic theory, on which fuzzy control is based, provides an effective means of capturing the

[*] **Corresponding author, tel: +234-703-154-2457**

approximate, imprecise nature of the real world. The papers of Zadeh on the linguistic approach and system analysis based on the theory of fuzzy sets led to the pioneering research by Mamdani and his colleagues [12]. According to Zadeh, fuzzy set theory provides a means for representing uncertainty. In general, probability is the primary tool for analyzing uncertainty, and assumes that uncertainty is a random process. However, not all uncertainty is random and fuzzy set theory is used to model the kind of uncertainty associated with imprecision, vagueness and lack of information.

The fuzzy logic controller (FLC) provides an algorithm which can convert a linguistic control strategy into an automatic control strategy [6].Viewed in this perspective, the essential part of the fuzzy logic controller (FLC) is a set of linguistic control rules related to fuzzy implication and its rule of inference [9]. The Fuzzy controller converts a linguistic control strategy into an automatic control strategy. Fuzzy rules are constructed by expert experience or knowledge database [6, 10, 16].The basic fuzzy logic control system is composed of a set of input membership functions, a rule-based controller, and a defuzzification process. The structure of a fuzzy logic controller is illustrated in Figure 1.

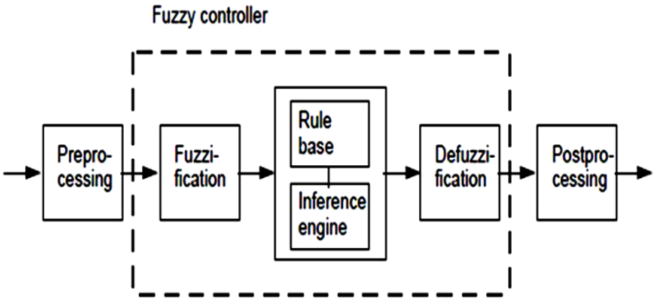


Figure 1: Structure of a fuzzy logic controller (Source [6]).

A typical fuzzy logic controller as illustrated in Figure 1, has the following modules: preprocessing, fuzzification, fuzzy inference, rule-base, defuzzification, post-processing. Each module is explained/ described as follows:

Preprocessing: The inputs are most often hard or crisp measurement from some measuring equipment rather than linguistic. A preprocessor, shows the conditions and measurements before entering the controller [7].

Fuzzification: The process of converting a numerical variable (real number or crisp variables) into a linguistic variable (fuzzy number) using the membership functions stored in the fuzzy knowledge base [9, 11].

Inference Engine: There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic ***Toolbox:*** Mamdani-type and Sugeno-type [8, 11]. The rules of the fuzzy controller map the strategy course to be undertaken by the inference engine. Should an error exist, the inference engine looks up the corresponding membership values as defined by the condition of the rule and maps it to the appropriate output membership function to be defuzzified (i.e. converted to a crisp output).

Rule Base: This step involves regulating a process output around a desired set point or reference value. There are a variety of different methods available for presenting the "IF-THEN" rule format [1]. The fuzzy rule base consists of a set of antecedent and consequent linguistic rules of the form; IF part called the "antecedent" and the "THEN" part called the "consequent"

Defuzzification: Defuzzification is the procedure for mapping from a set of inferred fuzzy control signals contained within a fuzzy output window to a non-fuzzy (crisp) control signal [11]. According to [15, 1, 16], defuzzification occurs when all the actions that have been activated are combined and converted into a single non-fuzzy output signal which is the control signal of the system. The output levels are depending on the rules that the systems have and the positions depending on the non-linearity's existing to the systems. There are five built-in methods supported: Centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum. The centre of area (centroid) method is the most well-known defuzzification technique which is expressed as:

$$Crisp\ Control\ Signal = \frac{sum\ of\ first\ movement\ of\ area}{sum\ of\ areas} \qquad (1)$$

For a continuous system, equation (1) becomes

$$u(t) = \frac{\int_1^n u\mu(u)du}{\int_1^n \mu(u)du} \qquad (2)$$

According to [12] an alternative equation for a discrete system is given as

$$u(KT) = \frac{\sum_{i=1}^{n} u\mu(u)}{\sum_{i=1}^{n} \mu(u)} \qquad (3)$$

In (3), μ represents the fuzzy membership values from μ_i .....μ_n, u is the crisp output signal.

Post-processing: Post-processing is used to scale the output of the controller. Not every control signal sent from the controller to the post-processing block will require scaling[13]. Therefore this block would be defined by a process engineer according to the process dynamics of the system. It often contains an output gain that can be tuned and also serve as an integrator [3].

3. METHODOLOGY

Two design techniques adopted in this study were: mathematical modeling based on first principles; and via simulations. These two techniques complement each other as they are both needed to achieve desirable results. A simulation environment in MATLAB was built. Prior to running the simulation in MATLAB/SIMULINK, the fuzzy logic controller was designed. This was done using the fuzzy inference system (FIS) editor. The design of the fuzzy logic controller required choosing appropriate membership functions after which a convenient rule base was created.

