

Performance Analysis and Improvement of Robot Arm 5DOF Using PID and Fuzzy Controllers: A Comparative Study

Mohamed M.Gouda¹, Fatma El-Zahraa Said^{2*}, El Sayed Elmoushi³, and Sabreen A. Abdelwahab⁴

^{1,2}Electronics Department, Faculty of Technology and Education, Helwan University, Cairo, Egypt, email: mohamedgouda@techedu.helwan.edu.eg, fatmaawad@techedu.helwan.edu.eg,

^{3,4} Production Technology Department, Faculty of Technology and Education, Helwan University, Cairo, Egypt, email: e_elmoushi@yahoo.com, engsabreenabdallah@gmail.com

*Corresponding author, Email address: fatmaawad@techedu.helwan.edu.eg, DOI: [10.21608/PSERJ.2022.132466.1177](https://doi.org/10.21608/PSERJ.2022.132466.1177)

ABSTRACT

Robot modeling is essential before using control systems to verify that the desired task is completed with the lowest feasible error rate according to the inputs. When modeling a robot, the derivation of the forward movement of the robot axes is a fundamental step based on the method of Denavit–Hartenberg. This research aims to control the robot arm motion using two controllers. PID controller was used as a reference to compare the results with those of an FLC. First, an FLC was used to enhance the nonlinearity of the robot arm. FLC was designed based on Mamadani pro-Max inference. Four different defuzzification methods were used and compared to obtain the control signal. These are BOA, MOM, SOM, and COG. The results of this controller were then compared to the PID results on a transient response scale. Finally, simulation was done using MATLAB Simulink software. Based on the simulation results, BOA, MOM, and SOM techniques yield almost identical results; however, the COG strategy yields a broad range of outcomes. Implementing a simple defuzzification approach resulted in system optimization due to the complexity of processes like fuzzification and defuzzification. Based on the simulation results, the FLC system yields better results than those obtained with a PID. The FLC has lower rise time, settling time, steady-state error, and less overshoot than the PID controller.

Keywords: Fuzzy logic controller, PID controller, Robotic arm, Forward kinematics, Open loop dynamics.

Received 9-4-2022

Revised 20-4-2022

Accepted 6-5-2022

© 2022 by Author(s) and PSERJ.

This is an open access article licensed under the terms of the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



1. INTRODUCTION

The robot arm consists of a series of connected links, that can move in a transitional or rotational movement (such as an articulated robot). These links are connected with joints forming a kinematic chain; From the base to the end effector, all joints are actuated and stretched. With the final terminal effect, the robot arm is identical to a human arm. The robot arm resembles and performs the same functions as the human hand.

The main parts in a robot arm are the base, joints, links, and a gripper. The base is the basic part of the arm; it may be fixed or active.

Then separate links are attached to it. The link is to fix and support the clutch, and the clutch is used to hold and move objects [1].

As a result, before utilizing robot manipulators to operate with high accuracy, modeling and analyzing the robot manipulators and control techniques is critical. Robot kinematics may be forward or inverse kinematics. The purpose of the control assignment is to move the robot arm from one position to another. Therefore, the intended position or angle of each joint must be known in advance, and forward kinematics is used with the angle as input and inverse kinematics with the required end position as input. Computing the necessary joint angles for the end-effector coordinates is more complicated than forward kinematics [2].

1.1. Literature Review

As an advanced investigation into this emerging field, many studies have been done on manipulator robots. Some literature has explored the kinematic analysis of industrial and educational robot arms such as the PUMA 560, SCARA, and SG5-UT processors. [3], [4], and [5]. Other papers in control technology included Proportional Integral Derivative (PID) and Fuzzy Logic Controller (FLC). [6], [7], [8], and [9]. A brief review of some literature is as follows:

In a study done by Delibes [10], FLC and PID control algorithms were used to regulate the position of a DC motor. Both controllers (PID and FLC) were created using the LABVIEW application. After applying the controllers, the target position was obtained with a 0.4 percent overrun and an 80 m sec settling time for FLC, but with a 4 percent overshoot and a 120 msec settling time for PID. The impact of FLC on the procedure of a robot movement simulation controlled by a digital controller was highlighted by Delibes. The authors in [11] presented the fuzzy supervisory approach for adjusting PID parameters. The performance of parameters provided by Z-N is improved using this method. The FLC is shown to be superior in the simulation.

Dzulhizzam Bin Dulaidi [12] controlled the robot arms using FLC to obtain the desired position. The FLC performance was then compared with the PID. Based on their study, FLC has proven to be more efficient in PID response behavior.

V. K. Bang et al. [13] used the FLC and Genetic Algorithms to map a path. FLC is used to achieve the optimal control of an automatic arm, and the benefit is that FLC is an efficient and realistic method to achieve better mechanical arm movement.

Yung Tao et al. [14] used the PID approach for each sampling to ensure the consistency of the final transponder route. Testing shows more reliable and effective fuzzy PID performance than the PID power. The Fuzzy PID method is also used because it is effective and precise in the preparation of route planning.

Jaffa et al. [15] used Neural-Fuzzy Controller (NFC) to control the automatic arm position. The neural network was used to control input and output information and hybrid learning. The network was trained using an algorithm. The simulation results show that the NFC controls the automatic arm path better than the PID.

Usman Kabir et al. [16] used a 3-DOF robot manipulator, and three distinct position control approaches were created and assessed. Each link of a robot manipulator was fitted with PID, PD, and FLC controllers, and performance comparisons were made using transient and steady-state characteristics. All three controllers followed the setpoint with slight steady-state error, according to the results. In addition, the PID and PD controllers performed better with respect to rising

and settling times, while the FLC controller had less overshoot.

Alassar et al. [17] used a PID and a fuzzy smart controller to control the 5-DOF robot arm. The findings revealed that FLC-based PID parameter tuning is superior to traditional approaches.

Jafar Tavooosi et al. [18] designed two independent Neural-Fuzzy controllers for the trajectory tracking of a robot arm. According to the simulation findings, the Neural-Fuzzy controller outperformed the PID controller in terms of trajectory control.

Jamal Abd Altayef et al. [19] used FLC and PID control algorithms to display the direction control of the DC motor. The architecture of the two controllers (PID-FUZZY) was based on a LabVIEW system. The results revealed that the FLC is superior to maintaining the target location than the PID controller.

Yong-Lin Kuo et al. [20] used the PID, Fuzzy, and Fuzzy PID systems to control the robotic arm. The fuzzy-PID controller resulted in a lower steady-state error.

Nairi Dersarkissian et al. [21] presented a controller design to control the position of a wheelchair-mounted robotic arm using PID and FLC. The FLC outperforms traditional tuning approaches, according to the findings. To obtain greater control in the future, the authors integrated a PID controller with FLC to achieve better control.

Z.F. Baghli et al. [22] controlled a two-degree arm manipulator robot utilizing two different control approaches, single-output control based on the classical PID model and smart adaptive FPID. They found that Fuzzy PID was more stable in situ and produced better results than conventional PID controllers.

Athar Ali et al. [23] controlled an upper rehabilitation robot using PID and FPID controllers. Their findings revealed that the FPID controller outperforms the traditional PID controller. Furthermore, the FPID controller proved robust for the upper limb rehabilitation robotic system.

Masoud Solouki et al. [24] introduced a robot arm manipulator controlled with 5-DOF using two controllers, PID and Fuzzy. Although the PID controller is a standard controller for linear systems against nonlinear systems, the results showed that using fuzzy rules leads to better results than traditional methods.

Fucheng Cao et al. [25] used assistive robots to help people with physical limitations enhance their independence and quality of life by interacting with them. The PID and fuzzy systems were used to regulate it. The findings revealed that using the FLC outperforms the PID controller.

Angel and Viola [26] proposed a delta manipulator tracking control based on a fractional-order proportional integral derivative (PID) controller and a calculated torque control strategy. Then, they established a collaborative simulation model of a delta robot for investigating, designing, and verifying a control strategy.

The durability of the controller to external disturbances was evaluated using performance indexes such as joint and space errors, joint torque, and trajectory tracking. The results showed that the fractional-order proportional integral derivative (PID) and calculated torque control strategy were resilient and automatically disturbance-rejected when used to a parallel robot's tracking task. It is usually necessary to coordinate actions when performing implementation tasks in the human environment. The control of robot arm movement also requires smart coordination skills

According to biomechanics, Pedrammehr et al. [27] studied the description of dual-arm motion tasks based on extended mutual task space impersonation and estimated the performance of specified tasks. The experimental results showed that the task specification based on ECTS was intuitive and effective.

Jinjun et al. [28] suggested a new symmetric, adaptive, variable-admittance control method for position continuation tracking of a dual-arm cooperative robot. For the first time, the endorsement parameters were modified online to track the desired position and force based on the tracking error to compensate for the anonymous trajectory deviation. As a result, the symmetrical adaptive variable-admittance control with two arms coordination was realized. Furthermore, the simulation experiments showed that the method could achieve good position and force tracking implementation.

Nasr and Ayman [29] controlled a system to reach a desired joint angle position through simulation of PID controllers using MATLAB/Simulink. The results showed that a slight change in initial joint angle positions of the robot arm resulted in different desired joint angle positions. Therefore, it was necessary to adjust and turn the gains of the PID controllers every moment to prevent overshoot and oscillation due to changes in parameters values.

Nairi Dersarkissian [30] This study ranks the different methods used To design a controller to control the position Wheelchair-mounted robotic arm using advanced fog inference system. This study is also used to identify and Demonstrate the best solution for designing a control method. The Simulation results for the five membership states and the seven membership states show that both controllers are working correctly and has an overflow, something considered within the acceptable range. However, the steady state error in the seven members The function controller is smaller compared to the 5 MFS. GPS design using DC motor Fuzzy logic shows that the use of FLC in position control Application, a shorter settling time can be achieved by adjusting Control rules, membership functions and the world output variable letter.

