

Optimizing Membership Function Tuning for Fuzzy Control of Robotic Manipulators using PID-Driven Data Techniques

Phichitphon Chotikunnan¹, Rawiphon Chotikunnan^{2*}, Anuchit Nirapai³, Anantasak Wongkamhang⁴,
Pariwat Imura⁵, Manas Sangworasil⁶

^{1, 2, 3, 4, 5, 6} College of Biomedical Engineering, Rangsit University, Pathum Thani, Thailand

Email: ¹ phichitphon.c@rsu.ac.th, ² rawiphon.c@rsu.ac.th, ³ anuchit.n@rsu.ac.th,

⁴ anantasak.w@rsu.ac.th, ⁵ pariwat.i@rsu.ac.th, ⁶ manas.s@rsu.ac.th

*Corresponding Author

Abstract— In this study, a method for optimizing membership function tuning for fuzzy control of robotic manipulators using PID-driven data techniques is presented. Traditional approaches for designing membership functions in fuzzy control systems often rely on the experience and knowledge of the system designer, which can lead to suboptimal performance. By utilizing data collected from a PID control system, the proposed method aims to enhance the precision and controllability of robotic manipulators through improved fuzzy logic control. A Mamdani-type fuzzy logic controller was developed and its performance was simulated in Simulink, demonstrating the effectiveness of the proposed optimization technique. The results indicate that the method can outperform conventional P control systems in terms of overshoot reduction while maintaining comparable transient response specifications. This research highlights the potential of the PID-driven data-based approach for optimizing membership function tuning in fuzzy control systems and offers valuable insights for the development and evaluation of fuzzy logic control in robotic manipulators. Future work may focus on further optimization of the tuning process, evaluation of system robustness under various operating conditions, and exploring the integration of other artificial intelligence techniques for improved control performance.

Keywords—Fuzzy Logic Control; PID Control; Robotic; Simulation

I. INTRODUCTION

Robotic arms play a crucial role in various industries, including healthcare, by performing a wide range of tasks. One of the main challenges in controlling these systems is minimizing errors during arm movement towards a desired point. In practice, repetitive arm movements from the same position can result in repeated errors with a range of constant values across cycles. To address this issue, a theoretical approach is necessary to improve the movement of the arm from one location to another [1].

Previous research has demonstrated the usefulness of controlling robotic arms in automating welding processes for stainless steel and commercially pure titanium [2]. Various robot motion control systems have been developed to regulate the movement of robotic arms [3]. This study utilized GrblGru to construct a 5-axis robotic arm and studied its operation using the basic G-code instruction set [4]. Other tools, such as MATLAB GUI and Arduino IDE, can also be

used to control small robots, such as in simulations of a robot picking up an object [5]. This research may have implications for developing a robotic system for collecting COVID-19 specimens to assist medical personnel in the future [6].

Developers of mechanical robotic arms design both mechanical and motor control systems [6]. Motor control design is a vital aspect of many industrial and mechanical applications. Research efforts focus on developing motor control systems that can respond quickly, accurately, and with minimal overshoot, typically using DC servo motors [7], DC motor control [8]-[11], BLDC motors [12]-[15], and other electronic control systems. Commonly used control methods include PID control systems [16]-[21], the Ziegler-Nichols method [22]-[25], and the CHR tuning method [26],[27]. Improvements in PID control have been made in controlling robotic arms, such as using the Fuzzy Neural Network Algorithm [28] to design the gain of PID control [29] and adjusting the input to the system to increase flexibility. Other controllers used to improve the internal control system of robotic arms include adaptive fuzzy systems or developed fuzzy PID controller systems [30]-[34], optimal control [35]-[40], adaptive sliding mode [41]-[46], and adaptive neural network tracking control [47]-[60].

In controlling a robotic arm, the operator must define the motion path for the arm. Two main types of control are used in this process: point-to-point control [61]-[65] and trajectory tracking control [66], [67], each with its advantages and disadvantages. Repetitive controls [68] are a common aspect of robotic arm control, in which the same task is performed repeatedly. However, this can result in errors in the movement of the robotic arm due to the repetition of motion. To address this issue, iterative learning control [68]-[75], repetitive control, and run-to-run control are often utilized in the system design.

Fuzzy logic type 2 has only been applied in the balancing of a wheelchair [76], but it has demonstrated flexibility in control and has been used to adjust the gain in other control systems such as sliding mode controllers [77] and PID control systems [78], [79]. Fuzzy logic, in general, has a wide range of applications, including designing controllers for regulating the rotation of motors [80], electric fans [81], quadrotors [82],



[83], Internet of Things (IoT) systems [84], and medical decision-making [85].

Finding the optimal membership function for fuzzy logic controllers is crucial in controller design, but achieving accuracy using personal expertise or data analysis alone can be difficult. Various techniques have been developed to address this issue, including the KNN Classifier [86], a control algorithm (CA) module, a learning algorithm (LA) module, an artificial neural network (ANN) algorithm [87], and a genetic algorithm [88]. Each technique has its own strengths and limitations, and designers must consider the information used for learning and membership function design when choosing a method. Simulink is a tool that can be used to design controllers for linear, non-linear, continuous, or discrete systems.

This study proposes a method for designing the membership functions of a fuzzy logic controller using a PID-driven data acquisition technique and multiple polynomial regression with bisquare weights. This approach has been demonstrated to be effective in identifying suitable membership functions for the robot system being controlled, and it can be applied in practical applications. The performance of the proposed fuzzy logic controller is compared with P control, showing that the former outperforms the latter in terms of accuracy and efficiency, making it a viable option for controlling robot systems.

II. METHOD

A. Designing System

Fuzzy logic control system design can be challenging, particularly in designing or configuring membership functions. However, this problem can be addressed through membership function tuning by observing the behavior of the system using a closed control system, and then using the obtained value to design the membership function.

An easier way to collect the behavior of the system while it is being controlled is to estimate the value of the membership function in the fuzzy logic system using the value obtained from controlling the original system. The membership function tuning process typically involves designing a closed feedback system first, followed by taking the difference between the error value and the control input of the system and normalization. Based on these values, a membership function is designed in the fuzzy logic system.

If the designed membership function can control the system stably, it can be applied to the system; otherwise, the storage conditions must be redesigned. Fig. 1 provides a block diagram of the overall membership function tuning process. The workflow in this research involves constructing a closed loop system, inserting a setpoint signal, storing and normalizing the obtained parameters from the system response, and checking the stability of the membership function in controlling the system. If the membership function is found to be unstable in the controlling the system, the setpoint is redesigned, and the iteration process is repeated to determine the optimal membership function tuning that aligns with the response of the system. Once stability is achieved, the membership function is applied to

the designed system, completing the process of membership function tuning.

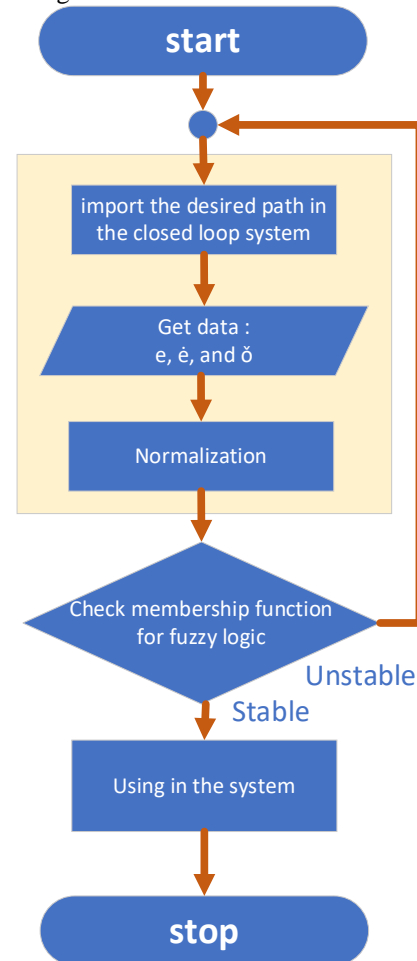


Fig. 1. Block diagram of the overall membership function tuning.

Fig. 2 illustrates the process of creating the membership function table, which is an extension of Fig. 1 in terms of normalization. The process will end once the membership function table meets the acceptable criteria. The proposed method can be applied to any tuning method or technique in theory.

To obtain the membership function of the system, the working process involves several steps described in the topic of optimizing membership function tuning, which consists of seven steps in total. First, a feedback control system is designed. Second, the signal received from the feedback control system is stored. Third, the maximum and minimum values are determined. Fourth, the dataset is set to be within the range of -100% to 100% . Fifth, the criteria for rounding the decimal point are established. Sixth, the dataset level is specified in the membership function. And finally, the membership function is constructed for the fuzzy logic system.

If the designed system tests the membership function and stabilizes the system, the operation is considered complete. If not, the system control conditions must be modified or adjusted to the polynomial regression equation.