3.1 System Design Model

The chosen system is an industrial type, armature-controlled separately excited permanent magnet DC motor (PMDC). Its equivalent circuit is shown in Figure 2.

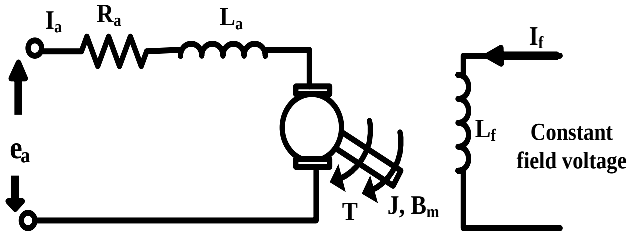


Figure 2: Equivalent Circuit diagram of an armature-controlled separately excited DC .motor.

In Figure 2, R_a is the armature resistance (Ω); L_a is the armature self-inductance caused by armature flux (H); I_a is the armature current (A); I_f is the field current (A); e_a is the input voltage (V); e_b is the back electromotive force (EMF) (V); τ_m is the torque developed by motor (Nm); θ is the angular displacement of the motor shaft; ω is the angular velocity of rotor (rad/s); J is the rotor inertia (Kgm^2); B is the equivalent coefficient of friction of motor and load referred to the motor (Nms/rad); K_v is the EMF constant (Vs/rad); K_T is the torque constant (Nm/A).

In the description of the motor, the armature reaction effects are ignored. This assumption is justifiable since the motor used has either interpoles or compensating winding to minimize the effects of armature reaction. When the field current is constant, the flux induced by the field winding remains constant, and usually it is held at its rated value Φ. The voltage, e_b, is the back emf in volts. In a separately-excited DC motor, the back emf is proportional to the product of speed of motor and the field. Considering the armature controlled D.C. motor in Figure 2 and assuming that the demagnetizing effect of armature reaction is neglected, the magnetic circuit is assumed linear and the field voltage is constant i.e. I_f = constant.

The back EMF e_b is directly proportional to the speed, hence:

$$e_b\,(t) = k_b\, {}^{d\theta}\!/_{dt} = k_v \omega(t) \qquad (4)$$

Applying KVL to the armature circuit we get

$$e_a(t) = R_a i_a(t) + L_a \frac{di_a}{dt} + e_b(t) \qquad (5)$$

From Newton's 2nd law of mechanical dynamics, the motor torque can be obtained as

$$\tau_m(t) = J\frac{d^2\theta(t)}{dt^2} + B\frac{d\theta(t)}{dt} + \tau_l(t) = k_T i_a \quad (6)$$

Where $\tau_l(t)$ is the motor load torque (Nm)

Taking Laplace transform of equations (4), (5), and (6) respectively we obtain

$$E_b(s) = K_v W(s) \qquad (7)$$

$$E_a(s) = (R_a + L_a s)I_a(s) + E_b(s) \qquad (8)$$

$$\daleth_m(s) = (Js + B)W(s) + \daleth_L(s) = K_T I_a(s) \quad (9)$$

(All at zero initial conditions)

The angular position is given as:

$$\theta(s) = \tfrac{1}{s} W(s) \qquad (10)$$

Equations (7 – 10) can be represented in block diagram form as shown in Figure 3.

The transfer function with respect to input voltage is expressed as:

$$G(s) = \frac{W\,(s)}{E_a(s)} = \frac{K_T}{(L_a B + R_a J)s + R_a B + K_v K_T} \qquad (11)$$

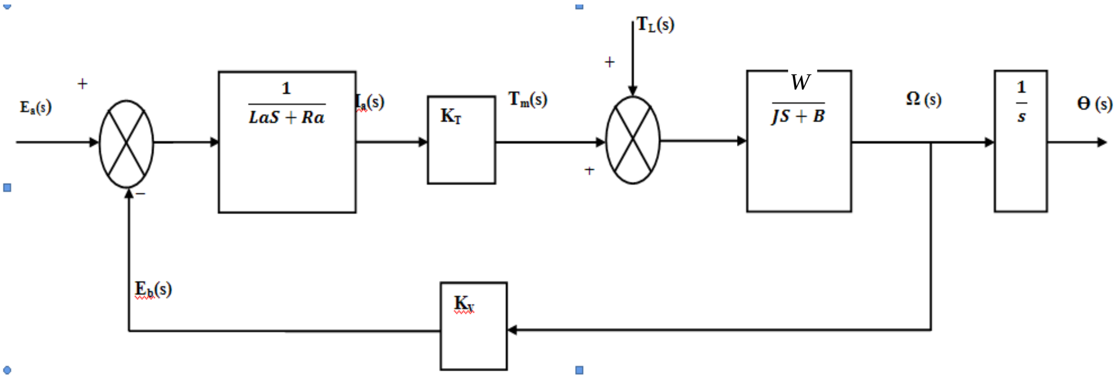


Figure 3: Block diagram of the armature controlled DC motor

$$G(s) = \frac{W\ (s)}{E_a(s)} = \frac{K_T}{[\,L_aJs^2 + \ (L_aB + \ R_aJ)s + \ (R_aB \ + \ K_vK_T)]} \qquad (12)$$

