Evolution in Electrical and Electronic Engineering Vol. 4 No. 2 (2023) 815-824
© Universiti Tun Hussein Onn Malaysia Publisher's Office

EEEE

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/eeee
e-ISSN : 2756-8458

Programmable DC Motor Position Using Fuzzy Logic Controller

Nur Azieyatul Akmal Mohamad Kamal[1], **Dirman Hanafi**[1]*, **Hisyam Abdul Rahman**[1]

[1]Departmant of Electrical Engineering, Faculty of Electrical and Electronic Engineering,
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/eeee.2023.04.02.098
Received 06 July 2023; Accepted 20 September 2023; Available online 30 October 2023

Abstract: This study aims to control the position of DC (direct current) motors using STM32 F411RE. The STM32 is used as an interface card between MATLAB Simulink, a platform for developing controllers, and a DC motor, the position can be controlled by a fuzzy controller. As a result, a programmable DC motor position controller utilizing fuzzy logic is successfully designed for hardware and Simulink MATLAB. It can also connect a DC motor to a computer using the STM32 NUCLEO-F411RE card as an interface for MATLAB. As a result, the FLC controller has a quick response due to the smaller average of rise time, settling time, and peak time rather than the proportional–integral – Derivative (PID). Other than that, FLC with a small overshoot, output angle is more accurate and precise as compared to the PID controller. It can be proven by the data that have been evaluating if program the desired angle to 90° to both controllers FLC it has lower overshoot with 0.08313% compared to PID controller with 19.4172%, FLC has fast peak time with 1.09617s compared to the PID is 1.1379s. FLC also has a fast transient time and settling time of 1.0922s compared to PID 1.9834s. By comparing the controller by program, the desired input is 180 degrees. FLC has a lower overshoot with 0.0835% approximately 0% compared to the PID controller which has reached 17.5621%. FLC also has a fast peak time of 1.1787s with a peak of 180.5° close to the desired angle compared to the PID is 2.1844s and a peak of 216.8°, and the result shows FLC has a fast response to the system due to the fast transient time and settling time with 1.1668s compare to PID 2.1844s.

Keywords: DC Motor, Position Control, Fuzzy Logic, PID

1. Introduction

Motors supply a substantial amount of power for mechanical operations. DC motors are one of the most common machines in use today, and they have a variety of industrial applications [1],[2]. It has been used to convert electrical to mechanical energy. The most widely recognized controller is the

---

**Corresponding author: dirman@uthm.edu.my*

Fuzzy Logic Controller (FLC). FLC was initially created by Zadeh in 1965 [3]. Fuzzy logic is a set theory that has been implemented in numerous control applications, including motor control. Fuzzy logic has facilitated the control of complex nonlinear systems with uncertain or unmodeled dynamics [3],[4]. Fuzzy logic has evolved as a practical and effective control technique for dealing with uncertain, imprecise, or qualitative decision-making issues. These days, MATLAB interface cards are quite expensive. As the interface of the control system for this paper, STM32 NUCLEO-F411RE was selected due to its low cost and low energy consumption. Moreover, compared to other interfaces, its reliability and performance are superior. STM32 NUCLEO-F411RE is a hardware platform that serves as an interface for computer-to-computer communication, allowing it to control the system as a brain. In this paper, the STM32 NUCLEO-F411RE will be used as an interface card between the controller development platform MATLAB Simulink and a DC (direct current) motor as the controlled object. DC motor shaft position is controlled by an FLC controller.

1.1 DC motor

An electric motor is a machine that converts electrical energy into mechanical energy. Alternating Current (AC) motors and Direct Current (DC) motors are the two types of motors [5],[6]. The four most common types of DC motors are brushed DC motors, brushless DC motors, stepper motors, and servo motors. These DC motors use three winding techniques: shunt DC motors, series DC motors, and compound DC motors. Currently, DC motor control is widely used to regulate motion (speed and position) [6]. The STMicroelectronics NUCLEO-F411RE STM32 was selected for use in this paper. This board was selected because it provides a flexible and cost-effective method for testing new concepts and prototypes, as well as a range of performance, power consumption, and features [7]. One of the MATLAB versions is version 2022b. It is a language for engineering and scientific programming. The mathematical calculation, design, analysis, and structural and mathematical optimization are all possible with MATLAB's fast, accurate, and precise results. MATLAB is a programming environment that allows users to create algorithms, analyze data, visualize it, and perform numerical calculations. The STM32 NUCLEO-F411RE interface will transfer data from MATLAB to the DC motor.

