Journal of Mechanical Engineering *Vol 19(3), 235-255, 2022*

Robotic Arm Position Control using Mamdani Fuzzy Logic on Arduino Microcontroller

Muhammad Hamizan Mohd Hamzah, Norashikin M. Thamrin, Mazidah Tajjudin*
School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA, Shah Alam, Malaysia
**norashikin@uitm.edu.my*

ABSTRACT

A robotic arm is the most often used robot in manufacturing to perform the same task accurately again and over again in a controlled environment. However, direct positioning control of the robotic arm is always inaccurate as it does not consider its position in the external environment. Therefore, this work implements the Mamdani fuzzy logic in the position control of the 6 DOF robotic arm to improve the accuracy and movement of each joint at this manipulator. The implemented fuzzy logic Mamdani inference system is done in MATLAB and finally converted to the C language to accommodate the main microcontroller, Arduino UNO environment. There are 6 servo motors controlled by this approach and the result is compared with the conventional method. It is found that the Mamdani fuzzy logic controller has an average error of 0.67% while the direct control method has an average error of 2.33%. Based on this result, it shows that the use of the Mamdani fuzzy logic increases the accuracy and reduces the swaying movement after stopping the robotic arm movement.

Keywords: *Robotic Arm; Mamdani Fuzzy Logic; MATLAB, Fuzzy Logic Controller; Microcontroller*

Introduction

When robots are utilized in industrial automation, operations become more efficient and effective. The robotic arm is a common piece of robotic equipment that was developed in the 1960s. It is commonly used for a variety of tasks such as assembly, material handling, painting, and welding. On

ISSN 1823-5514, eISSN 2550-164X

Received for review: 2022-02-28
Accepted for publication: 2022-08-01
Published: 2022-09-15

assembly lines, robotic arms are used to boost productivity and efficiency while also improving product quality. For human workers, most of these tasks are monotonous, unpleasant, or dangerous. The primary advantages of robotic arms over human employees are their resistance to fatigue and stress, as well as their tolerance to many environmental conditions. Additionally, robotic arms can be programmed and operated in such a way that they accomplish a task more effectively than a human.

Robotic systems have intricate dynamical behaviour. Friction, stiction, and gear backlash all influence robotic joints. The primary goal is to build a controller capable of tracking the current position concerning the desired trajectory. The increasing use of robotic arms necessitates the development of ever-evolving control mechanisms to regulate the movements of the arm's different joints and end effectors. Various control algorithms have been established in the past for processing the error signal generated when the actual joint angle is compared to the desired joint angle, resulting in a control voltage signal to be supplied to the actuator.

The problem with controlling an actuator, such as a numerous servo motor in this scenario, is that the typical way of controlling a servo motor is insufficient. According to a study, many servo systems establish a feedback loop based on the information from the servo actuator, but not on the shaft end [1]. This is referred to as a semi-closed loop control system, and it is insufficient for precise movement and control of the servo robot arm.

Fuzzy systems have been successfully applied to classification, modeling, and control problems, among other applications. In most cases, the key to success was the capacity of fuzzy logic systems to combine human expert knowledge with a mathematical model. However, the lack of Fuzzy logic controller (FLC) implementations on low-cost microcontrollers is also a problem. FLC is a sophisticated but versatile controller that may be used in a broad variety of operating settings and is easily customizable to different language terms [2]. Unfortunately, there are few implementations of a low-cost or basic system, which reinforces the idea that FLC is primarily only applicable in the industry.

Another problem is the robot manipulator's control algorithm. Robot manipulators demand a high degree of precision, and numerous controllers have been created and implemented in a variety of simple systems and robot manipulators [3]. FLC is one of the controllers that may be utilized to ensure high efficiency and low error in comparison to the standard method of controlling the robot arm. The reason FLC is chosen is that it is proven to be more reliable and efficient than the proportional, integral, derivative (PID) controller since it is less sensitive to changes in the system and consumes less energy [4].

Therefore, the objectives of this work are; (1) to develop a 6-DOF robotic model with the embedded controller, (2) to define the Mamdani fuzzy logic controller for the robotic arm control system, and (3) to validate its performance based on the position of the servo motor at each joint.

Related Literature

Mamdani Fuzzy Interference System (FIS)

In 1965, Professor Lotfi A. Zadeh of the University of California, Berkeley, developed the concept of fuzzy logic, an influential theory for the explanation and development of control systems that provides design engineers with a straightforward, intuitive means of integrating complex systems. Fuzzy logic systems allow for inputs with multiple degrees at multiple points at the same time, allowing engineers to define them more naturally. Since then, the fuzzy logic has evolved algorithmically and received enormous attention in automated systems. Significantly, in 1974, Mamdani and his colleagues used a fuzzy logic controller to control a steam engine and boiler combination using a set of linguistic control rules obtained from experienced human operators. The output of each rule in the Mamdani inference system is a fuzzy logic set.

Mamdani FIS is useful in controlling the multi-parameter and non-linear system effectively and effortlessly [5]. The fundamental of the Mamdani FIS structure in a closed-loop system is shown in Figure 1. Based on this figure, there are four stages of the operations in Mamdani fuzzy logic, namely, (1) the rule-base, (2) the inference mechanism, (3) the fuzzification interface, and (4) the defuzzification interface. The rule-base operation is the knowledge for the robotic control system using the IF-THEN-ELSE statement. Usually, the input of this operation is the deviation error between the actual output and the desired output. The inference mechanism evaluates which control rules are pertinent at present and then determines the plant's input requirements. The fuzzification interference can be regarded as the input interface to the fuzzy system. It modifies or scales the inputs from the feedback of the robotic system so that they can be compared against the rules in the rule-base operation. Finally, the interface for defuzzification serves as an output interface. It converts the output of the interference to a meaningful value that is fed into the actual process's input.