Our specified DC motor parameters are as follows: Rated Power: 3.7kW; Rated Speed: 1750rpm; Supply Voltage: 240V; L_a = 0.1214H; B = 0.002953Nms/rad; R_a = 11.4Ω; J = 0.02215 kgm^2; K_T = 1.28NmA; K_v = 0.0045 Vs/rad (Ahmed *et al,* 2013);

Putting these parameter values into the transfer function (equation 12), the equation becomes:

$$G(s) = \frac{1.28}{(0.2689\,X\,10^{-2})s^2 + \ (0.2529\,X\,10^{-2})s \ + \ 3.9424\,X\,10^{-2}} \qquad (13)$$

$$= \frac{1.28}{[0.2689s^2 + \ 0.2529s \ + 3.9424]X\,10^{-2}} \qquad (14)$$

Equation (14) is further simplified as

$$G(s) = \frac{128}{[0.2689s^2 + \ 0.2529s \ + 3.9424]} \qquad (15)$$

Equation (15) represents the transfer function of the DC motor based on chosen motor parameters. Figure 4 shows the DC motor model built in Simulink. The motor model is a single input single output system. The input port is the armature voltage (V_a), while the output port is the angular speed. The load torque (T_L) and armature current (I_a) outputs were included for observation purposes. This model is not pre-existent in the matlab environment.

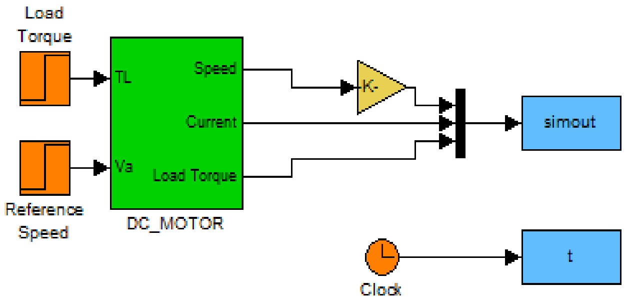


Figure 4: Simulink functional block model of the DC motor Subsystem

In the fuzzy logic controller design, the inputs and output linguistic variables were partitioned into seven (7) fuzzy subsets and are presented by seven (7) membership functions which are; Large Negative (LN), Medium Negative (MN), Small Negative (SN), Zero (Z), Small Positive (SP), Medium Positive (MP), Large Positive (LP). A triangular membership function was defined for each input and output variables which are: speed error (e), change in speed error (ce) and control output (u). The membership functions of the inputs and outputs were designed after the rule data base, and the controller was tested on the system to adjust any seemingly wrong parameters.

The rule base works by monitoring the error and change in error of the controller system. The rule base was formulated using fuzzy operators in IF – THEN statements i.e. IF (Process State – input parameters defined using fuzzy operator) THEN (Control Output). The choice of the fuzzy operator used was the Minimum (Product) fuzzy operator. The general rules for the DC motor speed control were that if motor speed is less than the desired speed, then increase motor speed and if motor speed exceeds the reference speed, then reduce the speed. If the error is large positive and the change in error is large negative, then the output of the controller must be zero and there is no need to increase the motor speed to meet the reference point which is directly proportional to the voltage of the controller. These fuzzy rules are summarized in the fuzzy rule matrix shown in Table 1.

Illustrating the fuzzy rule matrix Table 1 for three valid rules,

1. IF the error (e) is Small Negative (SN) AND the change in error (ce) is Large Positive (LP), THEN the control action (u) is Medium Positive (MP).
2. IF the error (e) is Zero (Z) AND the change in error (ce) is Large Negative (LN), THEN the control action (u) is Large Negative (LN)

Table 1: Fuzzy Rule Matrix Table

ce e	LN	MN	SN	Z	SP	MP	LP
LN	LN	LN	LN	LN	MN	SN	Z
MN	LN	LN	LN	MN	SN	Z	SP
SN	LN	LN	MN	SN	Z	SP	MP
Z	LN	MN	SN	Z	SP	MP	LP
SP	MN	SN	Z	SP	MP	LP	LP
MP	SN	Z	SP	MP	LP	LP	LP
LP	Z	SP	MP	LP	LP	LP	LP

3. IF the error (e) is Medium Positive AND the change in error (ce) is Medium Negative (MN), THEN the control action (u) is Zero (Z).

The error and change in error of the fuzzy controller is measured by the following formula:

$$\text{Error } e\ (k) = \omega_{\text{ref}} - \omega_m \quad (16)$$

$$\text{Change in error ce (k)} = \text{e(k)} - \text{e(k} - 1) \quad (17)$$

In (17), ω_{ref} is the reference speed and ω_{m} is the measured speed.

3.1 The Fuzzy System Toolbox Setup

The implementation of the FLC requires the choice of four key factors which are; the number of fuzzy sets that constitute linguistic variables, mapping the measurements into the support sets, control protocol that determines the controller behaviour and the shape of the membership functions.

Table 2 shows the numerical range of values for the input error and its corresponding linguistic notation ranging between -1 and 1 with seven triangular shaped membership functions.