1.2 Position of DC motor.

Before the data also be transferred to the L298N Driver motor. Motor drivers are the interface that connects the motors to the control circuits. The controller circuit operates on low current signals, whereas the motor requires a high current. Therefore, motor drivers are designed to increase the voltage and current of control signals so that they can power motors [8]. This module employs a pair of controls to manage the rotational direction and speed of the DC motors. H-Bridge allows for direction control of rotation, while pulse width modulation (PWM) regulates speed. The purpose of this paper is to regulate the position so that a DC motor can be rotated by an H-Bridge by switching the input voltage's polarity. After that, the data will be transferred to a rotary encoder. It is a type of position sensor used as a feedback mechanism for measuring the angular position of a rotation shaft. The rotary encoder generates an electrical signal, which can be either in analog or digital form, in response to rotational motion. There exist two fundamental classifications of rotary encoders, namely absolute encoders and incremental encoders. Several technologies have been in the construction of rotary encoders, including mechanical encoders and optical encoders. The incremental encoder is a widely utilized type of encoder employed for the measurement of position and speed within a system. The encoder utilizes an optical disc to transform the rotational motion or position of a shaft into electrical impulses, which accurately determine the direction and angle of the rotation for position sensing.

1.3 Fuzzy Logic Control.

Prof. L.A. Zadeh developed a systematic approach to Fuzzy Logic Controllers (FLC). The Mamdani and Assilian controlled a small toy steam engine with fuzzy sets and an adaptive feedback control strategy. This is a first-time submission. A fuzzy controller converts a linguistic control strategy into an automatic control strategy using fuzzy logic, and fuzzy rules are derived from expert experience or a knowledge base [1],[9]. FLC has several advantages over traditional controllers, including control simplicity, low cost, and the ability to be designed without knowing the exact mathematical model of the process [10],[11]. Fuzzy logic has emerged as an effective and efficient control method for uncertain, imprecise, or qualitative decision-making issues [5]. The data from MATLAB Simulink that has been programmed the position of the DC motor and the data output of the DC motor position which is error will be used to system make the best decision to minimize the error of the DC motor position. In this stage, there are four stages of Fuzzy Logic Control Fuzzification, Rule-based, Inference Engine, and Defuzzification. Fuzzification is the most important process because it aims to convert crisp input to linguistic value [9],[12]. In addition, it involves the conversion of input and output fuzzy sets to numerical form. The Fuzzy Rule base includes a fuzzy logic quantification of the linguistic description provided by the specialist regarding how to achieve good control. Defuzzification is the process of converting the conclusion of an interface into an input-like form. The most prevalent method for defuzzification is the centroid technique.

1.4 PID

The P-I-D controller has the best control dynamics, including no oscillations, higher stability, quick response (short rise time), and zero steady-state error. To prevent oscillations and overshoots in the system's output response, a derivative gain component must be used in addition to the PI controller. The P-I-D controller's ability to be used with higher-order processes that involve multiple energy storage systems is one of its main advantages. The Position PID algorithm then uses this information, along with the gains and feed forwards, to figure out how much Control Output should be sent to move the axis to the Target Position. To get good control, the gains and feed forwards must be set to the right values of P, I, and D. A step function is often used as the set point command variable, and then the response of the process variable is measured to figure out how well the control system is working. Most of the time, the response is measured by looking at certain waveform characteristics. Rise Time is the amount of time it takes for the system to go from 10% of the steady-state, or final, value to 90% of that value. Percent Overshoot is the amount that the process variable goes over the final value, shown as a percentage of the final value. Settling time is the amount of time it takes for the process variable to settle within a certain percentage (usually 5%) of the final value. The final difference between the process variable and the set point is the steady-state error such as below.