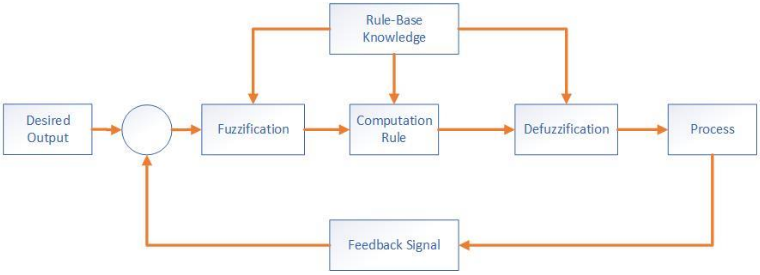

Figure 1: The basic components in Mamdani fuzzy logic controller in a closed-loop system

Previous work on fuzzy logic controller (FLC) in robotic arm control

PID is frequently used in industrial control systems that require continuously modulated control. A robot manipulator controlled by using a combination of PID and Fuzzy controller has been implemented on the Arduino microcontroller [6]. In their technique, Sugeno fuzzy model is used. The results indicated that the robotic arm's accuracy is near to the setpoint and is superior to that of a digital PID controller. The Sugeno FIS uses a weighted average for the output, which is in contrast with the Mamdani. Mamdani divide the output into levels and labeled them to correspond to the output membership functions. Additionally, there is an implementation of a fuzzy logic-based joint controller (FLJC) for a 6-DOF robotic arm that incorporates machine vision feedback [7]. The robotic arm is a closed-loop system in which the joint angles, gripper coordinates, and target object coordinates were all controlled by the FLJC through a communication protocol.

Another application of fuzzy logic is to tune the gain value in the PID for greater flexibility in controlling the manipulators and producing the optimal control function through the simulation platform, MATLAB [8]. It has been reported that a system with a PID-fuzzy logic tuning method outperforms a conventional PID tuned method. A comparative analysis is conducted to determine the optimal tuning strategy for controlling a 6-DOF robot arm [9]. The study compares three types of control methods with different tuning methods, namely, the Ziegler-Nichols PID controllers, the fuzzy logic controllers, and the Fuzzy-PID controllers. It compares the response time domain and the steady-state error. The simulation is carried out in MATLAB/Simulink and the results indicate that Fuzzy-PID controllers outperform the others in the presence of load changes and system disturbances.

An FLC can also be used to control the angular position of a robotic arm with rotating flexible joints [10]. The FLC's performance is assessed using simulations and experimental results, and the findings show that the

FLC performed satisfactorily in terms of regulating the appropriate tip angle position and reducing oscillations. FLCs, when properly constructed, can also function as nonlinear controllers [11]. FLC typically improved system efficiency, response time to command signal changes, and overshoot.

Furthermore, another study claims that developing a fuzzy model is extremely simple due to its simple structure [12]. The model does not require advanced mathematical knowledge and instead relies on process knowledge. Furthermore, neuro-fuzzy systems are a hybridization of fuzzy logic that combines the knowledge representation capabilities of fuzzy logic with the learning capabilities of artificial neural networks. Optimization can help with the resolution of kinematics problems and contribute to robotics. Another area of fuzzy logic research is the use of fuzzy control and adaptive network fuzzy interference system approaches to regulate the joint position of a 4-DOF robotic arm to achieve the desired movement [13]. Furthermore, the system's performance is compared to that of a traditional PID controller in terms of overshoot, settling time, and steady-state inaccuracy in this study. The results of the simulation and analysis show that using fuzzy logic improves system performance and reduces position tracking error.

A study [14] found that fuzzy logic can also be used to improve the precision of the robotic arm's position. The current PID controller is replaced by a fuzzy logic controller in the research, which uses an Arduino as the microcontroller. The performance of the robotic arm improved after switching to a fuzzy logic controller. The fuzzy logic compares the current location to the desired value, identifies the error, and adjusts the current position so that it is as close to the desired value as possible. Fuzzy logic can also be used for adaptive control by planning the robot arm's trajectory using the Denavit-Hartenberg (D-H) method. A 6-DOF robotic arm is used to validate the adaptive control design [15]. The hardware and software were designed, and the experimental results show that the D-H method can be used to determine the model of the robot arm and that MATLAB can be utilized to comprehend the processed data. The adaptive control has a reasonable level of stability and is simple to use. It is possible to improve the control of the robotic arm after learning about the capabilities of fuzzy logic. Furthermore, the potential of a robot arm must be recognized. As an example, one study uses a robotic arm and an Artificial Neural Network (ANN) to classify objects based on their colour [16]. An image processing system detects the object and generates an inverse kinematic model, which is then trained with an ANN system. The implementation of the robot arm is advantageous due to its high accuracy and low system cost. The fuzzy logic from MATLAB's fuzzy inference system is converted to the Arduino C programming language to simplify the fuzzy logic model in the microcontroller environment [17]. It simulates the fuzzy logic system with two sensor inputs and employs the Sugeno FIS in MATLAB before converting it to a format that the Arduino IDE can read. The conversion result is a variable, and the fuzzy model is an

IF-ELSE sentence. Although the Sugeno FIS seems to be more accurate, the consistency of Mamdani's inference process cannot be ruled out [18]. Mamdani FIS has a solid defuzzification process that keeps the result in a consistent form. Due to this reason, the Mamdani FIS is chosen in this work as it simplifies the implementation in the low-cost microcontroller.