The seven triangular shaped membership functions for the change in error ce (t) are illustrated in Figure 6.

The seven triangular shaped membership functions for the input variable error e(t) are illustrated in Figure 5

Table 3 shows the numerical range of values for the change in error and its corresponding linguistic notation ranging between -1 and 1 with seven triangular shaped membership functions.

Table 2: Numerical Range of Linguistic Variable for error e(t)

Linguistic variable	Notation	Numerical Range
Large Negative	LN	[-1, -0.667]
Medium Negative	MN	[-1, -0.667, -0.3332]
Small Negative	SN	[-0.667, -0.3333, 0]
Zero	Z	[-0.3333, 0, 0.3333]
Small Positive	SP	[0, 0.3333, 0.6667]
Medium Positive	MP	[0.3333, 0.6667, 0.9993]
Large Positive	LP	[0.667, 1]

Table 3: Numerical Range of Linguistic Variables for Change in error ce(t)

Linguistic variable	Notation	Numerical Range
Large Negative	LN	[-1, -0.667]
Medium Negative	MN	[-1, -0.667, -0.3332]
Small Negative	SN	[-0.667, -0.3333, 0]
Zero	Z	[-0.3333, 0, 0.3333]
Small Positive	SP	[0, 0.3333, 0.6667]
Medium Positive	MP	[0.3333, 0.6667, 0.9993]
Large Positive	LP	[0.667, 1]

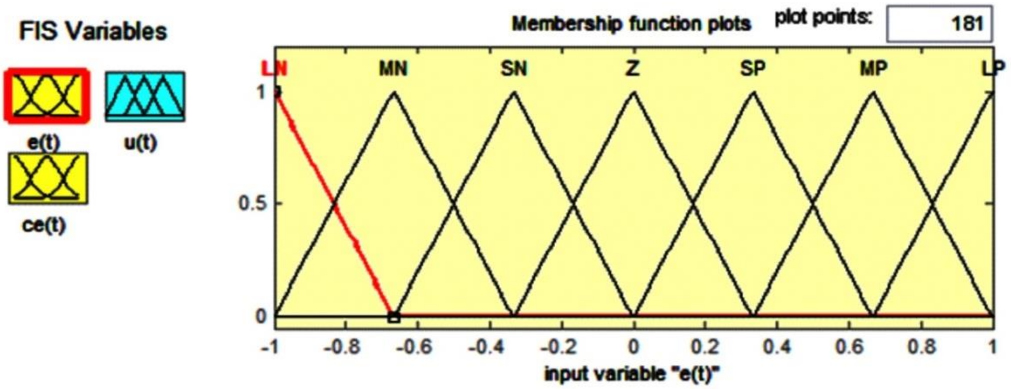


Figure 5: Triangular membership functions for input speed error e(t)

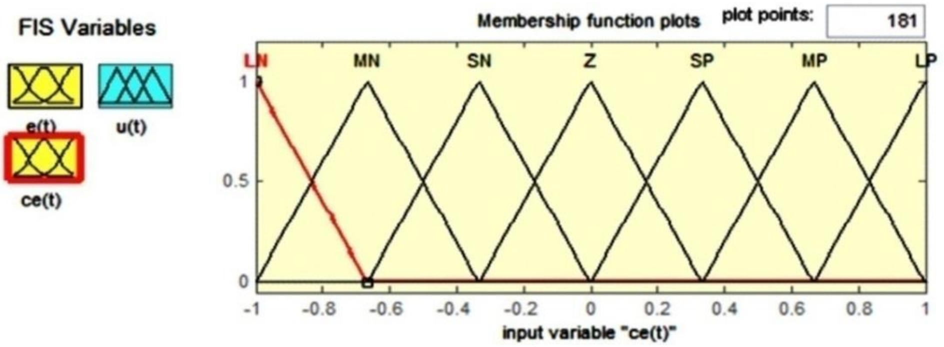


Figure 6: Triangular membership functions for change in error ce(t)

Figure 7 illustrates the representation of the seven output membership functions. The speed limit is given at 1750 rpm.

The rule base editor screen shown in Figure 8 locates where the FLC rules are set. A set of 49 rules was created for the three linguistic variables error e(t), change in error ce(t) and the output u(t) using the minimum inference rule. Figure 9 shows the rules viewer for the fuzzy controller. The rules behaviour can be checked here by changing the error e(t), and changing the change in error ce(t) and monitoring the output u(t). For instance in figure 9, when the input error e(t) = -0.2 and the change in error ce(t) = 0.5, the output u(t) = 0.312.