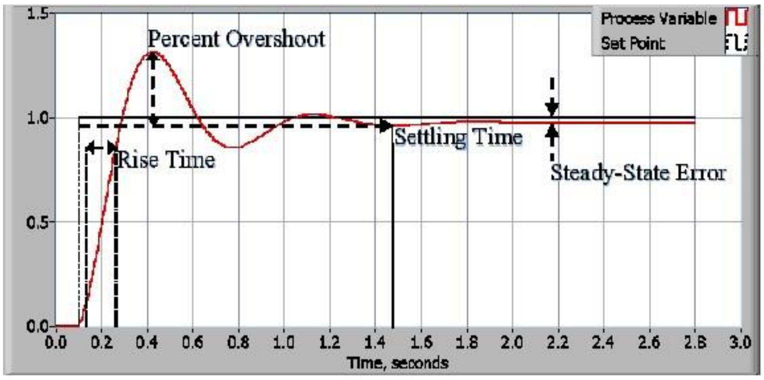


Figure 1: Response of a typical PID closed loop system.

2. Methodology Design for DC Motor Control

The STM32 NUCLEO-F411RE is an intermediary device connecting MATLAB Simulink, which serves as a platform for controller development, and a DC motor, which acts as the controlled object. The applied controller type for controlling the shaft position of a DC motor is a Fuzzy Logic Controller (FLC). The essential components necessary for the successful completion of this project include the

MATLAB software, STM32 Nucleo Board, a 12V DC motor, the L298N motor driver, and a rotary encoder as Figure 2.

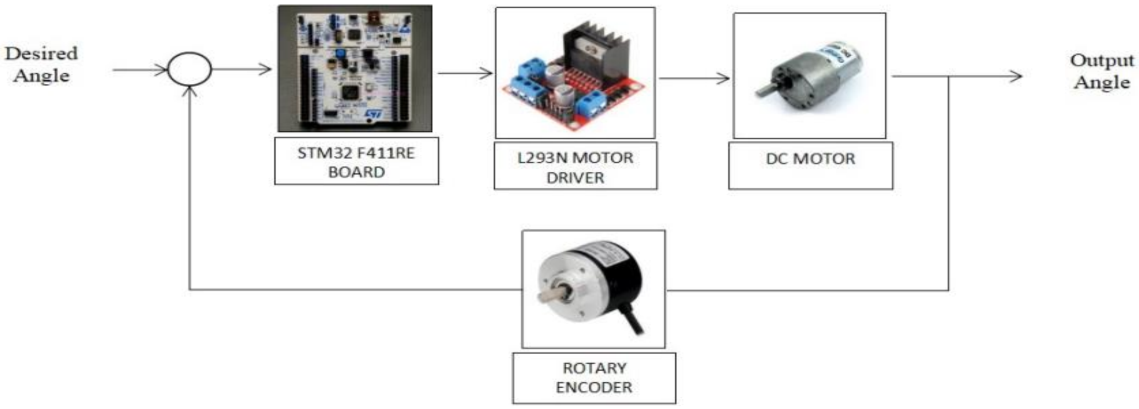


Figure 2: Block diagram of the project.

MATLAB is a necessary tool for controlling the DC motor via the STM32 NUCLEO-F411RE card. To connect MATLAB to the STM32 NUCLEO-F411RE card, the MATLAB software requires specific configuration steps. This is because the basic MATLAB software does not have specific tools that need to be installed to allow the STM32 NUCLEO-F411RE card to connect to the DC motor control system that will be created in the MATLAB software. These tools are required for the card to be able to connect to the system. The card can be used, and it works according to the user's instructions after the tools are downloaded. The specific way to download STM32 NUCLEO-F411RE in MATLAB software is to add more packages by clicking on the "Add-Ons" button which is:

A. The Simulink coder support package is an additional package needed for this paper because it uses an STM32 Microelectronics card to connect DC motors to MATLAB software.

B. The STM32 card needs a special package called the Embedded Coder support package for the STMicroelectronics STM32 processor to connect to the MATLAB software.

Fuzzy logic uses linguistics to process inputs to outputs. Fuzzification and defuzzification. MATLAB Basic needs the Fuzzy Logic Toolbox. Installing the tools lets the Fuzzy Logic Designer app design, test, and tune a fuzzy inference system (FIS) to model complex system behavior. The fuzzy logic controller uses a linguistic approach to process an input and generate the corresponding output in this paper by using the Mamdani method. Fuzzification involves the utilization of crisp inputs derived from the error and delta error obtained from the system's feedback.