Usman Kabir et al. [31] In this study, three different position control methodologies have been designed and analysed on a 3-DOF robot manipulator. The PID, PD and FLC controllers were applied to each link of the

robot manipulator and performance comparisons were made using transient and steady state characteristics. The results showed that all three controllers were able to track the setpoint with negligible steady state error. The PID and PD controllers gave better performance in terms of the rise time and settling time while the FLC resulted in decreased overshoot.

Mohamed Fawzy et al [32] A comparison study was implemented to control the arm robot angles with four degrees of freedom. The control of the arm robot was extensively investigated using two controllers under different operating conditions. These controllers are the fuzzy logic controller (FLC) and the 2-DOF PID controller. The system performance with two controllers was simulated using MATLAB/Simulink. Firstly, the desired model is built and for each joint of the arm robot a fuzzy logic controller and a 2-DOF PID controller is designed. Then in order to reducing the overshoot and increasing the speed of response, the reset mechanism is applied. The results showed that, the FLC has optimal performance (fast response with better rise time and no overshoot) in controlling of the arm robot compared with the 2-DOF PID controller.

Abdel-Azim S et al [33] controllers (FLC) has been used because it is efficient tools for control of nonlinear and uncertain parameters systems. This paper aims to design a fuzzy logic controller for position control of a PUMA 560 robot manipulator. Based on simulation results we conclude that the performance of the fuzzy

logic controller in term of position tracking error in case of disturbance or load is better than the conventional computed torque (PD-CTC and PID-CTC) controllers.

1.1. Problem Statement

Modeling, kinematic analysis, and control of robot arms are essential topics in the literature. This may be accomplished using various controllers. A controller reduces the difference between the required and the actual situation robot joints or end effectors. A controller must satisfy specific requirements to do so. For example, overshoot is reduced, the rising time is minimized, the steady-state error is eliminated, and the restlessness load on each joint motor is reduced.

The increasing complexity of robotic tasks requires controllers that are intelligent, powerful, easy to calculate, and easy to install and analyze in order to optimize and maximize the performance of the industrial robot arms. [34]. In the early decades, the integral proportional derivative (PID) was the most extensively used controller in commercial applications and industry. The PID controllers are simple to develop and install, have great flexibility and dependability, and are inexpensive, so the first attempt is to implement the PIDs. Due to the nonlinearity of robotic arm manipulators, PID controllers struggle to cope with the time response, skips, and steady-state errors of the robotic arm processor. Hence, they are insufficient to

meet the tracking control effectiveness requirements. These issues can be mitigated using nonlinear controllers. A nonlinear controller such as FLC is needed to address the nonlinearity problem. The conduct of both controllers must be evaluated. To effectively investigate the assignment required, the forward motion equations and dynamics of the robot arm must first be investigated. Then, the controllers must be created, implemented, and assessed in terms of the performance index in a mechanical manipulator system.

Noisy sensors, imprecise global models, and uncertainty in action execution can significantly impact the implementation of motion control in real-world robotic applications. The mathematical and functioning models of an industrial robot involve numerous equations. In addition, building and experimenting with a practical model takes a long time. The utility of industrial models designed by computer simulation software reduces the costs and time spent in designing and simulating industrial robots [35]. The most straightforward technique to represent a real robot without writing code or programming is to use computer software simulation, which solves mathematic equations. As a result, the software packages Matlab and Simulink will be used to simulate the dynamics and control of the robot arm. Finally, the robot arm will be built and tested as an actual prototype.

The research steps were organized according to the flowchart in Figure 1.

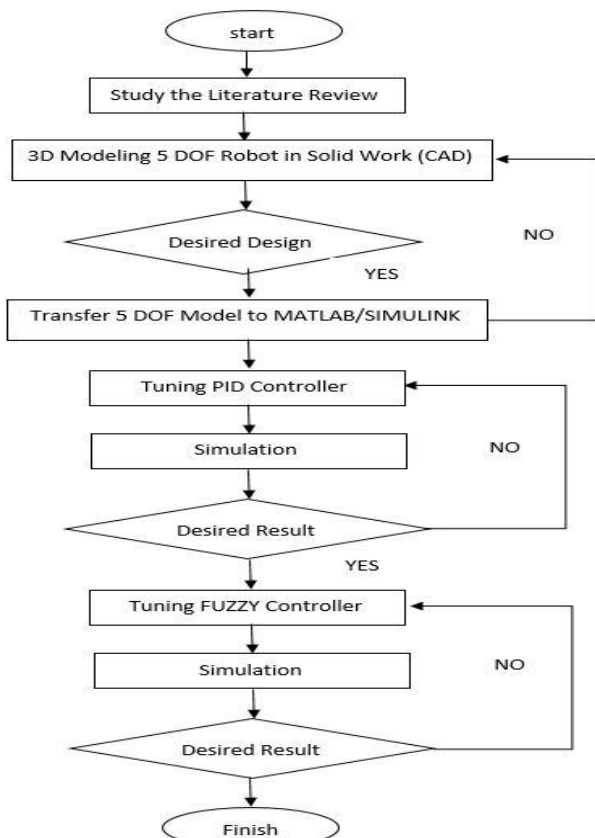


Figure 1: Research Methodology

This paper has nine sections, which are as follows: Section 1 is the introduction, and Section 2 presents the proposed mechanical design of the robot arm manipulator and working principle. Section 3 gives the design of the mechanism in SolidWorks. Section 4 describes the forward kinematic model of the proposed robot arm. Section 5 presents the development of the simulation model of the arm and the simulation of dynamics. Section 6 presents the development and control circuit of the robot arm prototype. Finally, Section 7 discussion, Section 8 presents Conclusions followed by Section 9 references.

2. PROPOSED MECHANICAL DESIGN AND WORKING PRINCIPLE OF ROBOTIC ARM

The robot arm’s mechanical design is based on a robotic manipulator that performs functions comparable to a human arm. [36, 37, 38].

Each joint gives the robot certain mobility freedom (DOF). The robotic arm consists of several links joined by hinges to facilitate movement. In this robotic arm, there are two species of joints. The first is a rotational joint, also known as a human joint, which only allows relative rotation between the two connections. In robotics, this is the most frequent form of joint. The second type is a sliding or prismatic joint. This joint allows only linear proportional movement between two long links along an axis.

The manipulator is considered part of a kinetic chain [39]. The robot’s base is attached to one end of the chain, while the other end is coupled to a tool such as a hand, gripper, or end effector.

Footnotes

3. MECHANISM DESIGN IN SOLIDWORKS

The SolidWorks program was chosen to create the robotic arm because it allowed simultaneous design and visualization. It also evaluates collisions and interferences on the arm. Since each link depends on the one before it, the robotic arm’s design needs to begin at the base and terminate at the gripper. Therefore, the base (Link 0) is the first to be designed, followed by Links 1, 2, 3, etc. The material proposed was wood for the robotic arm with a thickness of 5 mm, assuming a 150 mg load that can be carried and moved by the arm robot. The robot arm has five rotating joints and a movable grip. A robotic arm model developed in SolidWorks is shown in Figure 2.

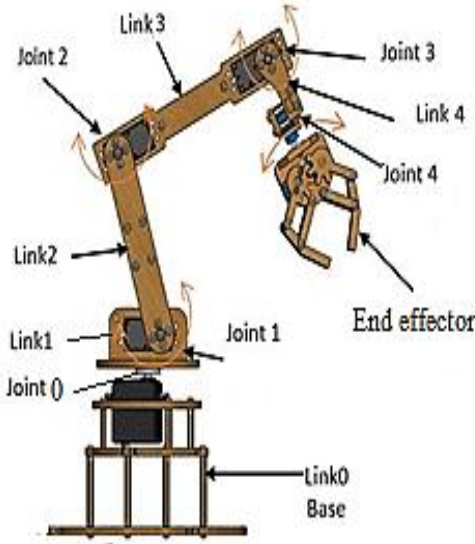


Figure 2: Robot Arm Model in SolidWorks.

4. ROBOTIC ARM FORWARD KINEMATICS MODEL

The forward movement shows the transition from one frame to the next, starting at the base and ending at the handle. As shown in Figure 3, a commonly used convention for selecting frames of reference in robotic applications is the Denavit–Hartenberg or the DH agreement. In this method, each T_i homogeneous transformation between the two adjacent frames is depicted as the product of four basic transformations, as shown in Equation (1) [9]. $T_i = \text{Rot}(z, \theta_i) \text{Trans}(z, d_i) \text{Trans}(x, a_i) \text{Rot}(x, \alpha_i)$

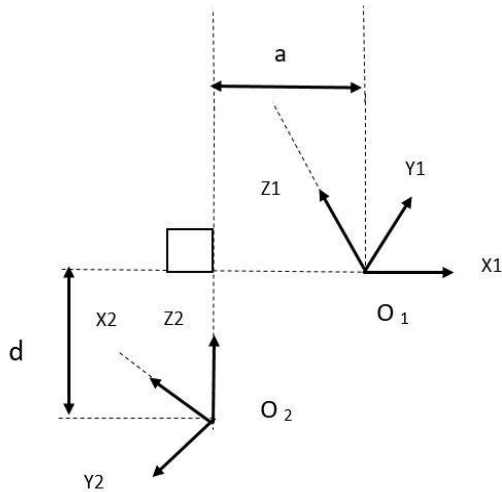


Figure 3: D-H Frame Assignment

Figure 3: D-H Frame Assignment

Where the notation $\text{Rot}(x, \alpha_i)$ stands for rotation about x_i axis by α_i , $\text{Trans}(x, a_i)$ is the translation along x_i axis by a distance a_i , $\text{Rot}(Z, \theta_i)$ stands for rotation about Z_i axis by θ_i , and $\text{Trans}(Z, d_i)$ is the translation

along Z_i axis by a distance d_i . Hence Equation (1) is rewritten.