Table 4: Numerical Range of Linguistic Variables for Output u(t)

Linguistic variable	Notation	Numerical Range
Large Negative	LN	[-1, -0.667]
Medium Negative	MN	[-1, -0.667, -0.3332]
Small Negative	SN	[-0.667, -0.3333, 0]
Zero	Z	[-0.3333, 0, 0.3333]
Small Positive	SP	[0, 0.3333, 0.6667]
Medium Positive	MP	[0.3333, 0.6667, 0.9993]
Large Positive	LP	[0.667, 1]

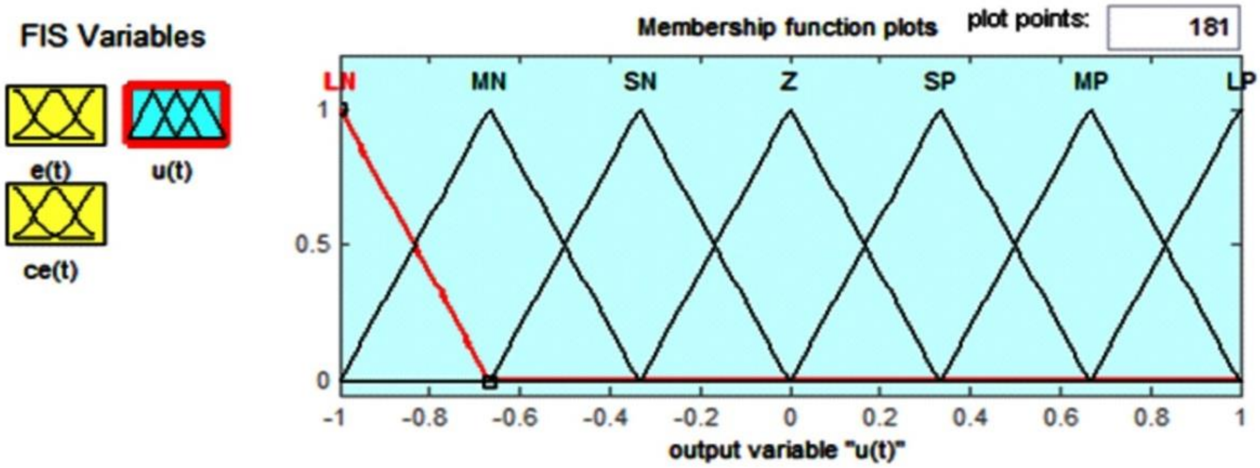
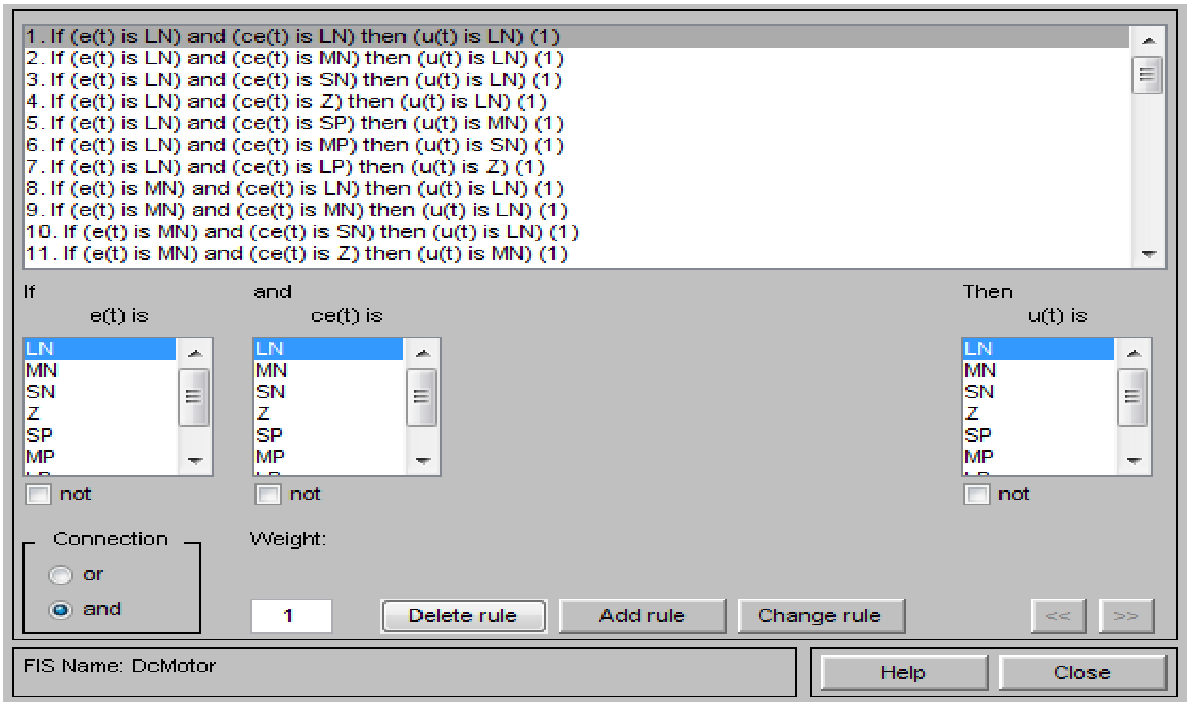


Figure 7: Triangular membership functions for variable output u(t)