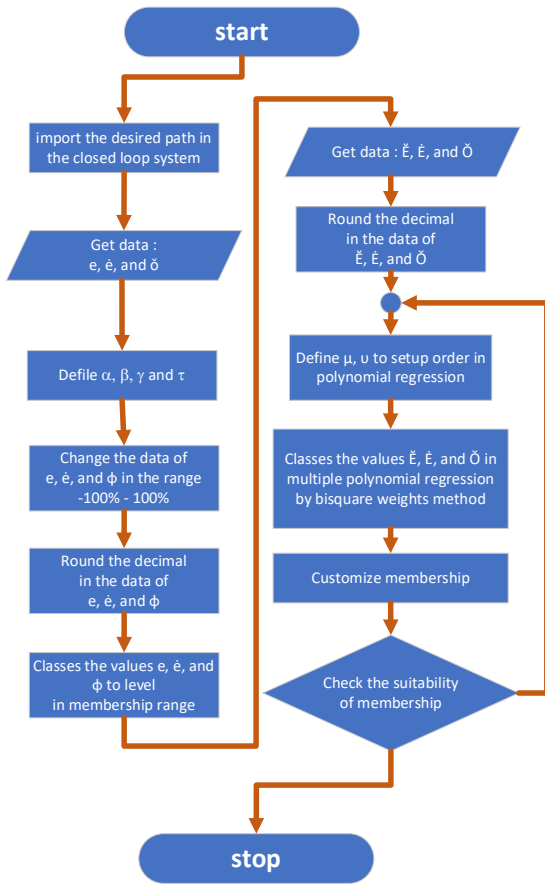


Fig. 2. Program flowchart of normalization.

B. Robotic manipulator Seiko D-Tran RT3200

The Seiko D-Tran RT3200 robot is a cylindrical type robot that is widely used in various industrial applications. Fig. 3 shows an example of this robot along with its controller device.

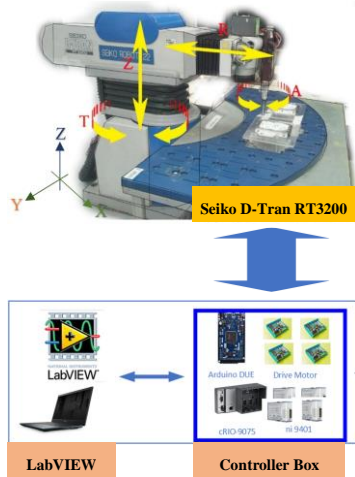


Fig. 3. Seiko D-Tran RT3200 and device for controller.

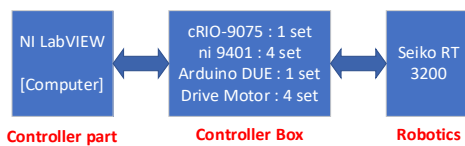


Fig. 4. Block diagram of the overall system.

The Seiko D-Tran RT3200 robot used in this research, as shown in Fig. 3 and Fig. 4, has a system equation designed as a discrete time with a sampling rate of 0.055 seconds. Previous studies have been conducted on the control of this robot, including works by P. Chotikunnan and B. Panomruttanarug [74], P. Chotikunnan et al. [75], and P. Chotikunnan and R. Chotikunnan [89].

These studies present a model of the system equation denoted as (1), with the coefficients listed in Table 1 representing the variables of the robotic arm.

$$P(z) = \frac{\gamma_1 z}{z^2 + \beta_1 z + \beta_0} \tag{1}$$

TABLE I. PARAMETERS USED IN THE OPEN-LOOP SYSTEM.

Joint	γ_1	β_1	β_0
Joint R	0.0333	-1.6871	0.6884
Joint T	0.0162	-1.7077	0.7111
Joint Z	0.0140	-1.7519	0.7526

III. PID CONTROLLER DESIGN

In the development of closed motor control systems [90]-[94], controlling the response of the system is a crucial aspect. The Proportional-Integral-Derivative (PID) control system [95]-[98] has been continuously improved and widely used. The PID control system utilizes the PID controller, as shown in Fig. 5, and is well known for its simple structure and effective performance in various control applications. It consists of three sub controllers, the Proportional (P) controller, the Integral (I) controller, and the Derivative (D) controller. Although the PID control system operates as a continuous-time controller in theory, it needs to be adapted to a discrete-time controller when implemented in microcontroller systems. The control input estimate to the system is represented by (2).

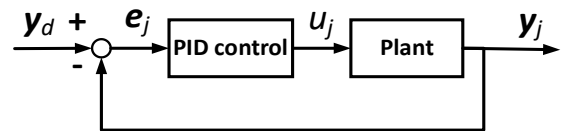


Fig. 5. Block diagram of PID controller

$$u_j(k) = K_p e_j(k) + K_i \sum_{k=N}^k e_j(k) + K_d (e_j(k) - e_j(k-1)) \tag{2}$$

In this study, the control inputs, denoted as u_j , consist of the proportional gain (K_p), integral gain (K_i), and derivative gain (K_d), as shown in the basic structure of feedback control in Fig. 6. The control input is restricted to a range of -100 to 100, and the discrete transfer function equation is utilized to determine the system. The saturation of the control input is also taken into account.

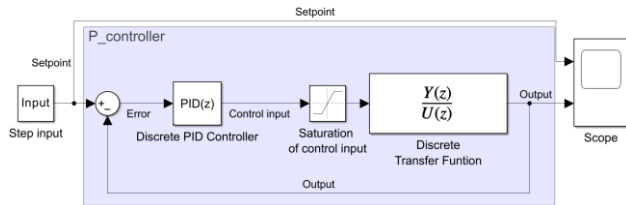


Fig. 6. PID control based system control program using simulink.

In this study, a PID controller was employed to regulate the motor system in joints R, T, and Z using CHR tuning to achieve minimal overshoot. The system design was investigated by P. Chotikunnan and R. Chotikunnan [89]. The R values or R conditions for joint R, joint T, and joint Z were found to be 12.75, 21.33, and 20.11, respectively. These values indicate that the P controller system should be utilized for control because the R value is greater than 10. The gain for each axis of the PID system as a P controller system is presented in Table 2 for joint R, joint T, and joint Z.

TABLE II. P GAIN VALUES OF JOINT R.

Joint	Joint R	Joint T	Joint Z
P controller	4.25	8.00	6.70

IV. FUZZY LOGIC CONTROL SYSTEM

This study aims to design a fuzzy logic control system that regulates the motion of a robotic arm motor using the Mamdani method to estimate the control input. The system control program is simulated in Simulink, as shown in Fig. 11. The design rules for input signal 1, input signal 2, and output signal are defined in this study. For example, input signal 1 represents the system error signal with 5 accompanying rules, while input signal 2 represents the difference value of the error signal with 5 rules. Both inputs range from -1 to 1 units, while the output signal has 9 rules defining its range also from -1 to 1 units. The membership function of input 1 and input 2 for all 5 rules and the output for all 9 rules are established and presented in Fig. 8, Fig. 9, and Fig. 10, respectively.

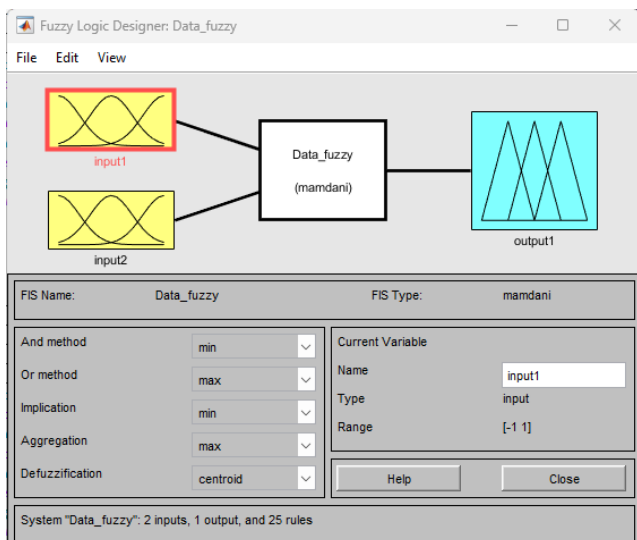


Fig. 7. Fuzzy logic designer.

Finally, the functionality of the designed membership function table is tested using the fuzzy inference method and

the Mamdani fuzzy rule system method, which is determined after optimizing the membership function tuning.

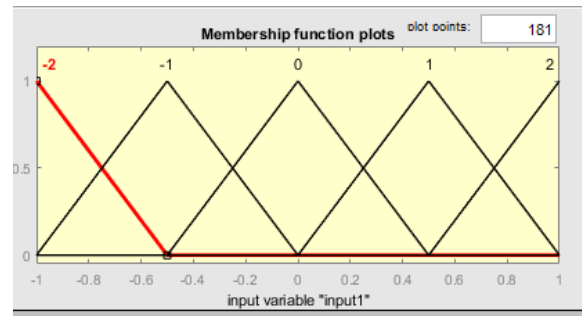


Fig. 8. Membership function of input 1.

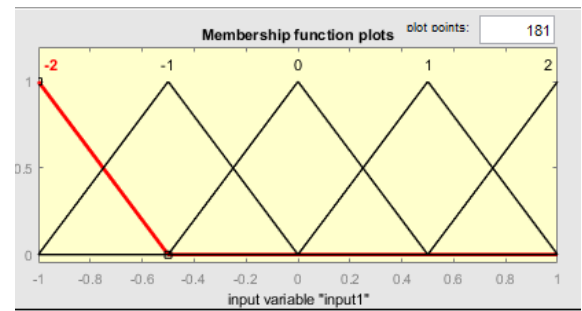


Fig. 9. Membership function of input 2.