Robotic kinematic

Knowledge of kinematics is critical for robotic manipulators. 6 DOF robotic manipulators are being studied for forward and inverse kinematics [19]. A movement flow plan is created and used to compute the end effector's desired location and orientation. The study also compares the outcome to an analytical solution to determine the model's inaccuracy. In research for a four-DOF robotic arm manipulator using an FLC, inverse kinematics analysis can also be performed [20]. The FLC determines the proper location and motion of the end effector based on sensor data. The setup is critical to avoid using excessive force or making a positioning error. Further confirmation on solving kinematics with fuzzy logic is to provide a solution to inverse kinematics difficulties [21]. Fuzzy logic and a PD controller are used in the research to control the multi-link robotic arm. The method's primary benefits are its simplicity and ease of use. The results show the utility of fuzzy logic in conjunction with PD control. Other studies that are directly related to the research goal include the development of a 6-DOF industrial robot using the capabilities of the artificial potential field approach [22]. Their investigation's goal is to see if the robotic arm can avoid obstacles while still maintaining the appropriate setpoint. Research aimed at presenting a design of control techniques for a 6-DOF robotic manipulator [23] can also be used to determine the design and hardware of a multi-DOF robotic manipulator. The outcome considers the efficiency of the manipulator's trajectory tracking as well as the utility of the control. A keyboard-based control and simulation of a 6-DOF robotic arm [24] provide another example. The goal of the research is to create a robotic arm that can be used for search and rescue operations and is controlled by a more user-friendly keyboard. The control interface is built with a robot operating system and is designed to be low-cost and simple to use. Another approach is to track the trajectory of a 6-DOF industrial robot using a direct adaptive robust tracking control [25]. It is based on the dynamic properties of a 6-DOF robot's end-working effector space. When compared to another controller, the result is more accurate trajectory tracking, even when there are several unknown variables.

A robot arm can also be attached to a quadrotor to function as an aerial manipulator capable of manipulating a wide range of environments as well as transporting and handling objects. The use of a 2-DOF robot manipulator that can operate as a 6-DOF manipulator when combined with the movement and angle of a quadrotor is investigated [26]. However, the system's inverse kinematics and control are quite difficult because multiple conditions must be considered, such as the yaw, pitch, and roll angles, as well as the amount of thrust required from the rotor to maintain the quadrotor's position.

Methodology

The method of this work is divided into three phases. The initial phase is to design the robot manipulator used in this work. The total weight of the robotic arm is considered in choosing the right servo motor so that it can move towards any point accordingly. At this phase, the kinematic of the robot manipulator is studied to determine the joints' position after every movement for result verification. In phase 2, the FLC is designed and simulated using Matlab before converting it into C code, to be compatible with the Arduino platform. The system requires multiple inputs from the position of the joint to determine the corrective control output. Lastly, in the final phase, the actual output of each joint is compared with its setpoint between the non-FLC and FLC system.

Phase 1: Robot manipulator design

Hardware design

A 6-DOF robot arm manipulator model used in this work is shown in Figure 2, which consists of six servo motors that act as the joints of the robot manipulator. The total weight of the robot manipulator is nearly 1 kg, with the arm weighing approximately 0.81 kg and the end effector weighing 0.105 kg due to its metal construction. This accumulative weight did not include the weight of the microcontroller, servo shield, battery as the main power supply, and the wooden base platform that supports the entire structure of the robot manipulator.

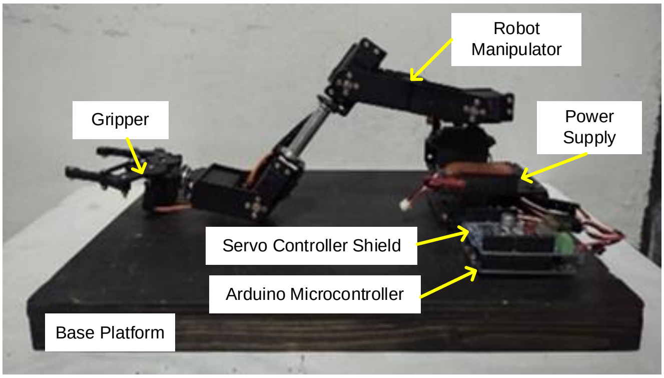

Figure 2: A 6-DOF robot manipulator is used in this work

The main controller used in this robot manipulator design is the Arduino Uno microcontroller, which is connected to 8-channels RC Servo Controller Shiel, whose electronic hardware schematic is shown in Figure 3. Figure 4 shows the MG996R servo motor used to control the movement of the robot's joints with the detailed specification given in Table 1. The servo motor used is a DC brushed type motor that is controlled by the internal integrated circuit for greater accuracy, improved bandwidth, and centering. There are 6 servo motors in total used in this 6-DOF robot manipulator, which are designated to control the movement of the base, shoulder, elbow, hand, wrist, and gripper. All of these servo motors are controlled through the servo motor shield, which is shown in Figure 5.

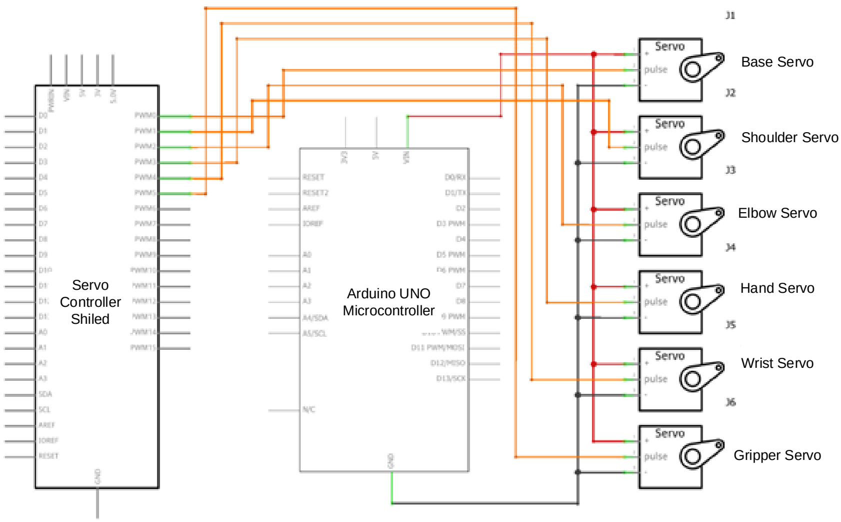

Figure 3: The schematic diagram of the 6-DOF robot manipulator

Figure 4: The MG996R servo motor used in this work

Table 1: The specifications of the MG996R servo motor

Description	Specification
Dimension	40.7 x 19.7 x 42.9 mm approx.
Stall torque	11 kgf.cm (6 V)
Operating speed	0.17 s/60º (4.8 V), 0.14 s/60º (6 V)
Operating voltage	4.8 V a 7.2 V
Running Current	500 mA –
Stall Current	2.5 A (6 V)
Dead bandwidth	5 µs