And the general transformation matrix is the Standard DH Parameter Matrix ${}^{i-1}T_i$ as in Equation (2) and Equation (3).

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

and

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i * \sin\theta_i & \sin\alpha_i * \sin\theta_i & a_i * \cos\theta_i \\ \sin\theta_i & \cos\theta_i * \cos\alpha_i & \sin\alpha_i * \cos\theta_i & a_i * \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

We can formulate the transformation matrix between each two successive frames as follow:

After each link's DH coordinate system has been constructed, a homogeneous transformation matrix may be simply created using frame $\{i-1\}$ and frame $\{i\}$. This transformation consists of five basic transformations below. From the last matrix

0T_5 the position and orientation of the end-effector with respect to the base can be extracted.

In a similar way all the 0T_1 , 1T_2 , 2T_3 , 3T_4 , and 4T_5

Will be found as in Equations from (4) to (8).

$${}^0T_1 = \begin{bmatrix} C_{\theta_1} & 0 & S_{\theta_1} & 0 \\ S_{\theta_1} & 0 & -C_{\theta_1} & 0 \\ 0 & 1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$${}^1T_2 = \begin{bmatrix} C_{\theta_2} & -C_{(90)}S_{\theta_2} & 0 & L_2C_{\theta_2} \\ S_{\theta_2} & -C_{(90)}S_{\theta_2} & -S_{(90)}C_{\theta_2} & L_2C_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$${}^2T_3 = \begin{bmatrix} C_{\theta_3} & -S_{\theta_3} & 0 & L_3C_{\theta_3} \\ S_{\theta_3} & C_{\theta_3} & 0 & L_3S_{\theta_3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$${}^3T_4 = \begin{bmatrix} C_{\theta_4} & -S_{\theta_4} & L_4S_{\theta_4} & 0 \\ S_{\theta_4} & 0 & -L_4C_{\theta_4} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$${}^4T_5 = \begin{bmatrix} C_{\theta_5} & -S_{\theta_5} & 0 & 0 \\ S_{\theta_5} & C_{\theta_5} & 0 & 0 \\ 0 & 0 & 0 & L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Then 0T_5 could be formed by matrix multiplication of the individual link matrices. Starting by multiplying 4T_5

and 3T_4 , 2T_3 which is multiplied by 1T_2 and so on until 0T_5 is obtained as in Equation (9) and (10) [4]:
 $T_H = {}^0T_5 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5$ (9)

And by multiplying the expanded matrices, we get the total transformation matrix of the robot:

Where:

$${}^B T_W = {}^0 T_5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$n_x = c12 * c345 \quad (11)$$

$$n_y = s12 * c345 \quad (12)$$

$$n_z = s345 \quad (13)$$

$$o_x = s12 \quad (14)$$

$$o_y = -c345 \quad (15)$$

$$o_z = s12 \quad (16)$$

$$a_x = -c12 * s345 \quad (17)$$

$$a_y = s12 * s345 \quad (18)$$

$$a_z = -c345 \quad (19)$$

$$p_x = s12 * d5 = c12 * a4 * c34 + c12 * a3 * c3 \quad (20)$$

$$p_y = -c12 * d5 + s12 * a4 * c34 + s12 * a3 * c3 \quad (21)$$

$$p_z = a3 * s34 + a3 * s3 + d1 \quad (22)$$

where P_x , P_y , and P_z are global coordinates specifying the end effector's spatial position. Using MATLAB programming to multiply the individual matrices. Then, the results are;

Where:

$$c_n : \cos(\theta_n), s_n : \sin(\theta_n)$$

The last equations can be noted as Equation (11,12,13,14,15,16,17,18,19,20,21,22) to simplify using it. These equations give the FK of the designed robot arm. Knowing the robot variables ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$) then 0T_5 will be identified, and the position and orientation of the robot wrist relative to the base frame will be known.

Where the Four quantities ($\theta_i, a_i, d_i, \alpha_i$) are the parameters of link and joint. The various parameters in previous equation are given the following names:

a_i (link length) is the distance from z_i to z_{i+1} measured along z_i ;

α_i (link twist), is the angle between z_i to z_{i+1} , measured about x_i ;

d_i (link offset), is the distance from x_i to x_{i+1} , measured along z_i ; and

θ_i (link angle), is the angle between x_i to x_{i+1} measured about z_i ;

p_x : Position of the end-effector in x-direction = $a_i C_{\theta_i}$
 p_y : Position of the end-effector in y-direction

$$= a_i S_{\theta_i}$$

p_z : Position of the end-effector in z-direction

Figure 5 shows the robot's manipulator, which includes five linkage arms that start to align to the x-axis. A1, A2, A3, and A4 are the lengths of the links, accordingly.

As seen in the picture, the first link advances by θ_1 , the second link by θ_2 , the third link by θ_3 , the fourth link by θ_4 , and the fifth link by θ_5 . Figure 4 depicts the kinematic model with frame assignments based on Denavit & Hartenberg (DH) codes. Table 1 shows the kinetic parameters calculated using this model.

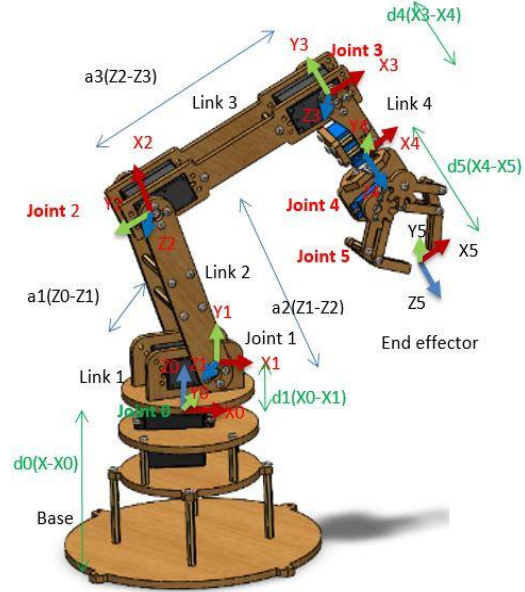


Figure 4: Links coordinates' diagram of the arm robot

Table 1: The link parameters of robot arm manipulator (D-H parameters).

Joint	α_{i-1} (°)	a_{i-1} (mm)	d_{i-1} (mm)	θ_{i-1} (°)
1	$\alpha_1 = 90$	a1	L1	0
2	$\alpha_2 = 0$	a2	0	θ_2
3	$\alpha_3 = 0$	a3	0	θ_3
4	$\alpha_4 = -90$	a4	0	θ_4
5	$\alpha_5 = 0$	0	d_5	θ_5
6	$\alpha_6 = 0$	0	0	Gripper

Table 2: The link lengths of robot arm 5DOF

Link Joint	Waist	shoulder	elbow	wrist
token	a1	a2	a3	a4
Link length (mm)	126.9	122	142	153

5. SIMULINK MODEL DEVELOPMENT AND DYNAMICS SIMULATION (OPEN AND CLOSED LOOPS USING PID AND FLC)

Simulink® is a software program for modeling, simulating, and analyzing dynamic systems developed by Mathworks Inc. [40]. It supports linear and nonlinear systems modeled in sequential time, sampled time, or a combination of the two. The system model is represented using block diagrams, describing the system mode [41].

5.1. Simulink Model of the Robotic Arm

The robotic arm model developed in SolidWorks was transferred into Matlab using the option “EXPORT-SIMSCAPE MULTI-BODY FIRST GENERATION,” where the file is a CAD assembly file exported to MATLAB as an XML file format based on the SimMechanics library.

Simulation is performed to verify the design dynamics in the form of a Simulink model containing the blocks shown in Figure 6(a). The blocks imported into Simulink are rearranged according to engineering laws, physical laws, and the required assembly. The Simulink model has three main block types: the input block, the system block, and the output block. The input block shows a simulation of the reference input signals (positions of joints) designed to obtain the arm robot’s temporal response (gripper position) before adding the controls. System blocks represent robot arm links and joints and the final gripper. The output block represents the gripper position. Table 3 shows the parts in the SolidWorks environment and their counterparts in the Matlab environment. Figure 6(b) shows a 3D visualization of the system blocks.

Table 3. Robot arm components in SolidWorks vs. its components in Simulink.

No.	Part in SolidWorks	Part in Matlab (SimMechanics)
1	part	Body
2	Mates	Joint
3	- Fundamental ROOT	Ground – Root weld – ROOT Body
4	Subassembly Root :(trunk –mates.....)	Subsystem linking robotic arm parts
5	Fixed part	Fixed part

5.2. Kinematics Model Simulation

Kinematics is a branch of physics that investigates motion without accounting for the forces that cause it. The study of position, velocity, acceleration, and higher derivatives of position variables is covered in this course.

Forward kinematics (FK) and inverse kinematics (IK) are the two kinematics solutions for robotic arms (IK). If the locations of all joints are known, FKs identify the locations of the robot’s gripper. As a result, FK stands for conversion from configuration to Cartesian space. In contrast, IK stands for transformation from Cartesian space to joint space. Robots (FK) are simulated in this work by calculating the end-effector coordinates knowing their joint angles. The approach used for kinematic modeling is the D-H. First, the D-H parameters are calculated via the D-H formulation method. The kinetic model may then be characterized by identifying these parameters [42, 43] by the D-H parameters presented in Table 1. Further, a 3D conception of the system blocks is shown in Figure 5(b).