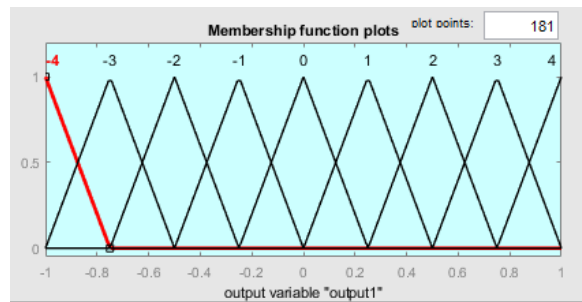


Fig. 10. Membership function of output 1.

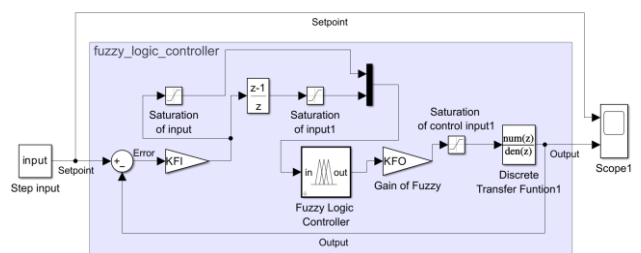


Fig. 11. The program in Simulink that uses fuzzy logic control to control the system.

In this study, the input values for the joints R, T, and Z are limited to a range of -1 to 1 due to the saturation of the input in Fig. 11. The parameter *KFI*, representing the error range, is calculated using (61).

The control input saturation limit for all 3 joints is set to -100 to 100, with a value of 100 for *KFO*. The values are scaled within the range of -1 to 1 for simplicity, and the output gain is set to 100. The power of the system can be adjusted within the range of -100% to 100% based on the operation of the system.

V. OPTIMIZING MEMBERSHIP FUNCTION TUNING

Membership function optimization is a crucial aspect of designing a stable fuzzy logic control system [99]-[101]. This study utilizes multiple polynomial regression with the bisquare weighting method to optimize membership function tuning and derive the membership function. The objective is to estimate the behavior of the closed system, which is controlled by the robot with the PID system, and subsequently evaluate the system. The process involves deriving values from the polynomial models' equations and determining the membership function tuning to obtain suitable results for the membership function table used in the fuzzy logic controller.

A. Polynomial models

Polynomial regression analysis is a type of regression analysis that is used to examine the relationship between one or more dependent variables (Y) and independent variables (X). When dealing with non-linear variables, it is necessary to follow the same preliminary steps, which involve collecting data to determine the dependent variable (Y) and the independent variable (X), and expressing this relationship in the form of (3).

$$y = \sum_{i=1}^{n+1} p_i x^{n+1-i} \quad (3)$$

The order of the polynomial is represented by $n + 1$ and the degree of the polynomial is represented by n .

This research is to determine the degree of the polynomial in $x = 3$ and $y = 3$ and to use the robust technique of the Bisquare weights method to estimate the data required to solve the equation in the polynomial system. A regression model of degree 3, also known as a quadratic model, in two variables x and y is given by (4).

$$f_{poly}(x, y) = \beta_0 + \beta_{10}x + \beta_{01}y + \beta_{11}xy + \beta_{20}x^2 + \beta_{02}y^2 + \beta_{21}x^2y + \beta_{12}xy^2 + \beta_{30}x^3 + \beta_{03}y^3 \quad (4)$$

The command function `fit([x,y],z, 'poly33', 'Normalize', 'Bisquare')` in the matlab program can be used to find the conditions where x is the value of input I, y is the value of input II, and z is the value of the output to be estimated. The value of β can be obtained from the fit function of the MATLAB program, which displays the parameters in the simulation results.

B. Membership function tuning

Training algorithms are utilized to determine the membership function of a system using data storage techniques in a control system that can control the behavior of the system. The preliminary data is analyzed in order to design a suitable membership function system through the execution of a search process for all 7 steps, as follows.

1. Design feedback control

A controller is created in this step to make the system that needs to be controlled a feedback control system. The PID control system can be used to easily adjust the system in order to find the membership function in the fuzzy logic system. This feedback control can be designed in the program MATLAB/Simulink as shown in Fig. 12. By setting the

signal setpoint, a smooth function signal in the form of a sin wave is designed to bring the signal value to the control system that needs the membership function of fuzzy logic and to collect the data from the feedback control system, which consists of the error (e), the derivative of the error (\dot{e}), and the control input (δ).

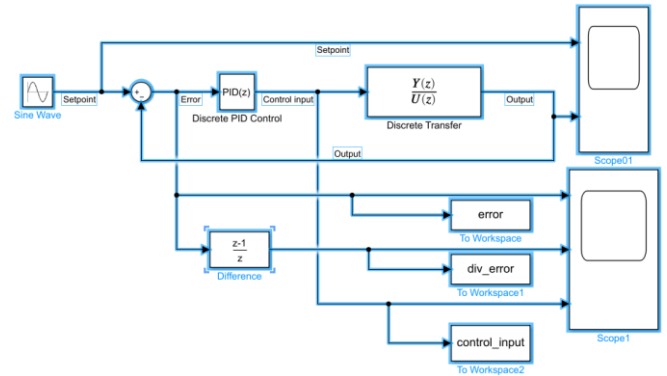


Fig. 12. Simulink was utilized to collect data from the feedback control.

2. Store the signal received from the feedback control system

The signal data or system behavior is stored in the data set of the desired path, actual path, error (e), derivative of error (\dot{e}), and control input (δ) as shown in Fig. 12. In order to use this data for estimation in finding the membership function, initial variables must be set for the estimation, including the number of membership functions input I (α), which is the total number of rules defined in input 1, the number of membership functions input II (σ), which is the total number of rules defined in input 2, and MB output (γ), which is the total number of rules defined in the output of the fuzzy logic to design the system. The analysis of the data system is not limited to a certain number of data sets, but if the data set is not sufficient, it may be necessary to increase the number of data for analysis. This data can be brought in at specific times for analysis.

3. Determine the maximum and minimum value

The data set contains error (e), derivative error (\dot{e}), and control input (δ). To analyze the amplitude of each data set and ensure it is within the specified limits, the maximum and minimum values can be obtained by using (5)-(10).

$$e_{max} = \max\{e\} \quad (5)$$

$$e_{min} = \min\{e\} \quad (6)$$

$$\dot{e}_{max} = \max\{\dot{e}\} \quad (7)$$

$$\dot{e}_{min} = \min\{\dot{e}\} \quad (8)$$

$$\delta_{max} = \max\{\delta\} \quad (9)$$

$$\delta_{min} = \min\{\delta\} \quad (10)$$

After obtaining the highest and lowest values, they are compared in order to find the highest value point as in (11), and then the highest value point in each data set is found as in (12)-(14).

$$M_x(x_{max}, x_{min}) = \begin{cases} x_{max}, & x_{max} \geq \|x_{min}\| \\ \|x_{min}\|, & x_{max} < \|x_{min}\| \end{cases} \quad (11)$$

$$M_e = M_x(e_{max}, e_{min}) \quad (12)$$

$$M_{\dot{e}} = M_x(\dot{e}_{max}, \dot{e}_{min}) \quad (13)$$

$$M_{\ddot{\delta}} = M_x(\ddot{\delta}_{max}, \ddot{\delta}_{min}) \quad (14)$$

4. Set the range of the dataset to be in the range -100%-100%

In this action is to establish a consistent range for all datasets within the design system for convenient usage. The data level can be transformed to fall within the range of -100% to 100% using (15), (16), and (17).

$$s_e = \frac{e}{M_e} \times 100 \quad (15)$$

$$s_{\dot{e}} = \frac{\dot{e}}{M_{\dot{e}}} \times 100 \quad (16)$$

$$s_{\ddot{\delta}} = \frac{\ddot{\delta}}{M_{\ddot{\delta}}} \times 100 \quad (17)$$

5. Establish the criteria for rounding the decimal point

After determining the range of a dataset to be -100% to 100%, a program can be developed to round off the decimal numbers using (18). To begin this process, the decimal portion must first be identified using (19), (20), and (21), and then the values can be entered into (22), (23), and (24) as a computational condition for the use of (18), which will produce the rounded result. In this study, the value of τ is set to 0.4, indicating the decimal point at which the program should truncate the fraction.

$$\rho(x) \begin{cases} 1, & x \geq \tau \\ -1, & (x > (-1)) \text{ and } (p_e \leq \tau) \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

$$p_e = s_e - \text{int}(s_e) \quad (19)$$

$$p_{\dot{e}} = s_{\dot{e}} - \text{int}(s_{\dot{e}}) \quad (20)$$

$$p_{\ddot{\delta}} = s_{\ddot{\delta}} - \text{int}(s_{\ddot{\delta}}) \quad (21)$$

$$\mu_e = \text{int}(s_e) + \rho(p_e) \quad (22)$$

$$\mu_{\dot{e}} = \text{int}(s_{\dot{e}}) + \rho(p_{\dot{e}}) \quad (23)$$

$$\mu_{\ddot{\delta}} = \text{int}(s_{\ddot{\delta}}) + \rho(p_{\ddot{\delta}}) \quad (24)$$

6. Specify the data set level in MB

To determine the level of a dataset, the procedure is to transform a dataset within the range of -100% to 100% into a dataset within a range that does not exceed the number of input membership functions I (α), the number of input membership functions II (σ), and the number of output membership functions (γ), which can be calculated using (25), (26), and (27).

$$l_e = \frac{\mu_e}{200} \times \alpha \quad (25)$$

$$l_{\dot{e}} = \frac{\mu_{\dot{e}}}{200} \times \sigma \quad (26)$$

$$l_{\ddot{\delta}} = \frac{\mu_{\ddot{\delta}}}{200} \times \gamma \quad (27)$$

After calculating the range of the system, round the number of membership functions using (28), (29), and (30).