Two input name error (e) and delta error (de) is used in this research. The input was the variable in this project to observe the output PWM based on the membership function setup. The first input error (e) value of the DC motor position error is selected from -360^{o} to 360^{o} and there as 5 membership that has been named Negative Big (NBe), Negative Small (NSe), Zero (Ze), Positive Small (PSe) and Positive Big (PBe). It is because of DC motor has a probability of producing error 0^{o} until 360^{o}. The second input delta error (de) value of the DC motor position error are selected from -180^{o} to 180^{o} and there as 5 membership that has been named Negative Big (NBde), Negative Small (NSde), Zero (Zde), Positive Small (PSde) and Positive Big (PBde), it is because DC motor has probability of delta error between -180^{o} until 180^{o}. Next from the two inputs, the fuzzy rule base uses IF-Then rules that can be derived to observe the output PWM value of the DC motor position. Fuzzy Rule base which contains a fuzzy logic quantification of the expert’s linguistic description of how to achieve good control. The fuzzy rules base obtained from 2 inputs will build 25 rule-base to determine the output which is PWM in Table 1.

Table 1: 5X5 Table Rules of the system to achieve zero error.

Delta Error / Error	Negative Big (NBde)	Negative Small (NSde)	Zero (Zde)	Positive Small (PSde)	Positive Big (PBde)
Negative Big (NBe)	CWB	CWB	CWS	CWS	Z
Negative Small (NSe)	CWB	CWS	CWS	Z	CCWS
Zero (Ze)	CWS	CWS	Z	CCWS	CCWS
Positive Small (PSe)	CWS	Z	CCWS	CCWS	CCWB
Positive Big (PBe)	Z	CCWS	CCWS	CCWB	CCWB

The values for the first input error (e) of the DC motor position error can range from -360° to 360°, and their five memberships have been given names such as Negative Big (NBe), Negative Small (NSe), Zero (Ze), Positive Small (PSe), and Positive Big (PBe)it can be seen in Figure 3.

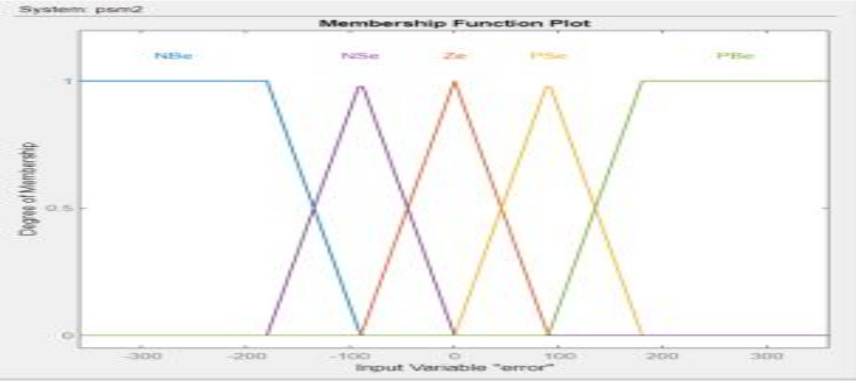


Figure 3: Error input setup membership in Fuzzy Logic Designer.

The second input delta error (de) value of the DC motor position error is selectable between -180° and 180°, with 5 memberships labeled as Negative Big (NBde), Negative Small (NSde), Zero (Zde), Positive Small (PSde), and Positive Big (PBde) as seen in Figure 4.

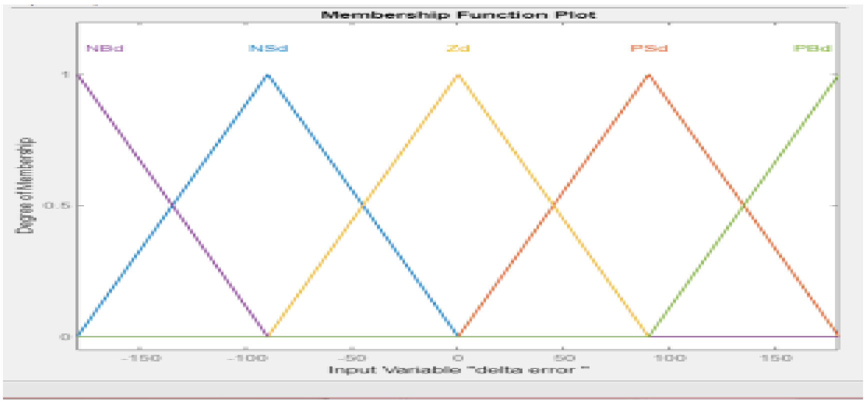


Figure 4: Delta Error input setup membership in Fuzzy Logic Designer.