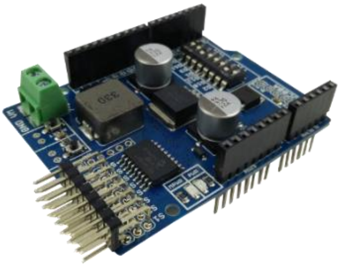

Figure 5: The 8-channels servo motor shield is used in this work

The servo motor shield can control up to eight independent servo motors on a single board via an integrated 5 V, and 5 A switching regulator. Each servo signal pin can generate servo pulses with a duty cycle range from 0.5 ms to 2.5 ms, which exceeds the range of other similar servos to operate in their full 180-degree. The shield is widely used and compatible with most of the Arduino microcontroller series. The detailed specifications of this servo can be found in Table 2.

Table 2: The parameters of the MG996R servo motor

Parameters	Details
Dimension	7.4 cm x 6.6 cm
Peak Current	5 A
Continuous Current	4 A
Operating voltage	7 V to 25 V
Logic level	5 V
Pulse Range	0.5 ms to 2.5 ms
Default baud rate	9600
Channels	8

Inverse kinematic equations

The kinematic model of this robot manipulator is shown in Figure 6. Inverse kinematics analysis is used to calculate the joint angles required to achieve the desired position and orientation in Cartesian space. The inverse calculations are represented by the following Equations (1) to (10) by using the trigonometric calculation. According to Figure 3, the $\theta_1, \theta_2, \theta_3$ and θ_4 servo rotation at Joint 1, Joint 2, Joint 3, and Joint 4. The l_1, l_2, l_3, and l_4 is the length of Link 1, Link 2, Link 3, and Link 4.

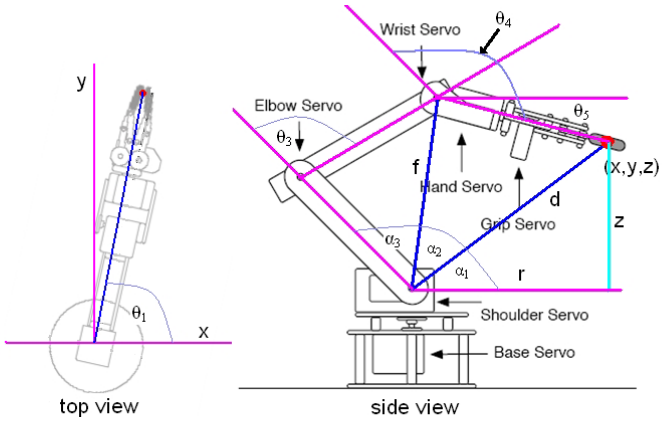

Figure 6: Kinematic model of the 6 DOF robot manipulator

$$\theta_1 = tan^{-1}\frac{y}{x} \tag{1}$$

$$\theta_1 = \alpha_1 + \alpha_2 + \alpha_3 \tag{2}$$

$$\theta_3 = \alpha_3 + cos^{-1}\left|\frac{{l_1}^2 - {l_2}^2 - f^2}{-2fl_2}\right| \tag{3}$$

$$\theta_4 = 90 + \theta_3 - \alpha_2 - \alpha_3 - \theta_x \tag{4}$$

$$d = \sqrt{x^2 + y^2 + z^2} \tag{5}$$

$$r = \sqrt{x^2 + y^2} \tag{6}$$

where d and r are the magnitudes of the position vector of that respective Joint 1. The α_1, α_1 and α_1 are the computed angles parallel to Joint 4 and Joint 5. These angles are useful in the determination of the θ_x when the end effector moves to the target position.

$$\alpha_1 = tan^{-1}\frac{z}{r} \tag{7}$$

$$\alpha_2 = cos^{-1}\left|\frac{{l_3}^2 - f^2 - d^2}{-2fd}\right| \tag{8}$$

$$\alpha_3 = cos^{-1}\left|\frac{{l_2}^2 - {l_1}^2 - f^2}{-2fl_1}\right| \tag{9}$$

$$\theta_x = cos^{-1}\left|\frac{f^2 - {l_3}^2 - d^2}{-2dl_3}\right| \tag{10}$$

Software design

The 6-DOF robot can be continuously moving or be manually controlled using the graphical user interface (GUI) that has been created using the Microsoft Visual Studio as shown in Figure 7. By using this GUI, the user will key in the desired coordinate of the end-effector which consists of x, y, and z-coordinates through the specified serial communication port. The *Test Connection* button tests whether the serial comport is presented or not. If it is presented, an LED is illuminated. Users must make sure that the correct comport is connected to the targeted Arduino microcontroller for their robot manipulator application.

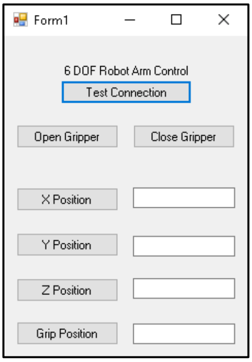

Figure 7: The GUI for controlling the 6 DOF robot manipulator is designed using the Microsoft Visual Studio

The *Open Gripper* button is used to spread the gripper of the end-effector to its maximum opening. Meanwhile, the *Close Gripper* button is used to reduce the distance between the grippers of the end-effector. The desired position of the end effector of the robot manipulator can be determined by the user by filling up the *X Position*, *Y Position*, and *Z Position* boxes to set the x, y, and z-coordinate, accordingly. Finally, the *Grip Position* is used to adjust the rotation of the end effector.