$$L_e = \text{int}(l_e) + \rho(l_e) \quad (28)$$

$$L_{\dot{e}} = \text{int}(l_{\dot{e}}) + \rho(l_{\dot{e}}) \quad (29)$$

$$L_{\ddot{\delta}} = \text{int}(l_{\ddot{\delta}}) + \rho(l_{\ddot{\delta}}) \quad (30)$$

The next step is to determine the maximum and minimum value of the number of membership functions using (31) - (36).

$$L_{e_max} = \max\{L_e\} \quad (31)$$

$$L_{e_min} = \min\{L_e\} \quad (32)$$

$$L_{\dot{e}_max} = \max\{L_{\dot{e}}\} \quad (33)$$

$$L_{\dot{e}_min} = \min\{L_{\dot{e}}\} \quad (34)$$

$$L_{\ddot{\delta}_max} = \max\{L_{\ddot{\delta}}\} \quad (35)$$

$$L_{\ddot{\delta}_min} = \min\{L_{\ddot{\delta}}\} \quad (36)$$

After identifying the highest and lowest values, compare these two values in order to find the point with the greatest value using (11). Then, find the point with the greatest value in each dataset using (37) - (39)

$$\hat{L}_e = M_x(L_{e_max}, L_{e_min}) \quad (37)$$

$$\hat{L}_{\dot{e}} = M_x(L_{\dot{e}_max}, L_{\dot{e}_min}) \quad (38)$$

$$\hat{L}_{\ddot{\delta}} = M_x(L_{\ddot{\delta}_max}, L_{\ddot{\delta}_min}) \quad (39)$$

7. Construction of the membership function for the fuzzy logic system

To determine the result of the membership function table, the dataset must be arranged in order of the defined level of the membership function. The membership function can be calculated using (40).

$$\check{L}(x) = \begin{cases} \eta_{a1}, & x = \text{odd number} \\ \eta_{a2}, & x = \text{even number} \end{cases} \quad (40)$$

$$\eta_{a1} = \left[\left(1 - \left(\frac{x+1}{2} \right) \right), \left(2 - \left(\frac{x+1}{2} \right) \right), \dots, \left(x - \left(\frac{x+1}{2} \right) \right) \right] \quad (41)$$

$$\eta_{a2} = \left[\left(1 - \left(\frac{x+2}{2} \right) \right), \left(2 - \left(\frac{x+2}{2} \right) \right), \dots, \left(x+1 - \left(\frac{x+2}{2} \right) \right) \right] \quad (42)$$

After that, substituting the number of membership functions into (40) will result in a dataset of the membership function level of the value of input I in (43) and input II in (44). These values can then be substituted into the estimated equation $f_poly(x,y)$ in (4) to obtain the solution for the membership function of the output of the system in (45).

$$\check{L}_e = \check{L}(\alpha) \quad (43)$$

$$\check{L}_{\dot{e}} = \check{L}(\sigma) \quad (44)$$

$$\check{L}_{\ddot{\delta}} = f_{poly}(\check{L}_e, \check{L}_{\dot{e}}) \quad (45)$$

Form (43), (44), and (45) can be used to generate a membership function table as shown in Table 1.

TABLE III. MEMBERSHIP FUNCTIONS FOR FUZZY LOGIC CONTROLLER DESIGN

		Input II, $f(de), \check{L}_e$				
		$\check{L}_e(j)$...	$\check{L}_e(j+n)$...	$\check{L}_e(\sigma)$
Input I, $f(e), \check{L}_e$	$\check{L}_e(i)$	η_{b1}	...	η_{b4}	...	η_{b7}
	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
	$\check{L}_e(i+m)$	η_{b2}	...	η_{b5}	...	η_{b8}
	\vdots	\vdots	\ddots	\vdots	\ddots	\vdots
	$\check{L}_e(\alpha)$	η_{b3}	...	η_{b6}	...	η_{b9}

The values for η_{b1} to η_{b9} from the Table 3 can be obtained from (46) - (54).

$$\eta_{b1} = \left(\check{L}_\delta(\check{L}_e(i), \check{L}_e(j)) \right) \quad (46)$$

$$\eta_{b2} = \left(\check{L}_\delta(\check{L}_e(i+m), \check{L}_e(j)) \right) \quad (47)$$

$$\eta_{b3} = \left(\check{L}_\delta(\check{L}_e(\alpha), \check{L}_e(j)) \right) \quad (48)$$

$$\eta_{b4} = \left(\check{L}_\delta(\check{L}_e(i), \check{L}_e(j+n)) \right) \quad (49)$$

$$\eta_{b5} = \left(\check{L}_\delta(\check{L}_e(i+m), \check{L}_e(j+n)) \right) \quad (50)$$

$$\eta_{b6} = \left(\check{L}_\delta(\check{L}_e(\alpha), \check{L}_e(j+n)) \right) \quad (51)$$

$$\eta_{b7} = \left(\check{L}_\delta(\check{L}_e(i), \check{L}_e(\sigma)) \right) \quad (52)$$

$$\eta_{b8} = \left(\check{L}_\delta(\check{L}_e(i+m), \check{L}_e(\sigma)) \right) \quad (53)$$

$$\eta_{b9} = \left(\check{L}_\delta(\check{L}_e(\alpha), \check{L}_e(\sigma)) \right) \quad (54)$$

By considering in \check{L}_e has an odd number of members, the position of the row at the center of the table is the position of $\check{L}_e(i+m)$ throughout the row and the value of 0 is greater than 50%. The original position of the table can be improved using this (55).

$$\check{L}_\delta(\check{L}_e(i+m), \check{L}_e(r)) = \eta_{c1} + \eta_{c2} \quad (55)$$

$$\eta_{c1} = \left(\check{L}_\delta(\check{L}_e(i), \check{L}_e(r)) + \dots + \check{L}_\delta(\check{L}_e(i+m-1), \check{L}_e(r)) \right) \quad (56)$$

$$\eta_{c2} = \left(\check{L}_\delta(\check{L}_e(i+m+1), \check{L}_e(r)) + \dots + \check{L}_\delta(\check{L}_e(\alpha), \check{L}_e(r)) \right) \quad (57)$$

where $r = 1, 2, 3, \dots, \sigma$.

By considering in \check{L}_e has an odd number of members, the position of the column at the center of the table is the position of $\check{L}_e(j+n)$ throughout the row and the value of 0 is greater than 50%. The original position of the table can be improved using this (58).

$$\check{L}_\delta(\check{L}_e(q), \check{L}_e(j+n)) = \eta_{d1} + \eta_{d2} \quad (58)$$

$$\eta_{d1} = \left(\check{L}_\delta(\check{L}_e(q), \check{L}_e(j)) + \dots + \check{L}_\delta(\check{L}_e(q), \check{L}_e(j+n-1)) \right) \quad (59)$$

$$\eta_{d2} = \left(\check{L}_\delta(\check{L}_e(q), \check{L}_e(j+n+1)) + \dots + \check{L}_\delta(\check{L}_e(q), \check{L}_e(\sigma)) \right) \quad (60)$$

where $q = 1, 2, 3, \dots, \alpha$.

K_{KFI} is the gain used to determine the input range of each system, which has been estimated in this research and can be calculated using the following (61).

$$K_{KFI} = K_r \frac{M_e \times 100}{M_\delta} \quad (61)$$

This equation represents the gain (K_r) utilized to design the range of the input signal for the system.

VI. SIMULATION RESULTS

This section presents two simulation results. This simulation system uses a robotic manipulator as the simulation model, which has a total of 3 system equations. The details of the simulation system are provided in the following section. The program can be processed and written in Simulink form, as shown in Fig. 6. and Fig. 11.

A. First result

In the first set of results, the signal data or system behavior is stored in the data set of the desired path, actual path, error (e), derivative of error (\dot{e}), and control input (δ) as shown in Fig. 13 and Fig. 14 for Joint R, Fig. 15 and Fig. 16 for Joint T, and Fig. 17 and Fig. 18 for Joint Z. In order to use this data for estimation in finding the membership function, initial variables must be set for the estimation, including the number of membership functions input I (α), which is the total number of rules defined in input 1, the number of membership functions input II (σ), which is the total number of rules defined in input 2, and MB output (γ), which is the total number of rules defined in the output of the fuzzy logic to design the system. The analysis of the data system is not limited to a certain number of data sets, but if the data set is not sufficient, it may be necessary to increase the number of data for analysis. This data can be brought in at specific times for analysis.

The first set of results in this study displays the system behavior data, which includes the desired path, actual path, error (e), derivative of error (\dot{e}), and control input (δ). The data are shown in Fig. 13 and Fig. 14 for Joint R, Fig. 15 and Fig. 16 for Joint T, and Fig. 17 and Fig. 18 for Joint Z. To estimate the membership function, the initial variables such as the number of membership functions input I (α), which is the total number of rules defined in input 1, the number of membership functions input II (σ), which is the total number of rules defined in input 2, and MB output (γ), which is the total number of rules defined in the output of the fuzzy logic system must be set. In this study, the setpoint value was a sine wave with a frequency of 0.1592 Hz and an amplitude of 100 for 30 seconds.