Next from the two inputs, the fuzzy rule base uses IF-Then rules that can be derived to observe the output PWM value of the DC motor position. The output membership contains Big Clockwise (CWB), Small Clockwise (CWS), Small Counterclockwise (CCWS), and Big Small Counterclockwise (CCWB) where the range selected from -2550 to 2550. In this paper, 25 rules were obtained from the input and output to control the error of this system and surface control view of the FLC as Figure 5 and Figure 6.

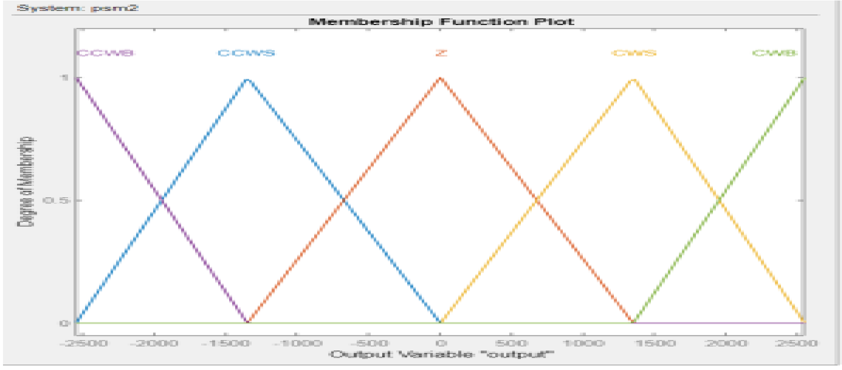
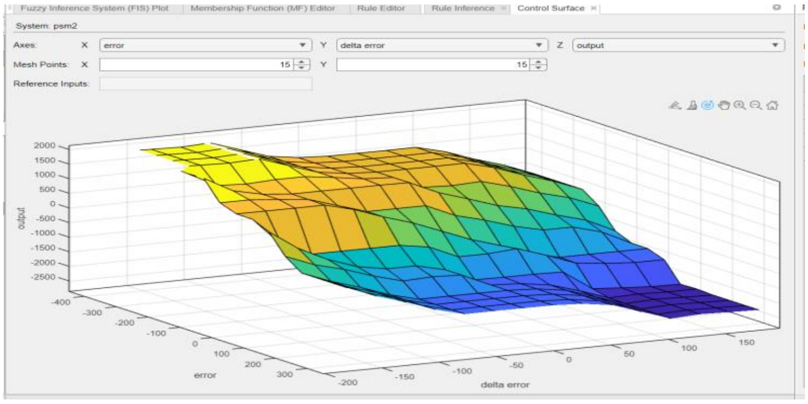


Figure 5: Membership output setup in Fuzzy Logic Designer.



System: psm2

Axes: X error Y delta error Z output

Mesh Points: X 15 Y 15

Reference Inputs:

Figure 6: View of control surface at Fuzzy Logic Designer.

2.3 Hardware Model

As shown in Figure 7, hardware for controlling the position of a DC motor. It was determined by prior research., the prototype was constructed using 3D printing and a combination of an STM32 board, driver motor, DC motor, encoder, and Perfboard. To simulate DC motor speed control using the STM32 NUCLEO-F411RE board as an interface with a DC motor in MATLAB Simulink, connect the STM32 board to the computer and configure the STM32 board's Pins to direct the system's workflow.

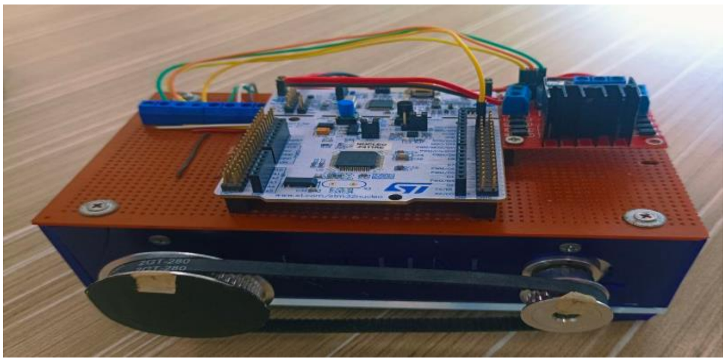


Figure 7: Hardware of the project

The MATLAB Simulink interface below was designed to control the position of a DC motor using a custom-designed STM32-F411RE card. The desired angle can be determined by modifying the parameter of the final value as below in Figure 8.