Phase 2: The fuzzy logic controller (FLC) design

A simulation of the Fuzzy Inference System (FIS) is created using the fuzzy logic toolkit in MATLAB. As previously mentioned, the FIS consists of the Fuzzification, the base knowledge, the computation rule, and the defuzzification processes [27]. Figure 8 shows the 3D surface of the FLC in MATLAB. This simulation is aimed to test the rule of the described fuzzy sets in controlling the position of the servomotors based on its error. The knowledge base contains a set of rules which construct the fuzzy rules. A summary of the fuzzy rules is in Table 3.

Table 3: Summary of the fuzzy rules in one servo motor

Δe\ e	**NL**	**NS**	**Z**	**PS**	**PL**
N	NL	NS	NS	PS	PL
Z	NL	NS	Z	PS	PL
P	NL	NS	PS	PS	PL

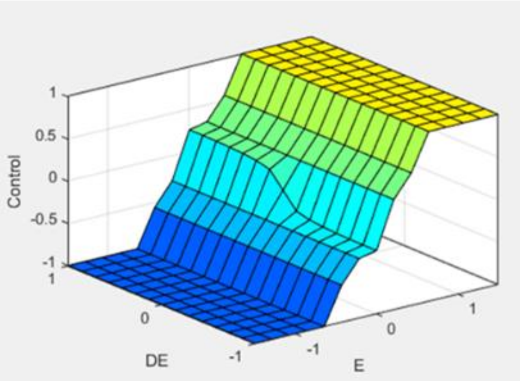

Figure 8: The 3-D FLC surface using MATLAB

According to Figure 8, the FIS has two inputs and one output. The two inputs are the error (E) and delta error (DE). While on the other hand, a control signal is the output of the plant that will energize the servomotor to reduce the error. The fuzzy membership functions for the two input parameters E and DE are shown in Figure 9 and Figure 10, respectively. Figure 11 shows the membership function for the output control signal. According to Figure 9, There are five subsets for the membership functions, namely, (1) NL-negative large, (2) NS-negative small, (3) Z-zero, (4) PS-positive small, and (5) PL-positive large. If a negative value is received, it means that the position of the servomotor should be added more to achieve the desired angle. Otherwise, it should be reduced. Meanwhile, in Figure 10, the delta error (DE) receives two obvious subsets of the membership functions, namely, (1) N-negative, and (2) P-positive. This input determines the direction in which the servomotors should be moving, either clockwise or counterclockwise in adjusting their position towards the desired location. There are five subsets in the membership function of the output signal. The subsets determine the range of the angle should be increased or decreased at the servomotors based on the received inputs from E and DE.

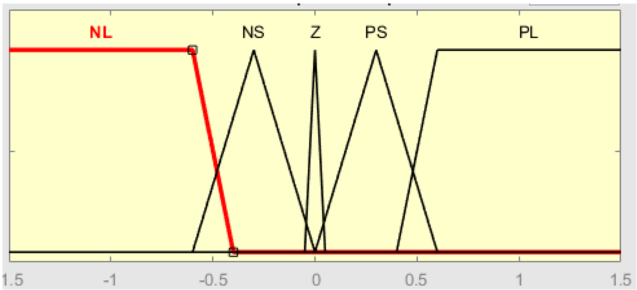

Figure 9: Membership function of error (E) as input

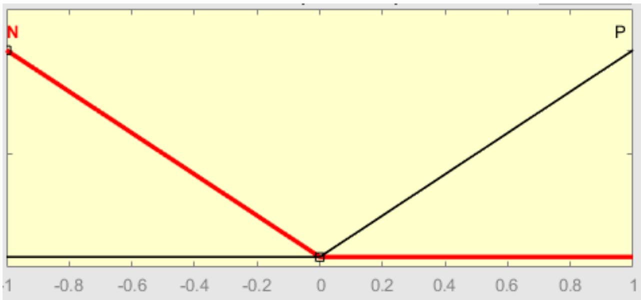

Figure 10: Membership function of delta error (DE) as input

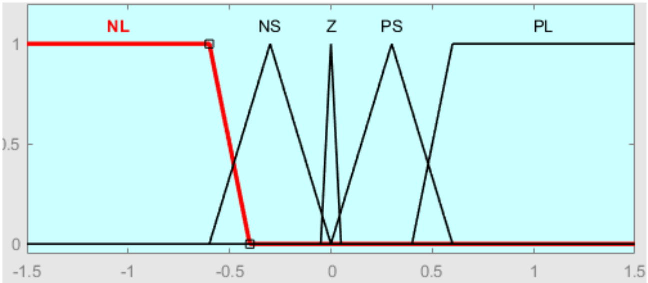

Figure 11: Membership function of CONTROL as output

To translate the FIS designed in the MATLAB to the microcontroller, the generated fuzzy design can be saved as a *.fis* file, which will be converted to an Arduino file format, *.ino* by adopting the Arduino FIST as shown in Figure 12. The *.ino* file package contains all the forms of a template file including the results of the preceding MATLAB rules design. Furthermore, as the data from the rules output results are inserted in the program code line, the execution of the physical conversion to C Arduino results in a fuzzy logic program template is simplified. In this work, all of the servomotors attached to the robot manipulator are controlled by the same FLC since they are possessing similar characteristics.

Figure 12: MATLAB -FIS used to convert the Matlab code to Arduino C code

<u>Overall control algorithm</u>

In this work, the desired position of the end-effector is defined by the user via the designed GUI to start the process as shown in Figure 13. The angle for each servo motor is determined through the inverse kinematic Equations (1) – (10). Then, the computed angles are sent to the Arduino IDE through the serial communication port and the movement of the servo motors is controlled by the Fuzzy Inference System (FIS) that has been created early in

MATLAB. With this, the end effector is guided to the desired coordinate [28].