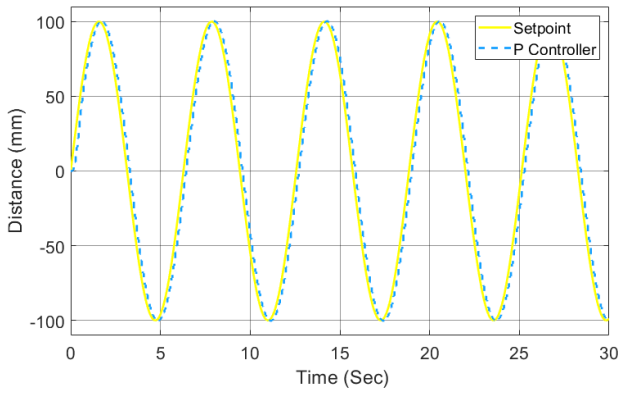


Fig. 13. Desired profile of joint R.

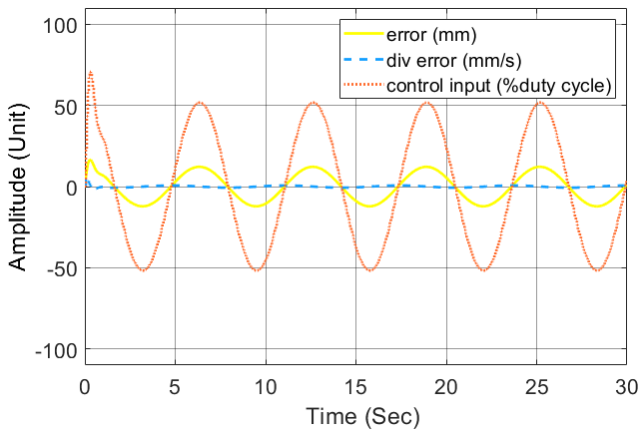


Fig. 14. The collected data from the feedback control of joint R.

The membership function for the robot system was designed by estimating the obtained data using the multiple polynomial regression with bisquare weight method, as shown in Fig. 14. The variable value was set for Joint R in this case. A constant value of $K_r = 3.5$ was used, and the values of $M_e = 12.1770$ and $M_\delta = 51.7524$ were obtained from the Training Algorithm method. These values were substituted into (4) along with the coefficients: $\beta_0 = -0.02787$, $\beta_{10} = 2.97$, $\beta_{01} = -0.1532$, $\beta_{20} = 0.006326$, $\beta_{11} = -0.001186$, $\beta_{02} = 0.006011$, $\beta_{30} = -0.2444$, $\beta_{21} = 0.04956$, $\beta_{12} = -0.173$, and $\beta_{03} = 0.03149$. Table 3 was then used to derive the values, and the resulting membership function table is presented in Table 4.

TABLE IV. MEMBERSHIP FUNCTIONS FOR FUZZY LOGIC CONTROLLER DESIGN OF JOINT R

		Input II, $f(de), \check{L}_e$				
		-2	-1	0	1	2
Input I, $f(e), \check{L}_e$	-2	-2	-4	-4	-4	-2
	-1	-2	-3	-3	-3	-2
	0	-1	1	0	-1	1
	1	2	3	3	2	2
	2	2	4	4	4	3

The membership function for the robot system was designed by estimating the obtained data using the multiple polynomial regression with bisquare weight method, as shown in Fig. 16. The variable value was set for Joint T in this case. A constant value of $K_r = 3$ was used, and the values of $M_e = 12.0960$ and $M_\delta = 96.7700$ were obtained

from the Training Algorithm method. These values were substituted into (4) along with the coefficients: $\beta_0 = -0.1829$, $\beta_{10} = 3.187$, $\beta_{01} = -0.2394$, $\beta_{20} = 0.0409$, $\beta_{11} = 0.00254$, $\beta_{02} = 0.03601$, $\beta_{30} = -0.296$, $\beta_{21} = 0.05267$, $\beta_{12} = -0.2352$, and $\beta_{03} = 0.0571$. Table 3 was then used to derive the values, and the resulting membership function table is presented in Table 5.

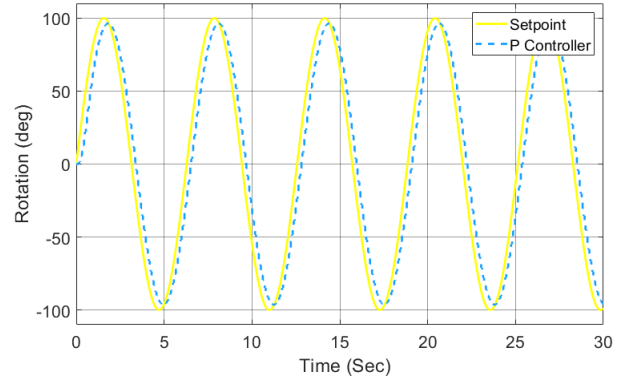


Fig. 15. Desired profile of joint T.

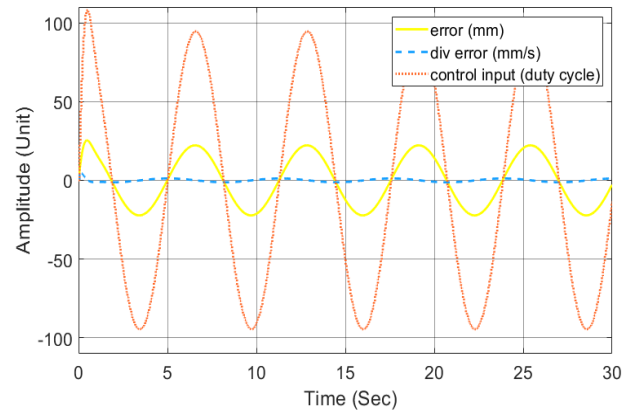


Fig. 16. The collected data from the feedback control of joint T.

TABLE V. MEMBERSHIP FUNCTIONS FOR FUZZY LOGIC CONTROLLER DESIGN OF JOINT T

		Input II, $f(de), \check{L}_e$				
		-2	-1	0	1	2
Input I, $f(e), \check{L}_e$	-2	-2	-4	-4	-3	-2
	-1	-2	-3	-3	-3	-2
	0	0	0	0	0	1
	1	2	3	3	2	2
	2	2	4	4	4	3

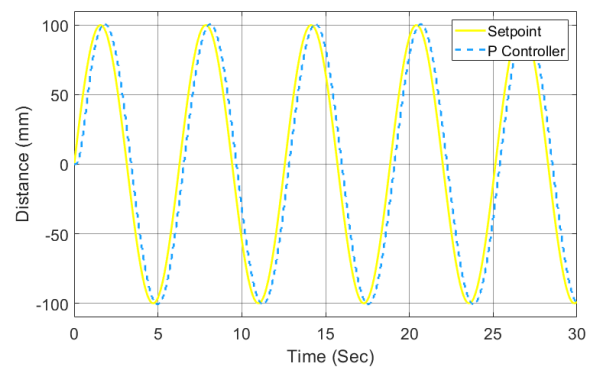


Fig. 17. Desired profile of joint Z.

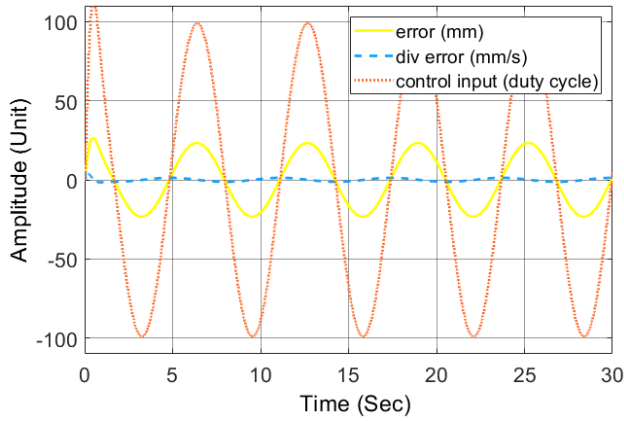


Fig. 18. The collected data from the feedback control of joint Z.

The membership function for the robot system was designed by estimating the obtained data using the multiple polynomial regression with bisquare weight method, as shown in Fig. 18. The variable value was set for Joint Z in this case. A constant value of $K_r = 7$ was used, and the values of $M_e = 14.8473$ and $M_\delta = 99.4767$ were obtained from the Training Algorithm method. These values were substituted into (4) along with the coefficients: $\beta_0 = -0.02341$, $\beta_{10} = 2.908$, $\beta_{01} = -0.1173$, $\beta_{20} = 0.005395$, $\beta_{11} = -0.001758$, $\beta_{02} = 0.004438$, $\beta_{30} = -0.2294$, $\beta_{21} = 0.04342$, $\beta_{12} = -0.1672$, and $\beta_{03} = 0.02175$. Table 3 was then used to derive the values, and the resulting membership function table is presented in Table 6.

TABLE VI. MEMBERSHIP FUNCTIONS FOR FUZZY LOGIC CONTROLLER DESIGN OF JOINT Z

		Input II, $f(de), \tilde{L}_p$				
		-2	-1	0	1	2
Input I, $f(e), \tilde{L}_e$	-2	-2	-4	-4	-3	-2
	-1	-2	-3	-3	-3	-2
	0	0	0	0	0	1
	1	2	3	3	2	2
	2	2	4	4	4	3

B. Second result

The simulation results showed that using a step input profile for the control of Joint R, Joint T, and Joint Z with fuzzy logic, as depicted in Fig. 19, Fig. 20, and Fig. 21, produced a responsive output to the design signal, outperforming the P controller when utilized with a step input because of its ability to reduce overshoot. In conclusion, the multiple polynomial regression with bisquare weighting method proved effective in designing the membership function for the robot system.