1. If (e(t) is LN) and (ce(t) is LN) then (u(t) is LN) (1)
2. If (e(t) is LN) and (ce(t) is MN) then (u(t) is LN) (1)
3. If (e(t) is LN) and (ce(t) is SN) then (u(t) is LN) (1)
4. If (e(t) is LN) and (ce(t) is Z) then (u(t) is LN) (1)
5. If (e(t) is LN) and (ce(t) is SP) then (u(t) is MN) (1)
6. If (e(t) is LN) and (ce(t) is MP) then (u(t) is SN) (1)
7. If (e(t) is LN) and (ce(t) is LP) then (u(t) is Z) (1)
8. If (e(t) is MN) and (ce(t) is LN) then (u(t) is LN) (1)
9. If (e(t) is MN) and (ce(t) is MN) then (u(t) is LN) (1)
10. If (e(t) is MN) and (ce(t) is SN) then (u(t) is LN) (1)
11. If (e(t) is MN) and (ce(t) is Z) then (u(t) is MN) (1)

If e(t) is: LN, MN, SN, Z, SP, MP — □ not

and ce(t) is: LN, MN, SN, Z, SP, MP — □ not

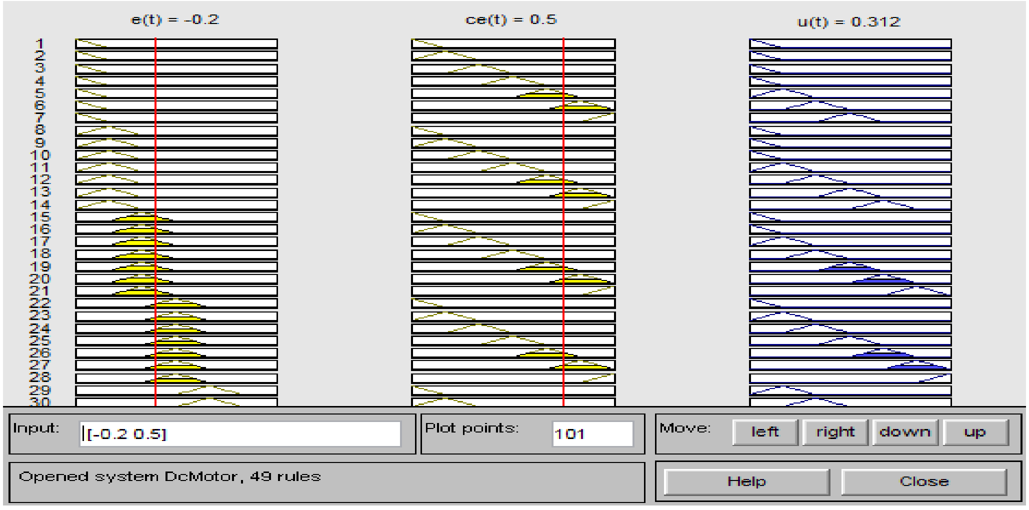
Then u(t) is: LN, MN, SN, Z, SP, MP — □ not

Connection: ○ or ● and

Weight: 1

FIS Name: DcMotor

Figure 8: Rule base Editor for the fuzzy controller



Input: [-0.2 0.5]

Plot points: 101

Opened system DcMotor, 49 rules

Figure 9: The Rules Viewer for the fuzzy controller

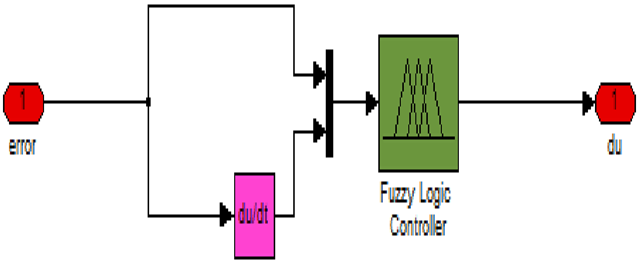


Figure 10: Simulink functional diagram of the fuzzy logic controller

Figure 10 represents the functional diagram of the fuzzy logic controller for the DC motor system built in Simulink. The reference speed is a unit step block. The load torque is also a step functional block ranging between no load to 20Nm load torque. The simulation of the above block was clocked and sent to the workspace to be called up for representation in the Graphical User Interface (GUI) screen.

4. RESULT AND DISCUSSION

The effects of the models' performance based on varying input model parameters such as load torque and reference speed. The impact of these varied input parameters on system performance in terms of rise time, settling time, peak overshoot and peak speed value were assessed. A reference speed of 1200 rpm was randomly selected at no load. This selected reference speed was later repeated for a load torque of 7Nm. Their corresponding responses with and without load were displayed with the PID controller, and with the Fuzzy controller.

4.1 System Performance for Reference Speed of 1200 rpm at no Load

The system's response and performance for a reference speed of 1200 rpm with the PID controller and with the FUZZY controller at no load, are presented in Figures 11, and 12 respectively.