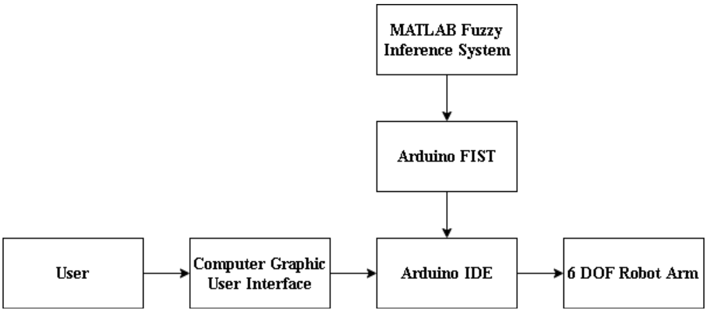

Figure 13: The program flow of the 6-DOF robot manipulator

Phase 3: performance validation

Using the average error and Mean Absolute Error (MAE), the performance of the robot manipulator is compared between the system with and without the FLC at this stage.

Average error

The average error of this experimental work is calculated by dividing the total error score for each joint. To compute this, Equation (11) is used.

$$Average\ Error = \frac{\sum_{i=1}^{n}(y_i - x_i)}{n} \tag{11}$$

with, y_i= desired output
x_i = true output
n = number of data point

Mean Absolute Error (MAE)

In this work, the Mean Absolute Error (MAE) is used to compare the performance of the servo motor position between the without-FLC and with-FLC robot manipulators. The MAE is calculated using Equation (12) between the three axes (X, Y, and Z) to determine the average error.

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} \text{ x } 100 \tag{12}$$

with, y_i = desired output
x_i = true output
n = number of data point

Result and Discussion

In this discussion, only Joint 1 (base), Joint 2 (shoulder), Joint 3 (elbow), Joint 4 (hand), and Joint 5 (wrist) are considered, as only their movement can alter the end position of the end-effector. Please note that the experimental work is performed without any object at the end-effector of the robot manipulator. The desired angle of each joint is compared with the actual angle received from the Arduino microcontroller through an algorithmic encoder as the feedback sensor. The angle position of each servo motor is tested for each angle, ranging from 0° to 180° with and without the FLC. Table 4 shows the outcome of the test for each joint, which is graphically illustrated in Figures 14 and 15, respectively. The MAE comparison between the robot manipulator with and without FLC is tabulated in Table 5.

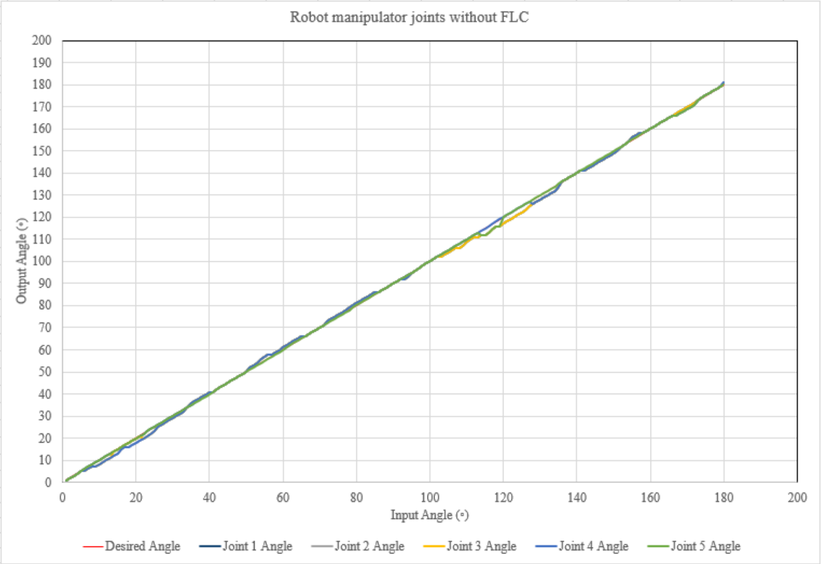

Figure 14: The output angle of each joint without the FLC

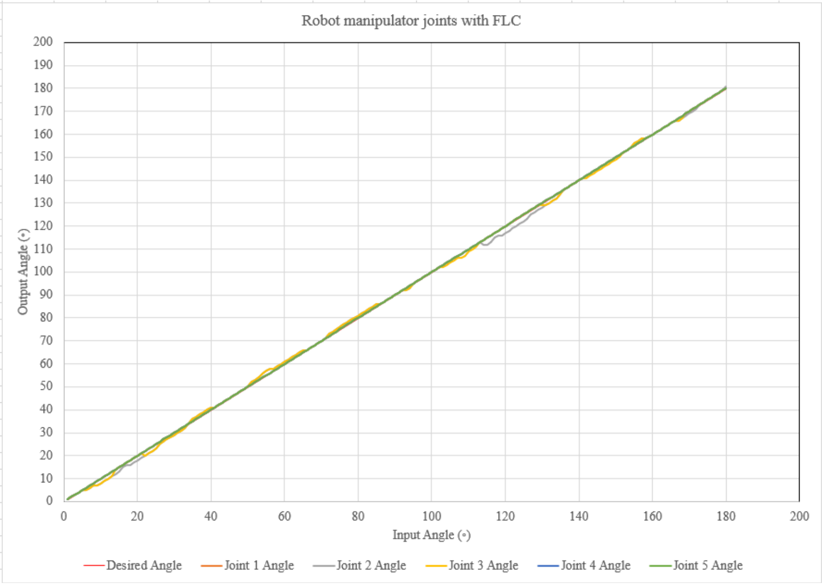

Figure 15: The output angle of each joint with the FLC

Table 4: The error analysis of each joint of the robot manipulator with and without the FLC

	Average Error (°)				
	Joint 1 (Base)	**Joint 2 (Shoulder)**	**Joint 3 (Elbow)**	**Joint 4 (Hand)**	**Joint 5 (Wrist)**
Without FLC	0.233272	0.441989	0.411295	0.208717	0.128913
With FLC	0	0.380602	0.147330	0	0

Table 5: The MAE percentage of the last position of the end-effector with and without the FLC

System	**MAE (%)**
Without FLC	1.424187
With FLC	0.527931

From Table 4, without FLC, each joint shows an error in the angle position of the servomotor. Even though the error is significantly small, it cannot be neglected for an application that concerns high precision and accuracy such as a precision manufacturing application. The MAE percentage for a robot manipulator without an FLC is approximately 1.42

percent. This error is reduced to 0.53 percent when an FLC is injected into the robot manipulator to control the angular position of each joint.