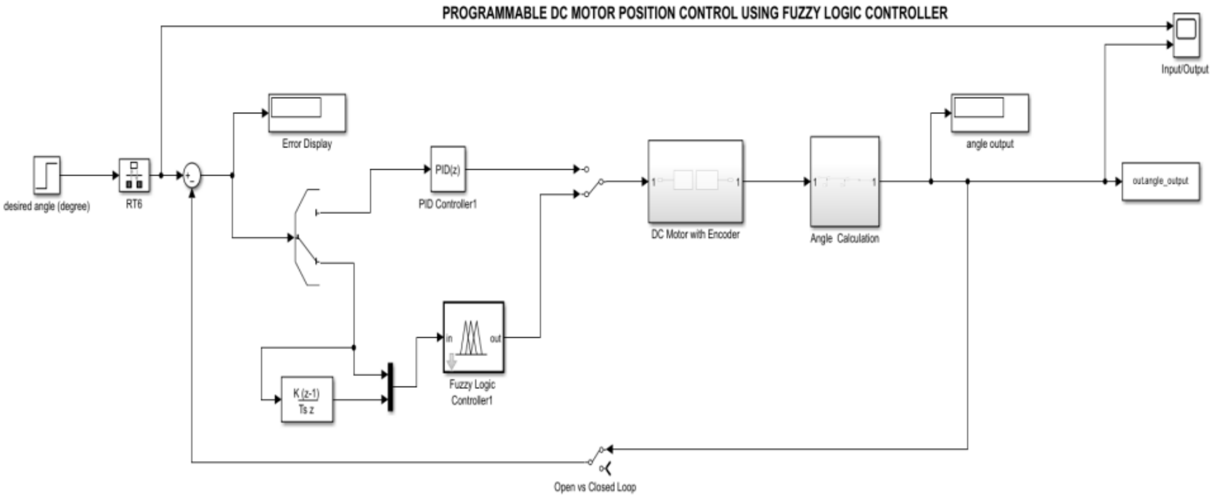


Figure 8: Model system at MATLAB Simulink of DC motor position

3. Results and Discussion

This system will control the DC motor's position output response using a fuzzy logic controller instead of a PID controller as in previous research. Step info will be used to observe the final results in MATLAB Simulink. since Simulink MATLAB and hardware are synchronized.

The simulations' output response to the position of the DC motor, which will be controlled using a PID controller and a fuzzy logic controller, will be used to determine the results in Table 2 to Table 6. STM32-F411RE hardware-based MATLAB Simulink interface controls DC motor position. With MATLAB Simulink, the desired angle is 90° or 180°. As a result, when the desired angle is 90° to study the system's effectiveness against the Fuzzy Logic controller, the parameters measured to determine the system's characteristics are very encouraging. Table 2 shows a summary average of rise time, transient time, settling time, overshoot, and peak time change a little after repeated measurements. which is 0.0774s, 1.0922s, 0.2494%, and 1.0987s to achieve desired angle.

Table 2: Summary data of step info of output when the desired value is 90° by implementing FLC.

Parameter \No. Of repeating	1	2	3	Average
Rise Time (s)	0.0718	0.0782	0.0822	0.0774
Transient Time (s)	1.0859	1.0935	1.0972	1.0922
Settling Time (s)	1.0859	1.0935	1.0972	1.0922
Settling Min (°)	88.9	88.5	86.2	87.9
Settling Max (°)	92.3	92.5	90.5	91.7
Overshoot (%)	0	0	0.2494	0.08313
Undershoot (%)	0	0	0	0
Peak (°)	92.3	92.5	90.5	91.7
Peak Time (s)	1.0914	1.0984	1.0987	1.09617

After that, the desired angle has been replaced from 90 ° to 180° to study the effectiveness of the system against the Fuzzy Logic controller. As a result, the overshoot was 0.0835%, and the peak time to achieve the desired angle was 1.1787s, with a mean rise time of 0.1272s and a mean settle time of 1.1668s as summary in Table 3.