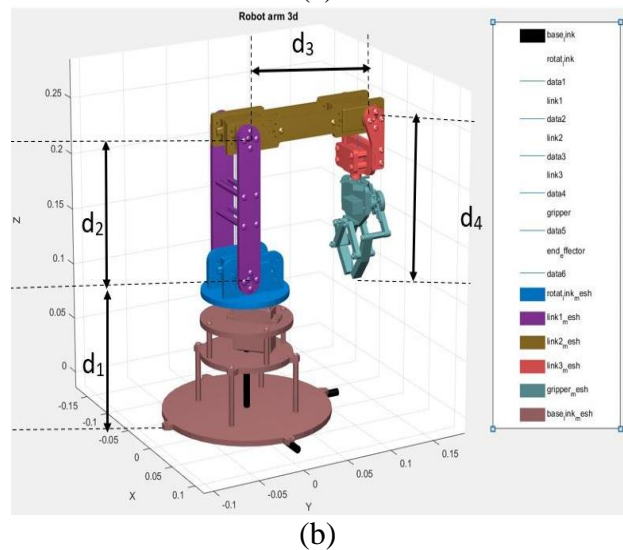
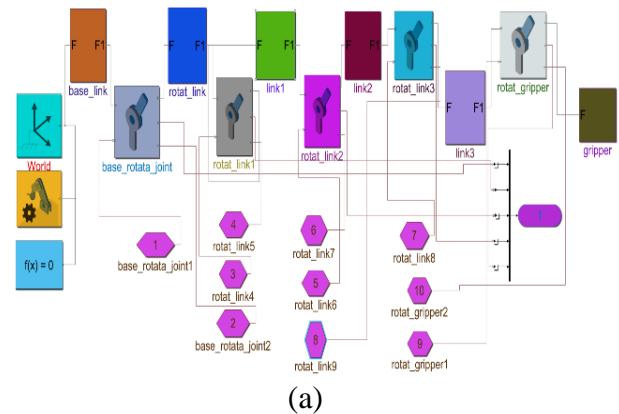


Figure 5: Robot arm model in Simulink, (a) Model blocks, (b) 3D visualization of the system blocks.

5.3. Control System Design

The servo motor Simulink model, open loop dynamics experiments, and controllers design and simulation will be presented in this section.

5.3.1 Servo Motor Simulink Model

Many mechanical systems and industrial applications, such as industrial and educational robots, use DC servo motors as an actuator. [44]. The main reason to use a servo motor in this study is that the position, speed, and torque of the servo motor can be

values were substituted from the motor specification datasheet. Table 4 shows the DC servo motor parameters based on the datasheet.

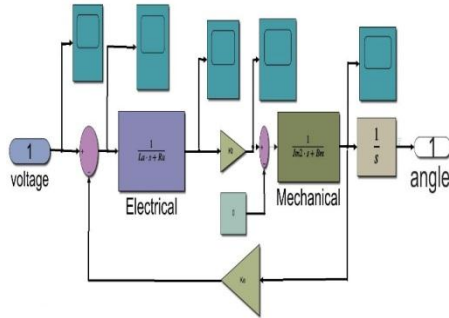


Figure 6: DC servo motor Simulink model block diagram.

Table 4: DC servo motor Parameters.

Parameters	Values	Unit
Moment of inertia (Bm)	0.0243	kg.m ²
Friction coefficient (Jm)	0.000026852	N.ms
Back EMF constant (Kb)	1.058	v/ms-1
Torque constant (Kt)	0.884	Nm/A
Electric resistance (Ra)	3.33	Ohm
Electric inductance (La)	0.000000015	H

The following equation depicts the total transfer function of a DC servo motor system. (23):

$$\frac{\theta_m(S)}{v_t(s)} = \frac{0.884}{3.315e^{-08} S^3 + 0.0001856 S^2 + 1.016 S} \quad (23)$$

where:

v_t : voltage input

θ_m : angle output

S: Laplace variable

5.3.2 PID Controller Design and Simulation

The PID algorithm is the most widely used feedback controller in the industry. It is a sturdy and easy-to-comprehend algorithm that provides excellent control effectiveness regardless of the dynamic behavior of a specific process plant.

$$G_{pid}(s) = k_p + k_i/s + k_d.s \quad (24)$$

controlled as needed, which is essential when building a robotic arm. The model of the DC servo motor was created in Simulink, as shown in Figure 6. The DC servo motor block diagram consists of the electronic and mechanical parts, and the parameter

The calculating algorithms offered in Equation (24) include three separate stationary coefficients: p, i, and d. These coefficients can be explained in terms of time, where p is for existing errors, and i is for the accumulation of past errors. Coefficient d means approaching error based on the uninterrupted pace of change [45].

Parameters of the PID controller were tuned using Simulink instead of traditional tuning methods ZN.

Figure 8 shows the system diagram of the robot arm 5-DOF framework and PID controllers. Five PID controllers were developed, one for each DC servo motor associate with each joint.

5.3.2. Control System Design

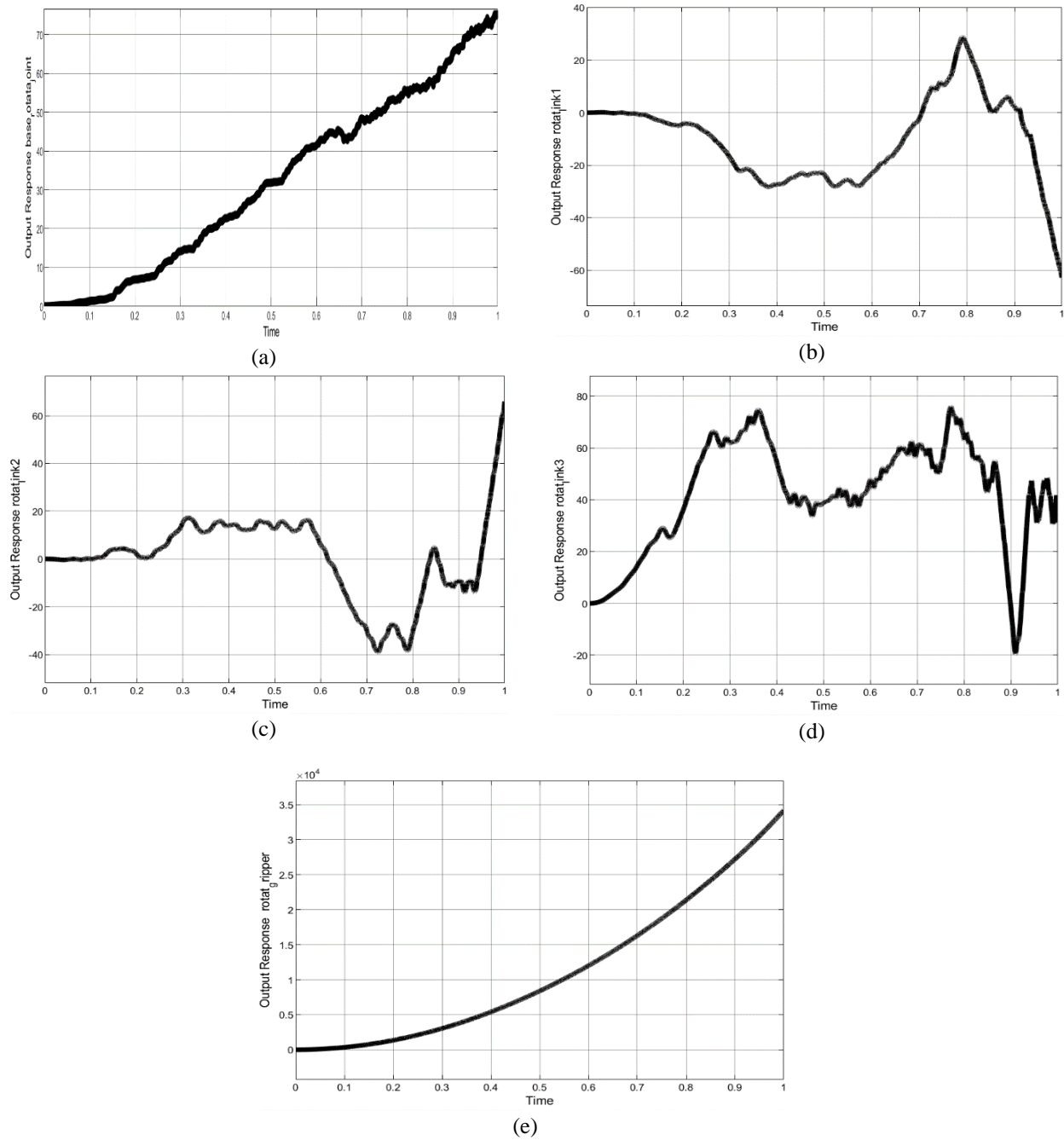


Figure 7: Input and output response of joints in open loop dynamics; Input and output position of: (a) joint 1, (b) joint 2, (c) joint 3, (d) joint 4, (e) joint 5.