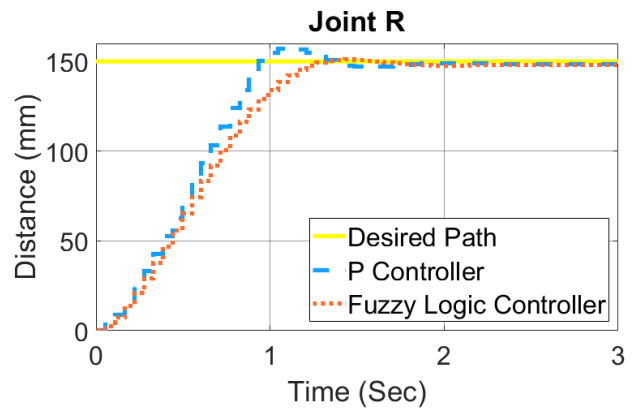


Fig. 19. The results of joint R.

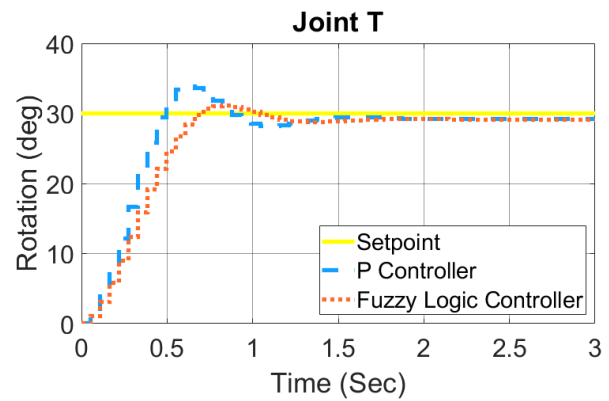


Fig. 20. The results of joint T.

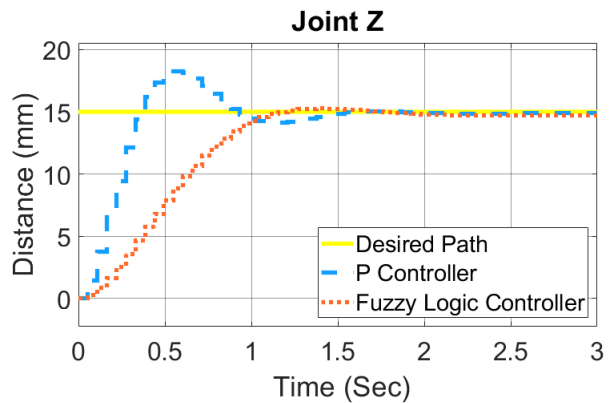


Fig. 21. The results of joint Z.

The simulation results in Fig. 19, Fig. 20, and Fig. 21, along with Table 7, demonstrate the transient response of the system to a step input motion in the R, T, and Z axes. The findings indicate that the fuzzy logic control resulted in a response time of 0.22, 0.22, and 0.66 seconds in the respective axes, while the P control had an overshoot of 4.35, 9.17, and 20.03%. Furthermore, the settling time for the system was better than 0.11, 0.495, and 0.33 seconds for fuzzy logic control, respectively. These results suggest that the membership function evaluation method used in the system is comparable to the control method used to collect data.

The use of fuzzy logic control demonstrated better controllability of the system than P controls, which were the

primary means of collecting learning data. Additionally, the results indicate that fuzzy logic control performs well in both overshoot and settling time, although P control is better in the rise time. However, having an overshoot in a high system may limit the system's usability in real-life applications. If the P control system is to be used, further tuning may be required to reduce the overshoot in the movement of the system.

TABLE VII. TRANSIENT RESPONSE SPECIFICATIONS

Type	Joint R		Joint T		Joint Z	
	P	FLC	P	FLC	P	FLC
Rise time (Sec)	0.94	1.16	0.44	0.66	0.39	1.05
Overshoot (%)	5.04	0.69	12.94	3.77	21.77	1.74
Settling time (Sec)	1.27	1.155	1.38	0.88	1.375	1.05

In summary, the simulation results showed that using the multiple polynomial regression method for optimizing the membership tuning function for fuzzy control with PID-Driven data techniques led to a stable and effective control system with comparable performance to the original closed control system. This approach also eliminates the need to rely solely on the experience of the control system designer in designing the membership functions. However, the multiple polynomial regression technique used to obtain the membership function results in a symmetrical form, which may not be suitable for systems requiring an asymmetrical membership function. The study provides important implications for the development and improvement of fuzzy logic control in robotic manipulators, and the simulation data for Joints R, T, and Z, along with the resulting membership function tables and transient response specifications, are presented in detail in the study.

VII. CONCLUSION

In this study, a fuzzy logic control system was developed for regulating the movement of a robotic arm motor using the Mamdani method and simulated in Simulink. The input and output signals' design rules were defined, and membership functions were established through multiple polynomial regression with the bisquare weighting method. Results indicated a reduction in overshoot ranging from 5-20% for each joint compared to the P control system, demonstrating the potential of fuzzy logic control in precision tasks such as industrial automation or medical robotics. However, the study's scope is limited to robotic arm motor movement, and the conclusions might not be directly applicable to other systems. The membership function evaluation method showed similar performance to the data collection control method, but further exploration and justification are needed. Future research should focus on optimizing membership function tuning, assessing the fuzzy logic control system's robustness and effectiveness under different conditions, and exploring synergies with other artificial intelligence approaches, such as reinforcement learning or neural networks, to advance robotic control systems.

ACKNOWLEDGMENT

The researcher would like to thank the Research Institute, Academic Services Center, and College of Biomedical Engineering, Rangsit University for the grant of research funding to the research team.

REFERENCES

- [1] J. An, F. You, M. Wu, and J. She, "Iterative learning control for nonlinear weighing and feeding process," *Mathematical Problems in Engineering*, pp. 1-9, 2018.
- [2] X. Li, J. Liu, L. Wang, K. Wang, and Y. Li, "Welding process tracking control based on multiple model iterative learning control," *Mathematical Problems in Engineering*, 2019.
- [3] M. Marsono, Y. Yoto, A. Suyetno, and R. Nuralasari, "Design and Programming of 5 Axis Manipulator Robot with GrblGru Open Source Software on Preparing Vocational Students' Robotic Skills," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 539-545, 2021.
- [4] R. Al Tahtawi, M. Agni, and T. D. Hendrawati, "Small-scale robot arm design with pick and place mission based on inverse kinematics," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 469-475, 2021.
- [5] C. Deniz and G. Gökmen, "A new robotic application for COVID-19 specimen collection process," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 73-77, 2022.
- [6] J. Jung and K. Kong, "Mechanical parameter tuning based on iterative learning mechatronics approach," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 906-915, 2018.
- [7] A. Dhyani, M. K. Panda, and B. Jha, "Moth-flame optimization-based fuzzy-PID controller for optimal control of active magnetic bearing system," *Iranian journal of science and technology, transactions of electrical engineering*, vol. 42, no. 4, pp. 451-463, 2018.
- [8] A. Ma'arif and A. Çakan, "Simulation and arduino hardware implementation of dc motor control using sliding mode controller," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 582-587, 2021.
- [9] A. Ma'arif and N. R. Setiawan, "Control of DC motor using integral state feedback and comparison with PID: simulation and Arduino implementation," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 456-461, 2021.
- [10] B. Hekimoğlu, "Optimal Tuning of Fractional Order PID Controller for DC Motor Speed Control via Chaotic Atom Search Optimization Algorithm," *IEEE Access*, vol. 7, pp. 38100-38114, 2019.
- [11] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, and E. T. Helmy, "Motor DC PID System Regulator for Mini Conveyor Drive Based on MATLAB," *Journal of Robotics and Control (JRC)*, vol. 1, no. 6, pp. 185-190, 2020.
- [12] H. Maghfiroh, A. Ramelan, and F. Adriyanto, "Fuzzy-PID in BLDC motor speed control using MATLAB/Simulink," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 8-13, 2022.
- [13] S. Gobinath and M. Madheswaran, "Deep Perceptron Neural Network with Fuzzy PID Controller for Speed Control and Stability Analysis of BLDC Motor," *Soft Computing*, vol. 24, no. 13, pp. 10161-10180, 2020.
- [14] K. Vanchinathan and N. Selvaganesan, "Adaptive Fractional Order PID Controller Tuning for Brushless DC Motor Using Artificial Bee Colony Algorithm," *Results in Control and Optimization*, vol. 4, 2021.
- [15] P. Dutta and S. K. Nayak, "Grey Wolf Optimizer Based PID Controller for Speed Control of BLDC Motor," *Journal of Electrical Engineering & Technology*, vol. 16, no. 2, pp. 955-961, 2021.
- [16] Y. Li, K. H. Ang, and G. C. Chong, "PID control system analysis and design," *IEEE Control Systems Magazine*, vol. 26, no. 1, pp. 32-41, 2006.
- [17] S. K. Mallempati, G. Sathesh, and S. Peddakotla, "Design of optimal PI controller for torque ripple minimization of SVPWM-DTC of BLDC motor," *International Journal of Power Electronics and Drive Systems*, vol. 14, no. 1, pp. 283, 2023.
- [18] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," *International Journal of Dynamics and Control*, vol. 9, no. 2, pp. 818-827, 2021.
- [19] T. Chao, N. Sutarna, J. S. Chiou, and C. J. Wang, "An optimal fuzzy PID controller design based on conventional PID control and nonlinear factors," *Applied Sciences*, vol. 9, no. 6, pp. 1224, 2019.
- [20] Somwanshi, M. Bunde, G. Kumar, and G. Parashar, "Comparison of Fuzzy-PID and PID Controller for Speed Control of DC Motor Using