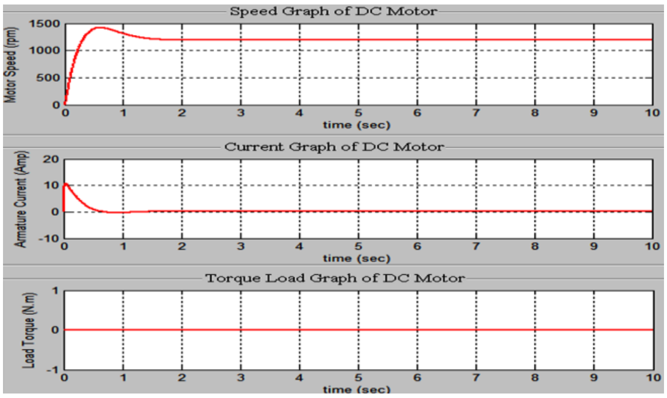


Figure 11: Response of system with the PID controller for reference speed of 1200 rpm at no load

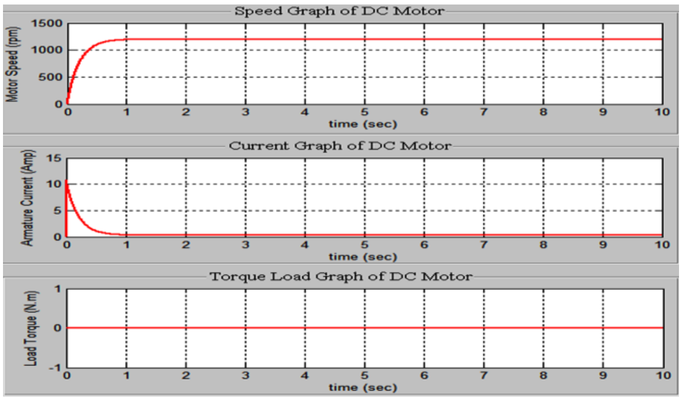


Figure 12: Response of system with fuzzy controller for reference speed of 1200rpm at no Load

The summary of the data in Table 5 shows the performance indices for the collective time responses of the DC motor with a PID controller and with a Fuzzy controller displayed in Figures 11 and 12 respectively for a reference speed of 1200 rpm at no load.

Table 5: System performance for reference speed of 1200rpm and at no load

Controller	Rise time Tr (Sec)	Settling time Ts (Sec)	Max. Overshoot (%)	Peak Speed Value (rpm)
With PID controller	0.224	1.387	18.3	1420
With Fuzzy controller	0.44	0.781	0	1200

Table 6: System Performance for reference speed of 1200rpm at a load of 7Nm

Controller	Rise time Tr (Sec)	Settling time Ts (Sec)	Max. Overshoot (%)	Peak Speed Value (rpm)
With PID controller	0.224	6.16	18.3	1420
With Fuzzy controller	0.44	0.781	0	1200

It can be deduced from this table that when compared with the PID controller, the fuzzy controller does not experience any overshoot due to its steady rise time. Also, it takes the fuzzy controller a shorter settling time to stabilize at the desired speed value of the motor.

The PID controller on the other hand possesses an overshoot due to its quicker rise time, resulting in a higher peak speed value thus taking a longer time to stabilize to the desired motor speed value.

4.2. System Performance for Reference Speed of 1200 rpm at 7Nm Torque Load

The system response and performance for a reference speed of 1200 rpm with the PID controller and with the FUZZY controller at 7Nm load are presented in Figures 13 and 14 respectively.

The summary of data analysis in Table 6 indicates the performance indices for the collective time responses of the DC motor with the PID controller and with the Fuzzy controller displayed in Figures 13 and 14 respectively for a reference speed of 1200 rpm at a load torque of 7 Nm.

In the event of an applied load torque, it can be deduced from Table 6 that when compared with the PID controller, the fuzzy controller does not experience any overshoot due to its unchanged steady rise time. The fuzzy controller's settling time also remains unchanged. It can also be observed in Figure 14, that up to the point of the applied load, the speed of the motor remained undisturbed. The fuzzy controller quickly adapts to the introduction of the load torque thereby maintaining the desired speed value of the motor.

On the other hand, the PID controller still possesses an overshoot, resulting in a higher peak speed value. However, in Figure 13, at the moment the load is applied, there is an indicated drop in the speed of the motor which eventually stabilizes after a few seconds.

5. CONCLUSION

The analysis on the performance of PID controllers and FUZZY LOGIC Controllers in the control of DC motor speed has been carried out in this work. The simulation results were obtained using MATLAB/SIMULINK software.

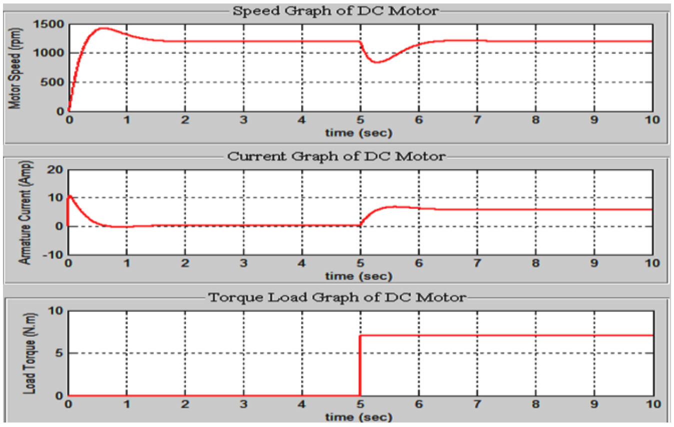


Figure 13: Response of System with PID controller for reference speed of 1200rpm at 7Nm load torque.