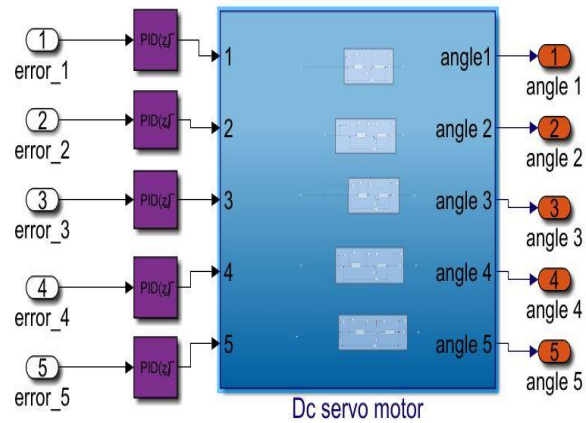
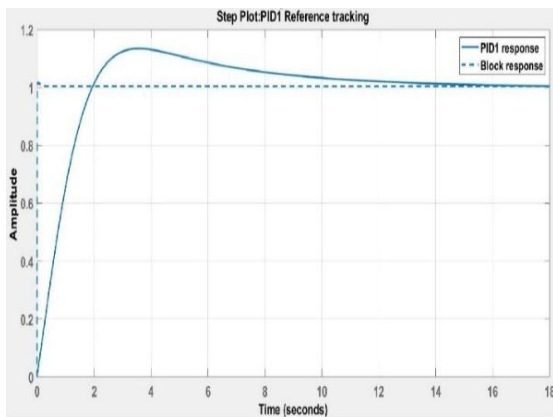


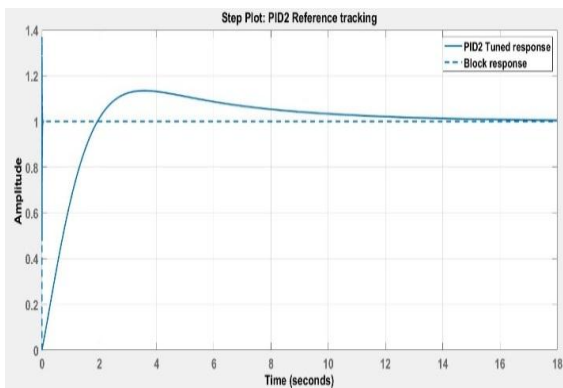
Figure 8: Block diagram of arm robot system with PID controllers.

The step response and parameters of the five developed PIDs are shown in Figure 9.



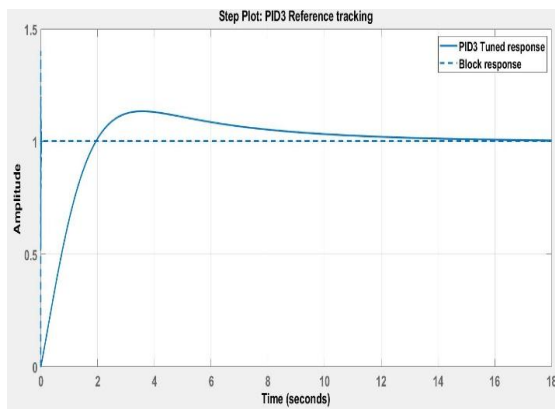
Controller Parameters		
	Tuned	Block
P	0.020618	5
I	0.0039429	25
D	-0.0043291	0.006
N	1.719	1000
Performance and Robustness		
	Tuned	Block
Rise time	1.43 seconds	0.008 seconds
Settling time	12.2 seconds	0.015 seconds
Overshoot	13.5 %	1.76 %
Peak	1.13	1.02
Gain margin	73.2 dB @ 3.14e+03 rad/s	11 dB @ 3.14e+03 rad/s
Phase margin	69 deg @ 1 rad/s	95.6 deg @ 269 rad/s
Closed-loop stability	Stable	Stable

(a)



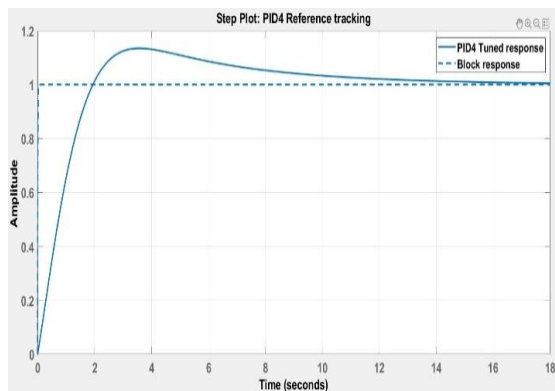
Controller Parameters		
	Tuned	Block
P	0.020611	10
I	0.0039397	2
D	-0.0042772	0.06
N	1.719	1000
Performance and Robustness		
	Tuned	Block
Rise time	1.43 seconds	0.001 seconds
Settling time	12.2 seconds	0.029 seconds
Overshoot	13.5 %	36.8 %
Peak	1.13	1.37
Gain margin	67.3 dB @ 755 rad/s	2.46 dB @ 1.46e+03 r...
Phase margin	69 deg @ 1 rad/s	22.2 deg @ 1.18e+03...
Closed-loop stability	Stable	Stable

(b)



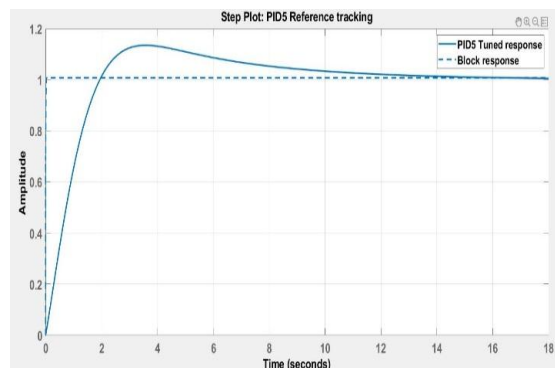
Controller Parameters		
	Tuned	Block
P	0.020611	10.9
I	0.0039394	17.5
D	-0.0042725	0.06
N	1.719	1000
Performance and Robustness		
Rise time	1.43 seconds	0.001 seconds
Settling time	12.2 seconds	0.025 seconds
Overshoot	13.5 %	40 %
Peak	1.13	1.4
Gain margin	67.2 dB @ 726 rad/s	2.9 dB @ 1.44e+03 rad/s
Phase margin	69 deg @ 1 rad/s	25.3 deg @ 1.11e+03 rad/s
Closed-loop stability	Stable	Stable

(c)



Controller Parameters		
	Tuned	Block
P	0.020618	2.9
I	0.0039429	0.2
D	-0.0043301	0.001
N	1.719	1000
Performance and Robustness		
Rise time	1.43 seconds	0.015 seconds
Settling time	12.2 seconds	0.026 seconds
Overshoot	13.5 %	0.0474 %
Peak	1.13	1
Gain margin	72 dB @ 3.14e+03 rad/s	20.6 dB @ 3.14e+03 rad/s
Phase margin	69 deg @ 1 rad/s	87.5 deg @ 145 rad/s
Closed-loop stability	Stable	Stable

(d)



Controller Parameters		
	Tuned	Block
P	0.020618	5
I	0.003943	10
D	-0.0043319	0.01
N	1.719	1000
Performance and Robustness		
Rise time	1.43 seconds	0.01 seconds
Settling time	12.2 seconds	0.019 seconds
Overshoot	13.5 %	0.74 %
Peak	1.13	1.01
Gain margin	70.8 dB @ 3.14e+03 rad/s	5.26 dB @ 3.14e+03 rad/s
Phase margin	69 deg @ 1 rad/s	109 deg @ 313 rad/s
Closed-loop stability	Stable	Stable

(e)

Figure 9: The step response and controller parameters of: (a) PID1, (b) PID2, (c) PID3, (d) PID4, and (e) PID5

5.3.2 Experiment 2

controller, the simulation was performed using the suggested PID controllers. The system model in Figure 9 (robot model with PID controllers) was subjected to two sets of reference step input signals representing the joints' position angles. The values of joint angles values were selected based on the space limit of each joint. Experiments were repeated for positive and negative values of Input joint angles. Each set had different joints' position angle values, the 1st set had positive values, and the 2nd set had negative values. Simulations were performed for 4 msec. The feedback signals of the joints'

position angles were then measured. All joints' speed, acceleration, torque, and the posture of the end effector (grabber) were also measured. The first set of reference step input signals was used to apply to Experiment 2(a), and the results are shown in Figure 10. The second set of

reference step input signals was used to apply to Experiment 2(b), and the results are shown in Figure 11.

a. Results of Experiment 2(a)