However, from the result, it can be noticed that Joint 2 and 3 have the greatest discrepancies between the desired and the obtained coordinates. This plausible effect is caused by the fact that both motors are mainly used to support the robot manipulator's arm weight during the upper and lower movements. Plus, the material used for building the frame of the robot manipulator also contributed to this effect. As the robot manipulator's arm is fully extended horizontally, the generated torque caused by the weight of this arm is extremely high, which is depicted in Figure 16. Additionally, when the robot is completely loaded, the second axis arm experiences increased tension, moving it closer to the spinning axis [29]. This issue can be resolved by removing unwanted weight from the robot arm such as excessive wires or using lightweight material, gradually decreasing the size of the succeeding servo motor to the end effector, or by using a stronger servo motor in the second axis.

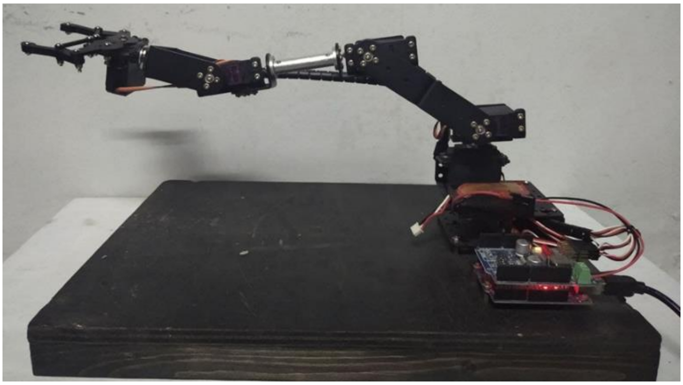

Figure 16: Horizontally extended robot manipulator's arm

Since FLC reads the real-time value of the servo motor's movement rather than simply providing the desired position, the implementation of FLC led to a reduction in vibration. The user only needs to input the desired coordinate for the end effector to move, as well as a few other inputs to control the grip orientation, opening, and closing.

Conclusion

The Mamdani fuzzy controllers have greater error-reduction stability. The Mamdani FLC proposed in this work can effectively eliminate hazardous oscillations and provide smooth operation during the transition period. The results indicate that the proposed fuzzy logic-based controller design is a superior option for the next robot design implementation. The GUI also improves the usability of the 6-DOF robot arm, as the user only needs to input the desired end-effector coordinate instead of controlling each servo independently.

Acknowledgment

Thanks to the College of Engineering, Universiti Teknologi MARA for the facilities, guidance, and financial support to complete this project.

References

[1] M. Nakamura, S. Goto, and N. Kyura, *Mechatronic Servo System Control*, 1st ed. Heidelberg: Springer , 2004.

[2] J. F. Silva and S. F. Pinto, "Linear and Nonlinear Control of Switching Power Converters," *Power Electron. Handb.*, pp. 1141–1220, Jan. 2018, doi: 10.1016/B978-0-12-811407-0.00039-8.

[3] P. N. Paraskevopoulos, *Modern control engineering*, 1st ed. CRC Press, 2017.

[4] M. M. Gouda, S. Danaher, and C. P. Underwood, "Fuzzy Logic Control Versus Conventional PID Control for Controlling Indoor Temperature of a Building Space," *IFAC Proc. Vol.*, vol. 33, no. 24, pp. 249–254, Sep. 2000, doi: 10.1016/S1474-6670(17)36900-8.

[5] J. Jalani and S. Jayaraman, "Design a Fuzzy Logic Controller for a Rotary Flexible Joint Robotic Arm," *MATEC Web Conf.*, vol. 150, pp. 1–6, Feb. 2018, doi: 10.1051/MATECCONF/201815001011.

[6] N. H. Nam, D. Van Huyen, and N. V. Hung, "An arduino based pid fuzzy controller for a robot arm," *Sci. Technol. Mag.*, vol. 151, no. 06, pp. 179–185, 2016.

[7] D. D. Ligutan, L. J. S. Cruz, M. C. D. P. Del Rosario, J. N. S. Kudhal, A. C. Abad, and E. P. Dadios, "Design and implementation of a fuzzy logic-based joint controller on a 6-DOF robot arm with machine vision feedback," *Proc. Comput. Conf. 2017*, pp. 249–257, Jan. 2017, doi: 10.1109/SAI.2017.8252111.

[8] P. Sarkhel, N. Banerjee, and N. B. Hui, "Fuzzy logic-based tuning of PID controller to control flexible manipulators," *SN Appl. Sci.*, vol. 2, no. 6, pp. 1–11, 2020, doi: 10.1007/S42452-020-2877-Y/TABLES/5.

[9] S. Al-Marzoqi and A. El-Bayoumy Mansour, "Comparative Study of Different Control Techniques Applied on 6 DOF Robot Arm," *J. Al Azhar Univ. Eng. Sect.*, vol. 12, no. 42, pp. 25–36, 2017.

[10] J. Jalani and S. Jayaraman, "Design a Fuzzy Logic Controller for a Rotary Flexible Joint Robotic Arm," *MATEC Web Conf.*, vol. 150, pp. 01011, Feb. 2018, doi: 10.1051/MATECCONF/201815001011.