- LabVIEW," in *Procedia Computer Science*, vol. 152, pp. 252-260, 2019.
- [21] S. J. Hammoodi, K. S. Flayyih, and A. R. Hamad, "Design and Implementation of Speed Control System for DC Motor Based on PID Control and Matlab Simulink," *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 1, pp. 127, 2020.
- [22] V. V. Patel, "Ziegler-Nichols Tuning Method," *Resonance*, vol. 25, no. 10, pp. 1385-1397, 2020.
- [23] R. R. Alla, N. Lekyasri, and K. Rajani, "PID Control Design for Second Order Systems," *Int. J. Eng. Manuf.*, vol. 9, no. 4, pp. 45-56, 2019.
- [24] E. Bashier and O. Mohammed, "Optimally Tuned Proportional Integral Derivatives (PID) Controllers for Set-Point," *Journal of Engineering and Computer Science (JECS)*, vol. 13, no. 1, pp. 48-53, 2019.
- [25] T. Y. Wu, Y. Z. Jiang, Y. Z. Su, and W. C. Yeh, "Using Simplified Swarm Optimization on Multiloop Fuzzy PID Controller Tuning Design for Flow and Temperature Control System," *Applied Sciences*, vol. 10, no. 23, pp. 8472, 2020.
- [26] V. Dubey, "Comparative Analysis of PID Tuning Techniques for Blood Glucose Level of Diabetic Patient," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 11, pp. 2948-2953, 2021.
- [27] A. O. Amole, O. E. Olabode, D. O. Akinyele, and S. G. Akinjobi, "Optimal Temperature Control Scheme for Milk Pasteurization Process Using Different Tuning Techniques for a Proportional Integral Derivative Controller," *Iranian Journal of Electrical and Electronic Engineering*, vol. 2170, pp. 2170-2170, 2022.
- [28] D. Zhang, B. Du, P. Zhang, and S. Chen, "Constant force PID control for robotic manipulator based on fuzzy neural network algorithm," *Complexity*, pp. 1-11, 2020.
- [29] B. Li, Y. Tan, J. Chen, X. Liu, and S. Yang, "Precise active seeding downforce control system based on fuzzy PID," *Mathematical Problems in Engineering*, 2020.
- [30] R. Kristiyono and W. Wiyono, "Autotuning Fuzzy PID Controller for Speed Control of BLDC Motor," *Journal of Robotics and Control*, vol. 2, no. 5, pp. 400-407, 2021.
- [31] K. Lee, D. Y. Im, B. Kwak, and Y. J. Ryoo, "Design of fuzzy-PID controller for path tracking of mobile robot with differential drive," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 3, pp. 220-228, 2018.
- [32] M. J. Mohamed and M. Y. Abbas, "Design of a Fuzzy PID Controller for Trajectory Tracking of a Mobile Robot," *Engineering and Technology Journal*, vol. 36, no. 1, pp. 100-110, 2018.
- [33] K. Eltag, M. S. Aslamx, and R. Ullah, "Dynamic Stability enhancement using fuzzy PID control technology for power system," *International Journal of Control, Automation and Systems*, vol. 17, pp. 234-242, 2019.
- [34] M. Rabah, A. Rohan, Y. J. Han, and S. H. Kim, "Design of fuzzy-PID controller for quadcopter trajectory-tracking," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 3, pp. 204-213, 2018.
- [35] X. Long, Z. He, and Z. Wang, "Online optimal control of robotic systems with single critic NN-based reinforcement learning," *Complexity*, pp. 1-7, 2021.
- [36] N. Razmjoooy and M. Ramezani, "Optimal control of two-wheeled self-balancing robot with interval uncertainties using Chebyshev inclusion method," *Majlesi Journal of Electrical Engineering*, vol. 12, no. 1, pp. 13-21, 2018.
- [37] L. Dembia, N. A. Bianco, A. Falisse, J. L. Hicks, and S. L. Delp, "Opensim moco: Musculoskeletal optimal control," *PLOS Computational Biology*, vol. 16, no. 12, p. e1008493, 2020.
- [38] Sánchez-Sánchez and D. Izzo, "Real-time optimal control via deep neural networks: study on landing problems," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1122-1135, 2018.
- [39] Zhao, D. Liu, and C. Luo, "Reinforcement learning-based optimal stabilization for unknown nonlinear systems subject to inputs with uncertain constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 10, pp. 4330-4340, 2019.
- [40] Y. Pan, C. A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Imitation learning for agile autonomous driving," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 286-302, 2020.
- [41] Ejaz, M.T. Hamayun, S. Hussain, S. Ijaz, S. Yang, N. Shehzad, and A. Rashid, "An adaptive sliding mode actuator fault tolerant control scheme for octorotor system," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019.
- [42] J.Y. Zhai and Z.B. Song, "Adaptive sliding mode trajectory tracking control for wheeled mobile robots," *International Journal of Control*, vol. 92, no. 10, pp. 2255-2262, 2019.
- [43] Jiang, H.R. Karimi, S. Yang, C. Gao, and Y. Kao, "Observer-based adaptive sliding mode control for nonlinear stochastic Markov jump systems via T-S fuzzy modeling: Applications to robot arm model," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 466-477, 2020.
- [44] L. Gracia, J.E. Solanes, P. Muñoz-Benavent, J.V. Miro, C. Perez-Vidal, and J. Tornero, "Adaptive sliding mode control for robotic surface treatment using force feedback," *Mechatronics*, vol. 52, pp. 102-118, 2018.
- [45] K. Liu, H. Gao, H. Ji, and Z. Hao, "Adaptive sliding mode based disturbance attenuation tracking control for wheeled mobile robots," *International Journal of Control, Automation and Systems*, vol. 18, no. 5, pp. 1288-1298, 2020.
- [46] T. Zhang, Y. Yu, and Y. Zou, "An adaptive sliding-mode iterative constant-force control method for robotic belt grinding based on a one-dimensional force sensor," *Sensors*, vol. 19, no. 7, p. 1635, 2019.
- [47] J. Yang, J. Na, G. Gao, and C. Zhang, "Adaptive neural tracking control of robotic manipulators with guaranteed nn weight convergence," *Complexity*, 2018.
- [48] Q. Zhou, S. Zhao, H. Li, R. Lu, and C. Wu, "Adaptive neural network tracking control for robotic manipulators with dead zone," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3611-3620, 2018.
- [49] Z. Chen, Y. Liu, W. He, H. Qiao, and H. Ji, "Adaptive-neural-network-based trajectory tracking control for a nonholonomic wheeled mobile robot with velocity constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5057-5067, 2020.
- [50] M. Rahmani and M. H. Rahman, "Adaptive neural network fast fractional sliding mode control of a 7-DOF exoskeleton robot," *International Journal of Control, Automation and Systems*, vol. 18, no. 1, pp. 124-133, 2020.
- [51] B. Rahmani and M. Belkheiri, "Adaptive neural network output feedback control for flexible multi-link robotic manipulators," *International Journal of Control*, vol. 92, no. 10, pp. 2324-2338, 2019.
- [52] F. Luan, J. Na, Y. Huang, and G. Gao, "Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence," *Neurocomputing*, vol. 337, pp. 153-164, 2019.
- [53] J. Yang and M. Tan, "Sliding mode control for flexible-link manipulators based on adaptive neural networks," *International Journal of Automation and Computing*, vol. 15, no. 2, pp. 239-248, Mar. 2018.
- [54] M. Rahmani, A. Ghanbari, and M. M. Etefagh, "A novel adaptive neural network integral sliding-mode control of a biped robot using bat algorithm," *Journal of Vibration and Control*, vol. 24, no. 10, pp. 2045-2060, 2018.
- [55] Z. Ma, J. Feng, and J. Yang, "Research on vertical air-water trans-media control of hybrid unmanned aerial underwater vehicles based on adaptive sliding mode dynamical surface control," *International Journal of Advanced Robotic Systems*, vol. 15, no. 2, 2018.
- [56] W. He, Z. Yin and C. Sun, "Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier Lyapunov function," in *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1641-1651, 2017.
- [57] J. Wang, P. Zhu, B. He, G. Deng, C. Zhang, and X. Huang, "An adaptive neural sliding mode control with ESO for uncertain nonlinear systems," *Int. J. Control, Autom., and Syst.*, vol. 19, pp. 687-697, 2021.
- [58] X. Liu, C. Yang, Z. Chen, M. Wang, and C. Y. Su, "Neuro-adaptive observer based control of flexible joint robot," *Neurocomputing*, vol. 275, pp. 73-82, 2018.