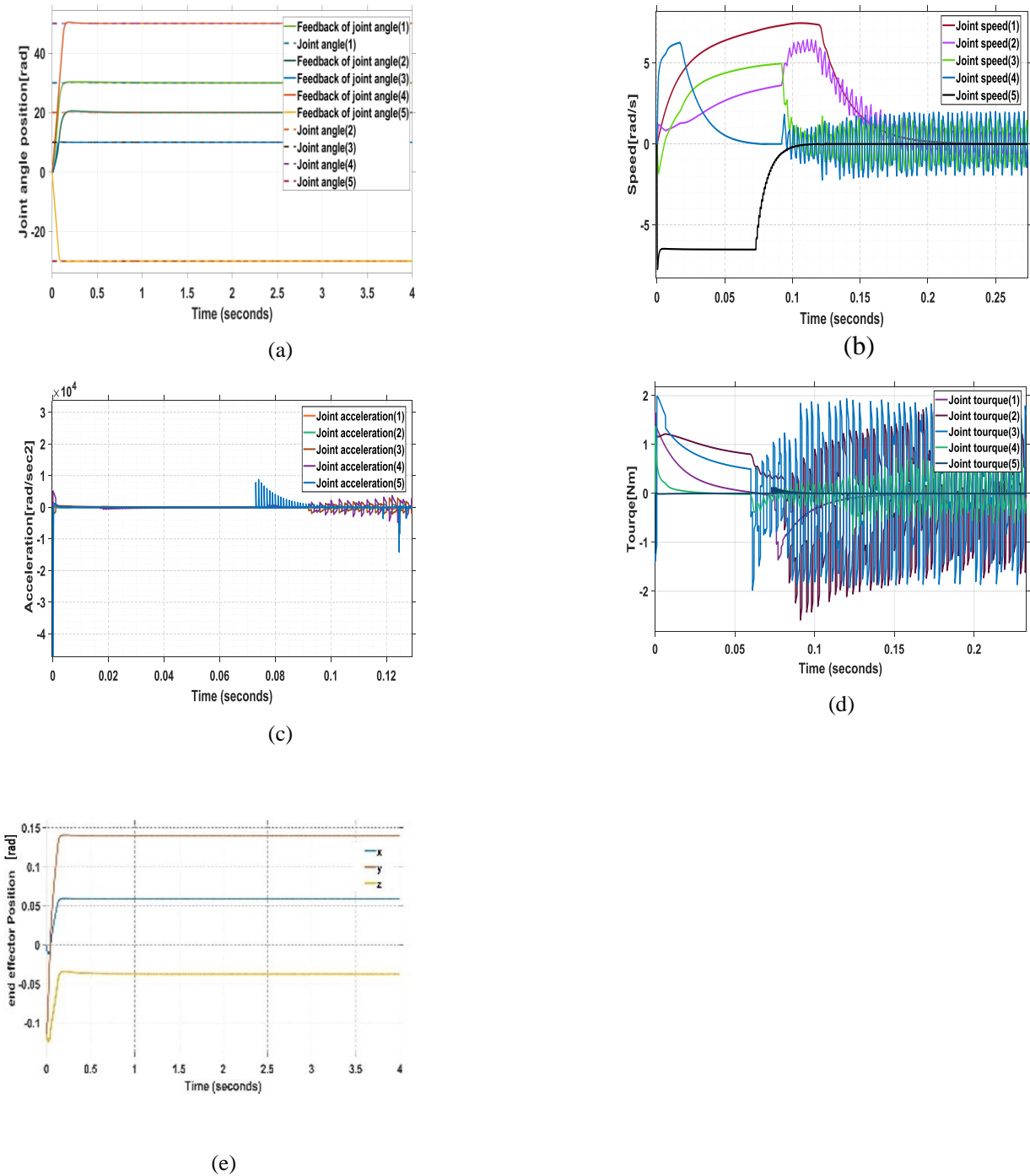
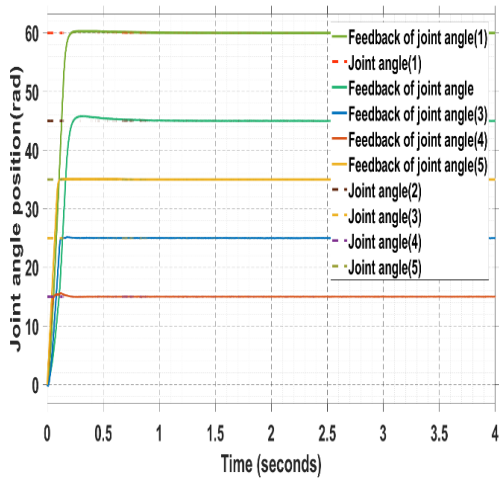
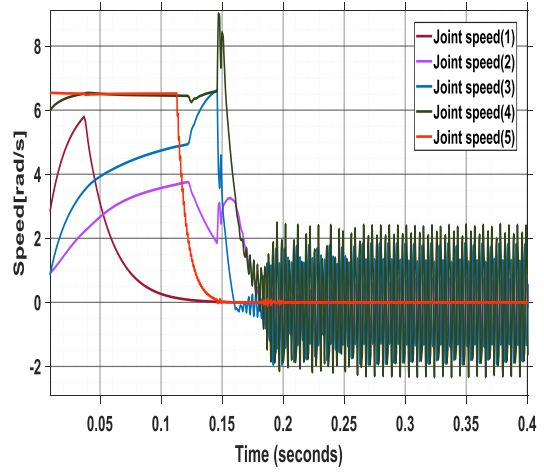


Figure 10: Simulation results of Experiment 2(a): (a) Input and output values of joints angles, (b) output values of joints speed,(c) output values of joints acceleration, (d) output values of joints torque (e) end effector position.

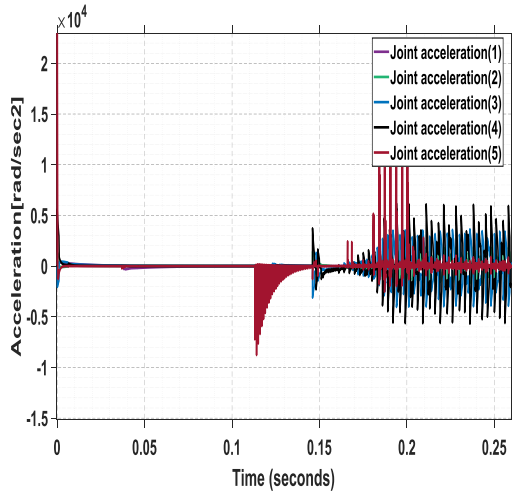
a. Results of Experiment 2(b)



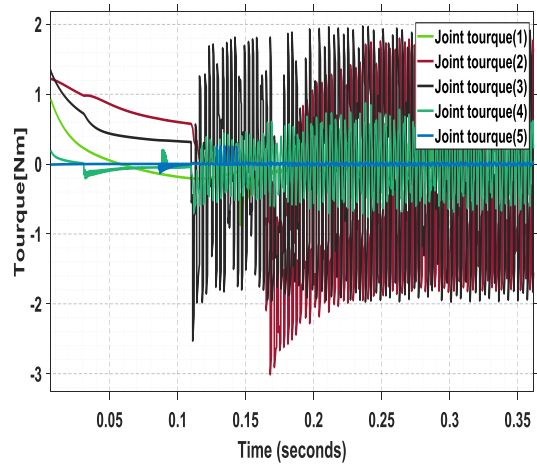
(a)



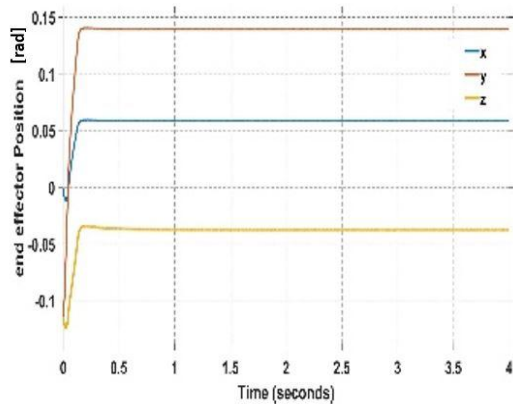
(b)



(c)



(d)



(e)

Figure 11. Simulation results of Experiment 2(b): (a) input and output values of joints angles, (b) output values of joints speed, (c) output values of joints acceleration, (d) output values of joints torque, and (e) end-effector position.

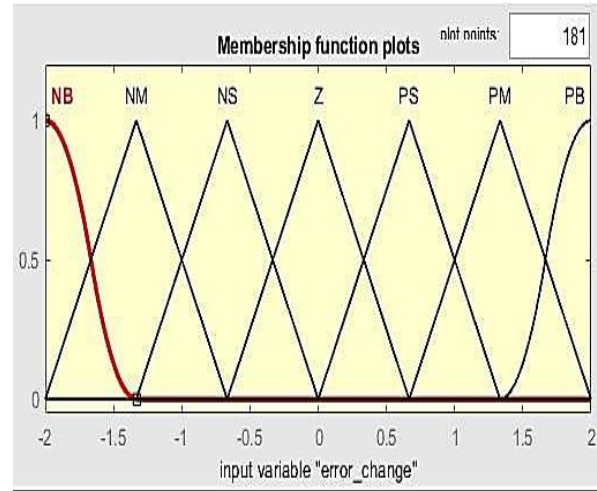
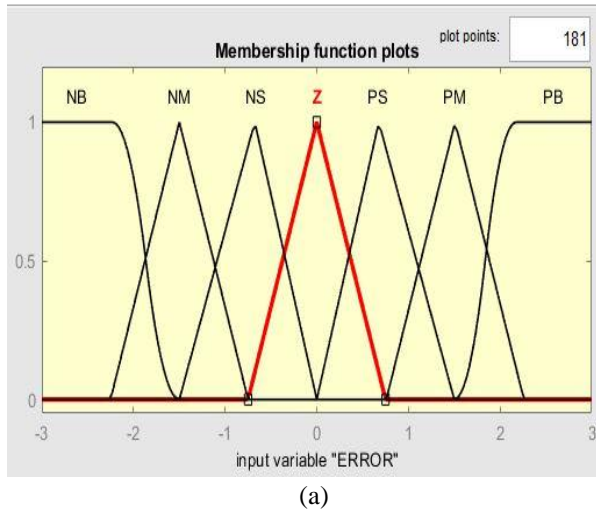
The closed-loop system outperforms the open-loop system. When torque is applied to a joint, the gravity on the link depends on the covariates. Hence, all joints provide the correct position, the response of each joint is stable, and system disturbances are small, leading to a perfectly controlled motion. In addition, the presence of feedback contributed to the system's stability.