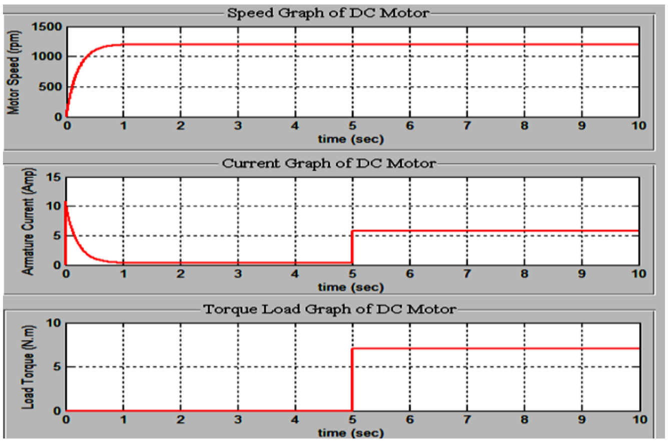


Figure 14: Response of System with Fuzzy controller for reference speed of 1200rpm at 7Nm load torque.

Comparisons were drawn from the DC motor responses between the PID controlled and Fuzzy controlled DC motor. The most desirable performance requires that the controllers have the smallest possible value for the maximum overshoot and the settling time. It is also required for the final speed value to be as close as possible to the desired/referenced speed value.

From the foregoing results, it can be observed that in controlling the speed of the DC motor, the Fuzzy controller performs more effectively than the PID controller due to the absence of an undesirable overshoot, its faster settling time and its ability to maintain DC motor speed irrespective of the torque load applied.

On the premise of these findings, it would be tactful in keeping the nation abreast with the era of artificial intelligence if fuzzy logic controllers could be readily implemented in our local industries to enhance precision and improve performance in operations involving DC motors.

6. REFERENCES

[1] Ahmed, H., Singh, G., Bhardwaj, V., Saurav, S., and Agrawal, S. "Controlling of DC Motor using Fuzzy Logic Controller", *Conference on Advances in Communication and Control Systems,*, DIT University, pp. 666-670, 2013.

[2] Aisha J., Sadi M., and Syed O. "Controlling Speed of DC Motor with Fuzzy Controller in Comparison with ANFIS Controller", *Intelligent Control and Automation*, Vol. 6, pp. 64-74, 2015.

[3] Asgharpour-alamdari H., Alinejad-beromi Y., and Yaghob H. "A fuzzy-based Speed Controller for Improvement of Induction Motor's Drive Performance", *Iranian journal of fuzzy systems*, vol. 13, no. 2, pp. 61-70, 2016.

[4] Burns, R. S. *Advanced Control Engineering,* Jordan Hill, Oxford, 2001

[5] Choi, J., "*Modelling of DC Motors: Control Systems Lectures"*, Department of Mechanical Engineering, University of British Columbia, pp. 1-15. 2008,

[6] Gupta, M. M. and Tsukamato, Y. *Fuzzy Logic Controllers – Perspective* San Francisco, California, 1980

[7] Harith M., Remya K. P. and Gomath S. "Speed Control of Brushless DC Motor Using Fuzzy Based Controller", *International Research Journal of Engineering and Technology (IRJET),* Volume 02 Issue 04, , pp. 875- 881. 2015.

[8] Kaur, A. & Kaur A. "Comparison of Mamdani Type and Sugeno Type Fuzzy Inference Systems for Air Conditioning System", *International Journal of Soft Computing and Engineering* (IJSCE), pp. 2231 – 2307. 2012.

[9] Suradkar, R. P. and Thosar, A. G. "Enhancing the Performance of DC Motor Speed Control Using Fuzzy Logic"*, International Journal of Engineering Research and Technology, IJERT*, pp. 103-110, 2012.

[10] Monika & Amrit K. "Comparison of Fuzzy Logic and Neuro Fuzzy Algorithms for Load Sensor", *International Journal of Soft Computing and Engineering (IJSCE),* Volume-3, Issue-2, pp. 219-222. 2013.

[11] Mathworks Inc "Simulink Getting Started Guide" www.apmath.spbu.ru/ru/staff/smirnovmn/files/sl_gs.pdf, Access on February 21, 2017.

[12] Nagrath, I. J. and Gopal, M. *"Control Systems Engineering".* Fifth Edition, Delhi, India, 2010

[13] Nwodoh T. and Ejimofor I. "Implementation of fuzzy logic based temperature-controlled, heat exchanger", *Nigerian Journal of Technology*, Vol. 29 Number.1, pp. 94-109. 2010.

[14] Ogbogu, S. E. & Itaketo U. T. *Systems and Control Engineering,* Nnewi: M C Computer Press, Nnewi, Nigeria, 2006

[15] Rai, J. N., Singhal, M. & Nandwami, M. "Speed control of DC Motor using Fuzzy Logic Technique". *Journal of Electrical and Electronics Engineering (IOSR - JEEE)*, pp. 41 – 48. 2012.

[16] Yen, J & Langari, R. *Fuzzy Logic.* Upper Saddle River, New Jersey, 2004.