- [59] J. A. Spanias, A. M. Simon, S. B. Finucane, E. J. Perreault, and L. J. Hargrove, "Online adaptive neural control of a robotic lower limb prosthesis," *J. Neural Eng.*, vol. 15, no. 1, p. 016015, 2018.
- [60] C. Sun, G. Li, and J. Xu, "Adaptive neural network terminal sliding mode control for uncertain spatial robot," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 6, p. 1729881419894065, 2019.
- [61] J. Jiang, S. Guo, L. Zhang, and Q. Sun, "Motor Ability Evaluation of the Upper Extremity with Point-To-Point Training Movement Based on End-Effector Robot-Assisted Training System," *J. Healthcare Eng.*, 2022.
- [62] E. Luis and A. P. Schoellig, "Trajectory generation for multiagent point-to-point transitions via distributed model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 375-382, 2019.
- [63] M. M. G. Ardakani, B. Olofsson, A. Robertsson, and R. Johansson, "Model predictive control for real-time point-to-point trajectory generation," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 972-983, 2018.
- [64] S. Xue, Z. Zhao, and F. Liu, "Latent variable point-to-point iterative learning model predictive control via reference trajectory updating," *Eur. J. Control*, vol. 65, 2022.
- [65] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9480-9487, 2020.
- [66] Y. Wang, K. Zhu, B. Chen, and H. Wu, "A new model-free trajectory tracking control for robot manipulators," *Mathematical Problems in Engineering*, vol. 2018, pp. 1-13, 2018.
- [67] X. Ma, Y. Zhao, and Y. Di, "Trajectory tracking control of robot manipulators based on U-model," *Mathematical Problems in Engineering*, vol. 2020, pp. 1-10, 2020.
- [68] Y. Wang, F. Gao, and F. J. Doyle III, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589-1600, 2009.
- [69] Q. Ai, D. Ke, J. Zuo, W. Meng, Q. Liu, Z. Zhang, and S. Q. Xie, "High-order model-free adaptive iterative learning control of pneumatic artificial muscle with enhanced convergence," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9548-9559, Nov. 2019.
- [70] B. P. Huynh, C. W. Wu, and Y. L. Kuo, "Force/position hybrid control for a hexa robot using gradient descent iterative learning control algorithm," *IEEE Access*, vol. 7, pp. 72329-72342, 2019.
- [71] X. Jin, "Iterative learning control for output-constrained nonlinear systems with input quantization and actuator faults," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 2, pp. 729-741, Jan. 2018.
- [72] X. Xing and J. Liu, "Modeling and robust adaptive iterative learning control of a vehicle-based flexible manipulator with uncertainties," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 8, pp. 2385-2405, 2019.
- [73] X. Jin, "Nonrepetitive leader-follower formation tracking for multiagent systems with LOS range and angle constraints using iterative learning control," *IEEE Transactions on Cybernetics*, vol. 49, no. 5, pp. 1748-1758, 2018.
- [74] P. Chotikunnan and B. Panomruttanarug, "Practical design of a time-varying iterative learning control law using fuzzy logic," *Journal of Intelligent & Fuzzy Systems*, vol. Preprint, pp. 1-16, 2022.
- [75] P. Chotikunnan, B. Panomruttanarug, and P. Manoonpong, "Dual Design Iterative Learning Controller for Robotic Manipulator Application," *Control Engineering and Applied Informatics*, vol. 24, no. 3, pp. 76-85, 2022.
- [76] P. Chotikunnan, and B. Panomruttanarug, "The application of fuzzy logic control to balance a wheelchair," *Journal of Control Engineering and Applied Informatics*, vol. 18, no. 3, pp. 41-51, 2016.
- [77] Q. T. Ngo and M. H. Nguyen, "Enhancement of the Tracking Performance for Robot Manipulator by Using the Feed-forward Scheme and Reasonable Switching Mechanism," *Journal of Robotics and Control (JRC)*, vol. 3, no. 3, pp. 328-337, 2022.
- [78] R. Kristiyono and W. Wiyono, "Autotuning fuzzy PID controller for speed control of BLDC motor," *Journal of Robotics and Control (JRC)*, vol. 2, no. 5, pp. 400-407, 2021.
- [79] Z. Lin, C. Cui and G. Wu, "Dynamic modeling and torque feedforward based optimal fuzzy PD control of a high-speed parallel manipulator," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 527-538, 2021.
- [80] Suwarno, Y. Finayani, R. Rahim, J. Alhamid and A. R. Al-Obaidi, "Controllability and Observability Analysis of DC Motor System and a Design of FLC-Based Speed Control Algorithm," *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 227-235, 2022.
- [81] W. Robson, I. Ernawati and C. Nugrahaeni, "Design of multisensor automatic fan control system using Sugeno fuzzy method," *Journal of Robotics and Control (JRC)*, vol. 2, no. 4, pp. 302-306, 2021.
- [82] A. H. Ginting, S. Y. Doo, D. E. Pollo, H. J. Djahi and E. R. Mauboy, "Attitude Control of a Quadrotor with Fuzzy Logic Controller on SO (3)," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 101-106, 2022.
- [83] I. Iswanto and I. Ahmad, "Second Order Integral Fuzzy Logic Control Based Rocket Tracking Control," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 594-604, 2021.
- [84] S. R. Utama, A. Firdausi and G. P. Hakim, "Control and Monitoring Automatic Floodgate based on NodeMCU and IOT with Fuzzy Logic Testing," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 14-17, 2022.
- [85] R. F. Arif, A. Firdausi and G. P. Hakim, "Nebulizer operational time control based on drug volume and droplet size using fuzzy Sugeno method," *Journal of Robotics and Control (JRC)*, vol. 2, no. 2, pp. 94-97, 2021.
- [86] H. Salem, M. Y. Shams, O. M. Elzeki, M. Abd Elfattah, J. F. Al-Amri, and S. Elnazer, "Fine-tuning fuzzy KNN classifier based on uncertainty membership for the medical diagnosis of diabetes," *Applied Sciences*, vol. 12, no. 3, p. 950, 2022.
- [87] M. Fayaz, I. Ullah, and D. Kim, "An optimized fuzzy logic control model based on a strategy for the learning of membership functions in an indoor environment," *Electronics*, vol. 8, no. 2, p. 132, 2019.
- [88] S. H. Khairuddin, M. H. Hasan, E. A. P. Akhir, and M. A. Hashmani, "Generating type 2 trapezoidal fuzzy membership function using genetic tuning," *Computers, Materials and Continua*, vol. 71, no. 1, pp. 717-734, 2022.
- [89] P. Chotikunnan and R. Chotikunnan, "Dual Design PID Controller for Robotic Manipulator Application," *Journal of Robotics and Control (JRC)*, vol. 4, no. 1, pp. 23-34, 2023.
- [90] N. R. Setiawan, A. Ma'arif, and N. S. Widodo, "DC Motor Controller Using Full State Feedback," *Control Systems and Optimization Letters*, vol. 1, no. 1, pp. 7-11, 2023.
- [91] Y. Zahraoui, M. Akherraz, and A. Ma'arif, "A comparative study of nonlinear control schemes for induction motor operation improvement," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 1-17, 2022.
- [92] S. Febriyan and R. D. Puriyanto, "Implementation of DC Motor PID Control On Conveyor for Separating Potato Seeds by Weight," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 15-26, 2021.
- [93] H. Maghfiroh, A. J. Titus, A. Sujono, F. Adriyanto, and J. S. Saputro, "Induction Motor Torque Measurement using Prony Brake System and Close-loop Speed Control," *International Journal of Robotics and Control Systems*, vol. 2, no. 3, pp. 594-605, 2022.
- [94] Y. I. Nadjai, H. Ahmed, N. Takorabet, and P. Haghgooei, "Maximum Torque per Ampere Control of Permanent Magnet Assisted Synchronous Reluctance Motor: An Experimental Study," *International Journal of Robotics and Control Systems*, vol. 1, no. 4, pp. 416-427, 2021.
- [95] R. Rikwan and A. Ma'arif, "DC Motor Rotary Speed Control with Arduino UNO Based PID Control," *Control Systems and Optimization Letters*, vol. 1, no. 1, pp. 17-31, 2023.
- [96] D. Prastiyo and W. S. Aji, "Irrigation Sluice Control System Using Algorithm Based DC Motor PID And Omron PLC," *Control Systems and Optimization Letters*, vol. 1, no. 1, pp. 19-26, 2023.
- [97] A. Cahya and R. D. Puriyanto, "The Design of Rice Milling and Screening Systems Uses the DC Motor PID Method," *Control Systems and Optimization Letters*, vol. 1, no. 1, pp. 1-6, 2023.

- [98] E. S. Rahayu, A. Ma'arif, and A. Çakan, "Particle swarm optimization (PSO) tuning of PID control on DC motor," *International Journal of Robotics and Control Systems*, vol. 2, no. 2, pp. 435-447, 2022.
- [99] Bounemour and M. Chemachema, "Adaptive fuzzy fault-tolerant control for a class of nonlinear systems under actuator faults: application to an inverted pendulum," *International Journal of Robotics and Control Systems*, vol. 1, no. 2, pp. 102-115, 2021.
- [100] H. Abdelfattah, S. A. Kotb, M. Esmail, and M. I. Mosaad, "Adaptive Neuro-Fuzzy Self Tuned-PID Controller for Stabilization of Core Power in a Pressurized Water Reactor," *Int. J. Robotics and Control Systems*, vol. 3, no. 1, pp. 1-18, 2023.
- [101] M. Elouni, H. Hamdi, B. Rabaoui, and N. B. Braiek, "Adaptive PID fault-tolerant tracking controller for Takagi-Sugeno fuzzy systems with actuator faults: application to single-link flexible joint robot," *Int. J. Robotics and Control Systems*, vol. 2, no. 3, pp. 523-546, 2022.