5.3.4. FLC Design and Simulation

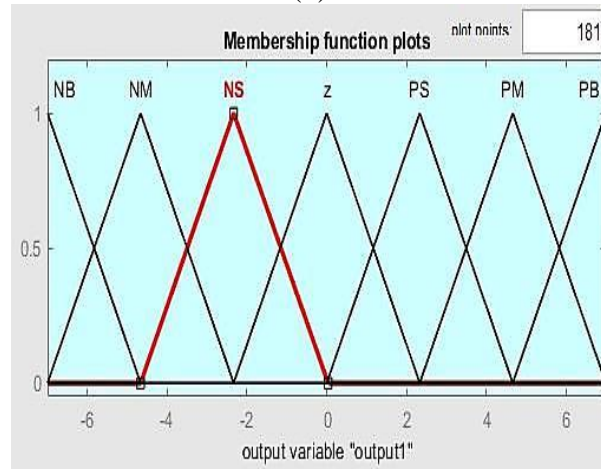
Fuzzy logic systems are easy to understand and design and perform better than other types of controllers. By executing basic principles guiding the system's behavior, FLC is transformed into an automated manner altering the language of the control step. Fuzzy logic enables the modeling of complicated systems that originates from the information and mastery by combining substitutional way of thinking using a higher level of the inference procedure is divided into four parts as follows:

Fuzzification of input variables, rule evaluation, aggregation of rule outputs, and defuzzification [46] The implementation of the FLC [47] is as follows:

1) Identifying FLC input and output. An FLC has two inputs: error $E(t)$ and change of error $\Delta E(t)$, and one output is the design control signal. 2) Fuzzifying input and output variables. Each variable of FLC inputs has seven fuzzy sets ranging from negative big (NB) to positive big (PB). 3) Determining the input-output relationship and designing an inference mechanism rule. This paper uses Mamdani (Pro Max) inference. 4) Defuzzifying the output variable of the fuzzy mechanism. Different defuzzification methods were used and compared to obtain the control signal. Figure 12 shows the fuzzy membership functions of $E(t)$,



(b)



(c)

Figure 12: Membership functions of (a) $E(t)$, (b) $\Delta E(t)$, and (c) output.

Table 4: Rule base description.

No.	Rule Description
1	If (ERROR is Z) and (error_change is NB) then (output1 is NS)
2	If (ERROR is Z) and (error_change is NM) then (output1 is z)
3	If (ERROR is Z) and (error_change is NS) then (output1 is z)
4	If (ERROR is Z) and (error_change is Z) then (output1 is z)
5	If (ERROR is Z) and (error_change is PB) then (output1 is PS)
6	If (ERROR is Z) and (error_change is PM) then (output1 is z)
7	If (ERROR is Z) and (error_change is PS) then (output1 is z)

Rule base and membership functions were designed as the rule characterization shown in Table 4. For the error signal $E(t)$ as an input and control signal $C(t)$ as an output, NB stands for negative big, NM means negative mean, Z means zero, PS means positive small, PM

means positive mean, and PB means positive big. To vary the error (ΔDE) as input, N means negative, and P Means positive. These are the linguistic variables that mean each time uneven fuzzy controller inputs

Where:

- Nb: Negative big
- Nm: Negative medium
- Ns: Negative small
- Z: Zero
- Ps: Positive small
- Pm: Positive medium
- Pb: Positive big

These are the linguistic variables that describe each of the time varying fuzzy controller inputs and outputs are used to define the rule base of the fuzzy controller.

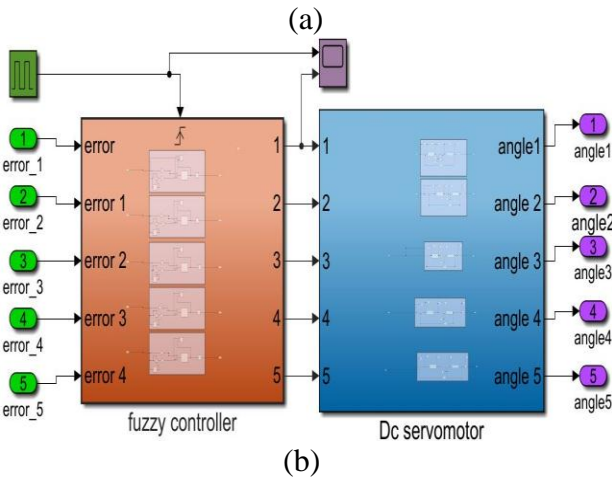
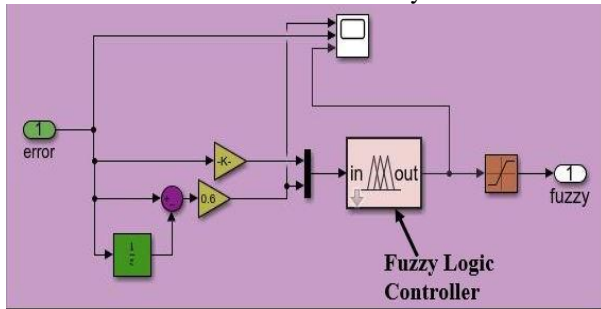


Figure 13: Simulink model block diagram of: (a) FLC; (b) Robot arm system with FLC.

And outputs that are used to define the rule base of the fuzzy controller. Figure 13 (a) shows the Simulink model for the FLC.

Figure 13 (b) shows the robotic arm Simulink model with the FLC added. An FLC was developed for each DC servo motor related to each joint.

Defuzzification is a method that converts a federation of fuzzy sets into a crisp value to produce a non-fuzzy action. The most common approaches are as follows: (1) center of gravity (COG), (2) bisector of area (BOA), (3) denotes of maximum (MOM), (4) smallest of maximum (SOM), and (5) largest of maximum (LOM). The purpose of COG, where the crisp control value " COG is the abscissa of the center of gravity of the fuzzy set, u_{COG} is calculated as follows in Equation (25) :

$$u_{COG} = \frac{\sum_i \mu_c(x_i)x_i}{\sum_i \mu_c(x_i)} \quad (25)$$

where X_i is a point in the world of the conclusion ($i = 1, 2, \dots$), and $\mu_c x_i$ is the membership value of the resulting conclusion set. For continuous sets, summations are replaced by integrals.

The BOA defuzzification technique calculates the coordinates of the vertical line that swears the area of the extracted membership function into two analogous areas as in Equation (26).

$$\left| \sum_{i=1}^j \mu_c(x_i) - \sum_{i=j+1}^{i_{max}} \mu_c(x_i) \right|, i < j < i_{max} \quad (26)$$

where i_{max} is the index of the largest abscissa x_i , and BOA is considered a computationally difficult procedure.

Another option for obtaining the crisp value is to select a point with the most members. There may be several points with the highest membership value in the total inferred fuzzy set. As a result, calculating the average value of these points is a common method. Using the method called MOM, the crisp (brittle) value is determined as in Equation (27):

$$u_{MM} = \frac{\sum_{i \in I} x_i}{|I|}, I = \{i | \mu_c(x_i) = \mu_{max}\} \quad (27)$$

where i is the crisp set of indices i , $\mu_c(x_i)$ reaches its maximum μ_{max} , and $|I|$ is its cardinality, a term used to describe the phenomenon of the number of members. The

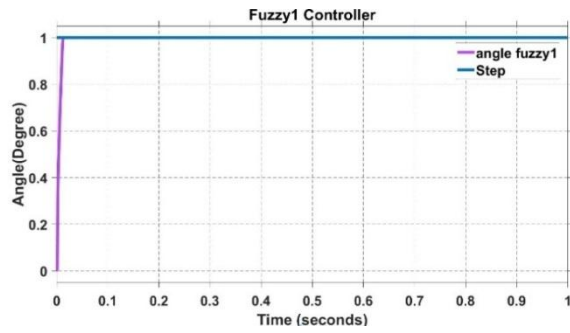
leftmost point among the points with the highest membership in the inferred fuzzy set can also be chosen. This procedure is referred to as the SOM defuzzification method. The crisp value is determined as in Equation (28):

$$u_{SOM} = x_{\min(I)} \quad (28)$$

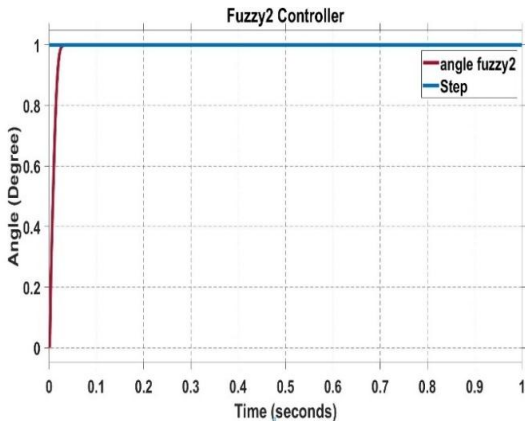
Another option is choosing the rightmost point among the points with maximum membership to the overall inferred fuzzy set. This technique is called the LOM defuzzification technique, where the crisp value is calculated as in Equation (29):

$$u_{LOM} = x_{\max(I)} \quad (29)$$

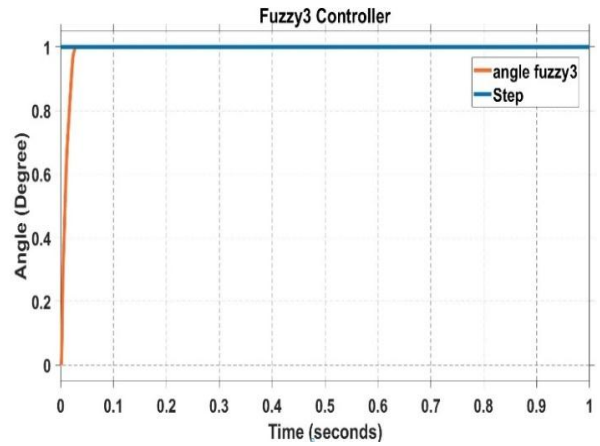
These methods have been suggested in the literature. Figure 14 shows the step response of the five developed FLCs related to each DC servo motor and joint. Table 5 shows the error values, allowing a quick comparison of the parameters of the PID and FLC controllers.