Table 3: Summary data of step info of output when the desired value is 180° by implementing FLC

Parameter \No. Of repeating	1	2	3	Average
Rise Time (s)	0.1291	0.1257	0.1267	0.1272
Transient Time (s)	1.1671	1.1661	1.1672	1.1668
Settling Time (s)	1.1671	1.1661	1.1672	1.1668
Settling Min (°)	163.8	162.9	163.6	163.4
Settling Max (°)	181.0	180.1	180.5	180.5
Overshoot (%)	0	0.2505	0	0.0835
Undershoot (%)	0	0	0	0
Peak (°)	181.0	180.1	180.4	180.5
Peak Time (s)	1.1728	1.1699	1.1933	1.1787

The observation results from the data when the desired angle is placed at 90°and 180° to study the effectiveness of the system against the PID controller, it is found that the results of the measured parameters to determine the characteristics of the system after repeating many times the measured parameters such as rise time, transient time, settling time, overshoot and peak time there is not much change as seen in Table 4 and Table 5 it is accurate but not precise. This can be concluded is an average rise time of 0.0557s, settling time is 1.9834s, overshoot is 19.4172% and the peak time is 1.1379s to achieve the desired angle of 90°. When the desired angle is placed at 180° to study the effectiveness of the system against the PID controller, it can be concluded is the average rise time of 0.0702s, the settling time is 2.1844s, the overshoot is 17.5621% and peak time is 1.1590s to achieve desire angle.

Table 4: Summary data of step info of output when the desired value is 90° by implementing the PID controller.

Parameter \No. Of repeating	1	2	3	Average
Rise Time (s)	0.0546	0.0562	0.0562	0.0557
Transient Time (s)	2.2500	1.8501	1.8501	1.9834
Settling Time (s)	2.2500	1.8501	1.8501	1.9834
Settling Min (°)	86.5125	89.1000	89.1000	88.2375
Settling Max (°)	115.5375	115.7625	115.7625	115.6875
Overshoot (%)	21.9715	18.1401	18.1401	19.4172
Undershoot (%)	0	0	0	0
Peak (°)	115.5	115.8	115.8	115.7
Peak Time (s)	1.1464	1.1336	1.1336	1.1379

Table 5: Summary data of step info of output when the desired value is 180° by implementing PID controller.

Parameter \No. Of repeating	1	2	3	Average
Rise Time (s)	0.0692	0.0710	0.0705	0.0702
Transient Time (s)	2.5735	2.3407	1.6389	2.1844
Settling Time (s)	2.5735	2.3407	1.6389	2.1844
Settling Min (°)	164.8125	172.5750	169.6500	169.0125
Settling Max (°)	216.1125	217.1250	217.2375	216.8250
Overshoot (%)	19.3909	16.2651	17.0303	17.5621
Undershoot (%)	0	0	0	0
Peak (°)	216.1125	217.1250	217.2375	216.8250
Peak Time (s)	1.1539	1.1609	1.1623	1.1590

The output response shows that the Fuzzy Logic Controller has better stability, smaller average overshoot, faster average settling time, and peak time than the PID controller for this system. This paper shows that FLC controllers respond faster than PID due to their lower rise, settling, and peak times. Additionally, FLC with a small overshoot output angle is more accurate and precise than PID controller. Table 6 shows that if the desired angle is 90°, FLC has a lower overshoot of 0.08313% compared to the PID controller's 19.4172% and a faster peak time of 1.09617s compared to 1.1379s. FLC has faster transient and settling times than PID at 1.0922s and 1.9834s, respectively. Compare the controller by programming 180°. PID controller has a 17.5621% overshoot, while FLC has 0.0835%. FLC has a fast system response due to its fast transient time and settling time of 1.1668s compared to PID's 2.1844s and peak at 216.8°.

Table 6: Output response comparison of FLC and PID controller.