[11] N. Dersarkissian, R. Jia, and D. L. Feitosa, "Control of a two-link robotic arm using fuzzy logic," *IEEE Int. Conf. Inf. Autom.*, pp. 481–486, Aug. 2018, doi: 10.1109/ICINFA.2018.8812444.

[12] S. Kumar, K. Rani, and V. K. Banga, "Robotic Arm Movement Optimization Using Soft Computing," *IAES Int. J. Robot. Autom.*, vol. 6, no. 1, pp. 1, Mar. 2017, doi: 10.11591/IJRA.V6I1.PP1-14.

[13] N. Nagpal, B. Bhushan, and V. Agarwal, "Intelligent control of four DOF robotic arm," *1st IEEE Int. Conf. Power Electron. Intell. Control Energy Syst.*, pp. 1–6, Feb. 2017, doi: 10.1109/ICPEICES.2016.7853357.

[14] A. A. Mohamed Ali, "Improving the Accuracy of Robotic Arm position using Fuzzy Logic Controller," Sudan University of Science and Technology, Sudan, 2019.

[15] C. Yu and S. Li, "Research on Adaptive Control System of Robot Arm with Six Degrees of Freedom," *Int. Conf. Sens. Netw. Comput. Eng.*, pp. 666–671, 2016, doi: 10.2991/ICSNCE-16.2016.129.

[16] W. S. Mada Sanjaya *et al.*, "Colored Object Sorting using 5 DoF Robot Arm based Artificial Neural Network (ANN) Method," *J. Phys. Conf. Ser.*, vol. 1090, no. 1, pp. 012070, Sep. 2018, doi: 10.1088/1742-6596/1090/1/012070.

[17] M. Khairudin, H. A. Wijaya, and Muslikhin, "Converter matlab fuzzy inference to arduino C system," *J. Phys. Conf. Ser.*, vol. 1456, no. 1, pp. 012010, Jan. 2020, doi: 10.1088/1742-6596/1456/1/012010.

[18] A. Hamam and N. D. Georganas, "A comparison of mamdani and sugeno fuzzy inference systems for evaluating the quality of experience of hapto-audio-visual applications," *IEEE Int. Work. Haptic Audio Vis. Environ. Games Proc.*, pp. 87–92, 2008, doi: 10.1109/HAVE.2008.4685304.

[19] T. P. Singh, P. Suresh, and S. Chandan, "Forward and Inverse Kinematic Analysis of Robotic Manipulators," *Int. Res. J. Eng. Technol.*, vol. 4, no. 2, pp. 1459–1469, 2017

[20] T. Dewi, S. Nurmaini, P. Risma, Y. Oktarina, and M. Roriz, "Inverse kinematic analysis of 4 DOF pick and place arm robot manipulator using fuzzy logic controller," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 1376–1386, 2020, doi: 10.11591/IJECE.V10I2.PP1376-1386.

[21] A. H. Mary, T. Kara, and A. H. Miry, "Inverse kinematics solution for robotic manipulators based on fuzzy logic and PD control," *Al-Sadiq Int. Conf. Multidiscip. IT Commun. Tech. Sci. Appl. AIC-MITCSA 2016*, pp. 160–165, Nov. 2016, doi: 10.1109/AIC-MITCSA.2016.7759929.

[22] S. N. Gai, R. Sun, S. J. Chen, and S. T. Ji, "6-DOF Robotic Obstacle Avoidance Path Planning Based on Artificial Potential Field Method,"

16th Int. Conf. Ubiquitous Robot. UR, pp. 165–168, Jun. 2019, doi: 10.1109/URAI.2019.8768792.

[23] S. A. Ajwad, J. Iqbal, R. U. Islam, A. Alsheikhy, A. Almeshal, and A. Mehmood, "Optimal and Robust Control of Multi DOF Robotic Manipulator: Design and Hardware Realization," *Cybern. Syst.*, vol. 49, no. 1, pp. 77–93, 2018, doi: 10.1080/01969722.2017.1412905.

[24] R. K. Megalingam, N. Katta, R. Geesala, P. K. Yadav, and R. C. Rangaiah, "Keyboard-based control and simulation of 6-DOF Robotic Arm using ROS," *4th Int. Conf. Comput. Commun. Autom.*, pp. 1–5, Dec. 2018, doi: 10.1109/CCAA.2018.8777568.

[25] X. Yin and L. Pan, "Direct adaptive robust tracking control for 6 DOF industrial robot with enhanced accuracy," *ISA Trans.*, vol. 72, pp. 178–184, Jan. 2018, doi: 10.1016/J.ISATRA.2017.10.007.

[26] M. Fanni and A. Khalifa, "A new 6-DOF quadrotor manipulation system: Design, kinematics, dynamics, and control," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1315–1326, Jun. 2017, doi: 10.1109/TMECH.2017.2681179.

[27] R. Tsutada, T.-T. Hoang, and C.-K. Pham, "An Obstacle Avoidance Two-Wheeled Self-Balancing Robot," *Int. J. Mech. Eng. Robot. Res.*, vol. 11, no. 1, pp. 1–7, 2022, doi: 10.18178/ijmerr.11.1.1-7.

[28] N. Elias and N. Mat Yahya, "Fuzzy Logic Controller Optimized by MABSA for DC Servo Motor on Physical Experiment," *Recent Trends Mechatronics Towar. Ind.*, vol. 730, pp. 561–570, 2022, doi: 10.1007/978-981-33-4597-3_51.

[29] M. Bugday and M. Karali, "Design optimization of industrial robot arm to minimize redundant weight," *Eng. Sci. Technol. an Int. J.*, vol. 22, no. 1, pp. 346–352, Feb. 2019, doi: 10.1016/j.jestch.2018.11.009.