	Fuzzy Logic Controller		**PID Controller**	
Desired Angle (°)	**90**	**180**	**90**	**180**
Average Rise Time (s)	0.0774	0.1272	0.0557	0.0702
Average Transient Time (s)	1.0922	1.1668	1.9834	2.1844
Average Settling Time (s)	1.0922	1.1668	1.9834	2.1844
Average Settling Min (°)	87.9	163.4	88.2	169.0
Average Settling Max (°)	91.7	180.5	115.7	216.8
Average Overshoot (%)	0.08313	0.0835	19.4172	17.5621
Average Undershoot (%)	0	0	0	0
Average Peak (°)	91.7	180.5	115.7	216.8
Average Peak Time (s)	1.09617	1.1787	1.1379	1.1590

4. Conclusion

In conclusion, a programmable DC motor position using a fuzzy logic controller is successfully designed and implemented in MATLAB. The STM32 NUCLEO-F411RE card connects the DC motor and computer with MATLAB software. Additionally, the system designed and implemented Fuzzy Logic. Fuzzy Logic Controller output response has better stability, smaller average overshoot, faster average settling time, and peak time than PID controller. The output step response has a lower overshoot with 0.08313% compared to 19.4172% for the PID controller and a faster peak time with 1.09617s compared to 1.1379s. FLC has faster transient and settling times than PID at 1.0922s and 1.9834s. If the desired input is 180 degrees, FLC has 0.0835% overshoot, while PID controller has 17.5621%. FLC has a fast system response due to its fast transient time and settling time of 1.1668s compared to PID's 2.1844s and peak at 216.8°. If the desired input is 180°, the PID controller overshoots 17.5621%, while the FLC overshoots 0.0835%, close to 0%. FLC has a fast peak time of 1.1787s and a peak angle of 180.5°, close to the desired angle. Unlike PID, which has a peak angle of 216.8° and a peak time of 2.1844s. FLC's 1.1668s transient and settling time is faster than PID's 2.1844s, demonstrating its fast system response.

Acknowledgement

The authors would also like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

References

[1] Manikandan, R., & Arulmozhiyal, R. (2014, January). Position control of DC servo drives using a fuzzy logic controller. In 2014 International Conference on Advances in Electrical Engineering (ICAEE) (pp. 1-5). IEEE.

[2] Thomas, N., & Poongodi, D. P. (2009, July). Position control of DC motor using genetic algorithm-based PID controller. In Proceedings of the world congress on engineering (Vol. 2, pp. 1-3).

[3] K. K. Kumbla and M. Jamshidi, "Control of robotic manipulator using fuzzy logic," Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference, 1994, pp. 518-523 vol.1, doi: 10.1109/FUZZY.1994.343731.

[4] Muruganandam, M., & Madheswaran, M. (2009, June). Modeling and simulation of modified fuzzy logic controller for various types of DC motor drives. In 2009 International Conference on Control, Automation, Communication and Energy Conservation (pp. 1-6). IEEE.

[5] Azman, M. A. H., Aris, J. M., Hussain, Z., Samat, A. A. A., & Nazelan, A. M. (2017, November). A comparative study of fuzzy logic controller and artificial neural network in speed control of separately excited DC motor. In 2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE) (pp. 336-341). IEEE.

[6] Nasrudin, I. H. B. M. (2019). Comparative performance of fuzzy controller with different rules for position of servo dc motor.

[7] STM32F411RE - STMicroelectronics. (2022, November 16). STMicroelectronics. https://www.st.com/en/microcontrollers-microprocessors/stm32f411re.html

[8] A. (2020, July 26). Introduction to L298N Motor Driver | How it’s work » ElectroDuino. ElectroDuino. https://www.electroduino.com/introduction-to-l298n-motor-driver-how-its-work/

[9] Martin, M., Margana, D. B., & Habinuddin, E. (2023). Fuzzy Logic Controller Implementation for Motor DC Control Position With Real-Time Operating System. IJISTECH (International Journal of Information System and Technology), 6(5), 654-663.

[10] Lee, C. C. (1990). Fuzzy logic in control systems: fuzzy logic controller. I. IEEE Transactions on systems, man, and cybernetics, 20(2), 404-418.

[11] Prema, K., Kumar, N. S., & Dash, S. S. (2014). Online control of DC motors using fuzzy logic controller for remote-operated robots. Journal of Electrical Engineering and Technology, 9(1), 352-362.

[12] Prema, K., Kumar, N. S., Dash, S. S., & Chandran, S. S. (2012, February). Online control of the fuzzy-based mine-detecting robot using virtual instrumentation. In 2012 International Conference on Computing, Communication and Applications (pp. 1-6). IEEE.