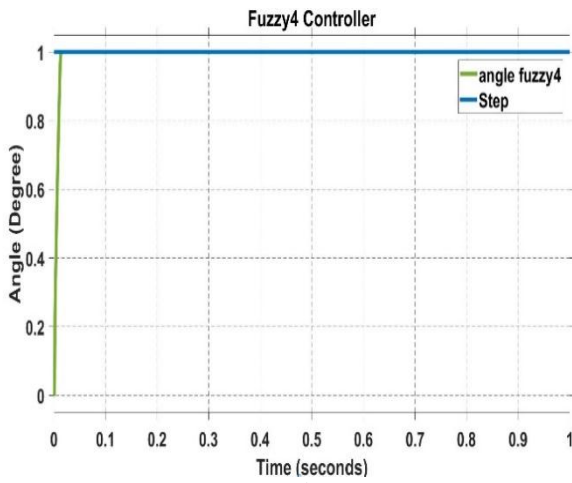
(a)



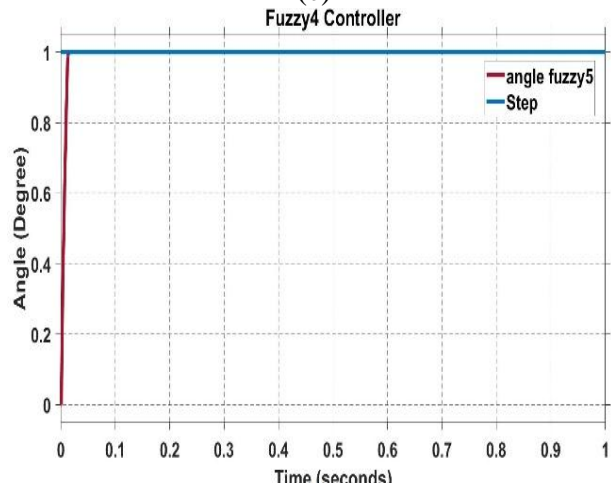
(b)



(c)



(d)



(e)

Figure 14: Step response of FLC developed for dc servo motor related to each joint: (a) FLC1, (b) FLC2, (c) FLC3, (d) FLC4, and (e) FLC5.

1- Experiment 3

The suggested FLC controllers used in the simulation evaluate each controller's performance. The system model in Figure 14(b) (robot model with FLC controllers) was subjected to the same two sets of reference step input signals used in Experiment 2

simulation, enabling result comparison of Experiments 2 and 3 (PID and FLC results). The input sets of reference step input signals represent the joints' position angles. Simulations were performed for 1 msec. the values of joint angles values were selected based on the space limit of each joint Experiments were repeated for positive and

negative values of Input joint angles The feedback signals of the joints' position angles were measured. The speed, acceleration, and torque of joints and the end-effector location were measured. The calculated steady-state error was recorded in Table 6. The first set of reference step input signals was applied in Experiment 3(a).

The second set of reference step input signals was applied in Experiment 3(b).

a. Simulation Results of Experiment 3(a)

Figure 15 shows the simulation results of Experiment 3(a): (a) Input and output values of joints angles, (b) output values of joints speed, (c) output values of joints acceleration, (d) output values of joints torque (e) end effector position.

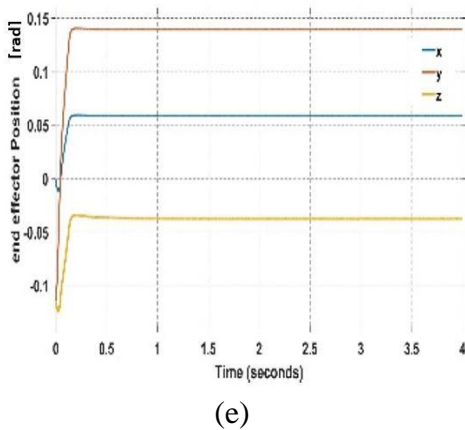
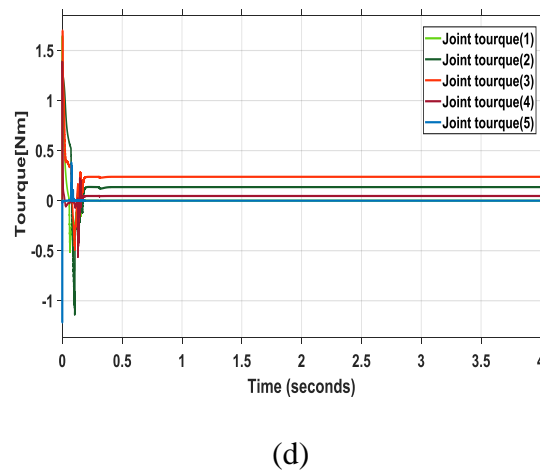
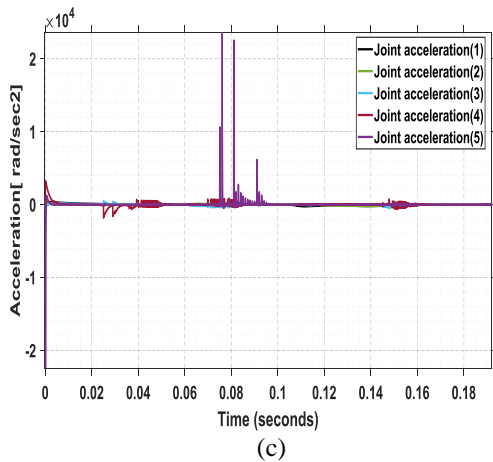
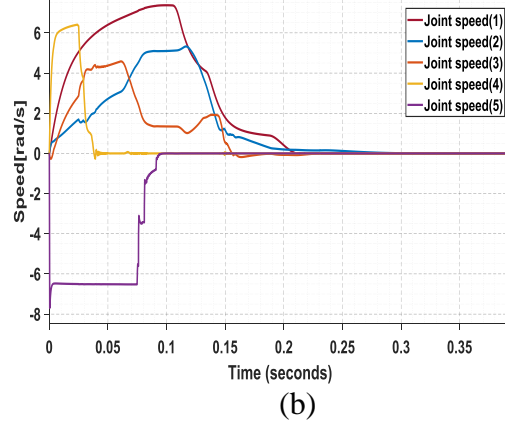
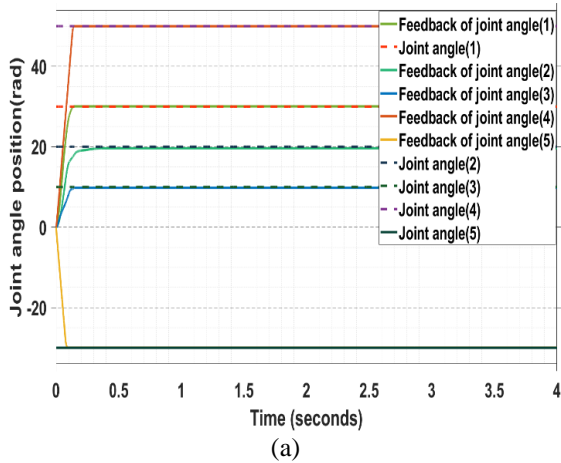


Figure 15: Simulation results of Experiment 3(a):
Input and (a)
output values of joints angles,
(b) output values of joints
speed,(c) output values of joints
acceleration, (d) output values of
joints torque
(e) end effector position.

b- Simulation Results of Experiment 3(b)

Figure 16 shows the simulation results of Experiment 3(b): (a) input and output values of joints angles, (b) output values of joints speed, (c) output values of joints acceleration, (d) output values of joints torque and (e) end-effector position.

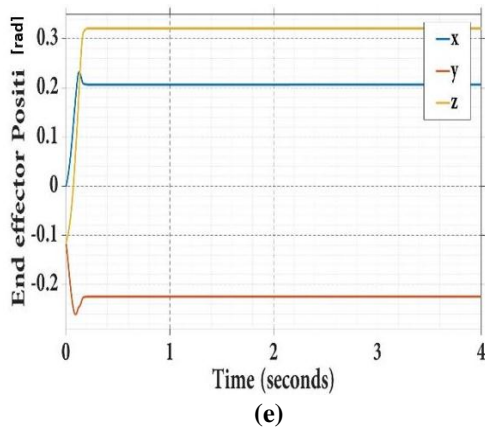
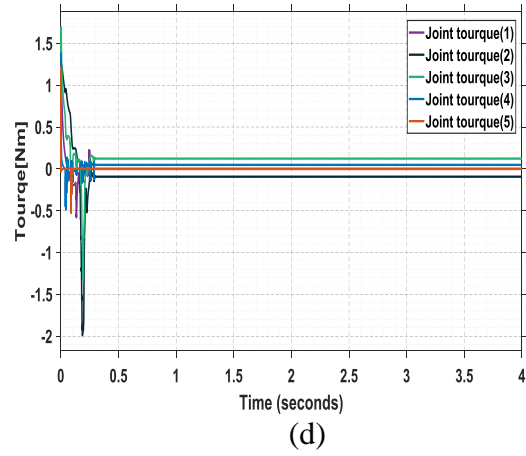
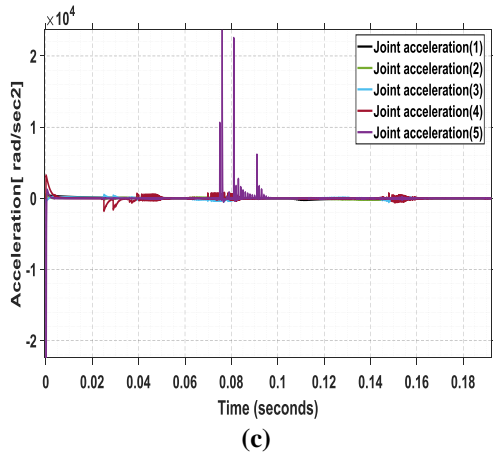
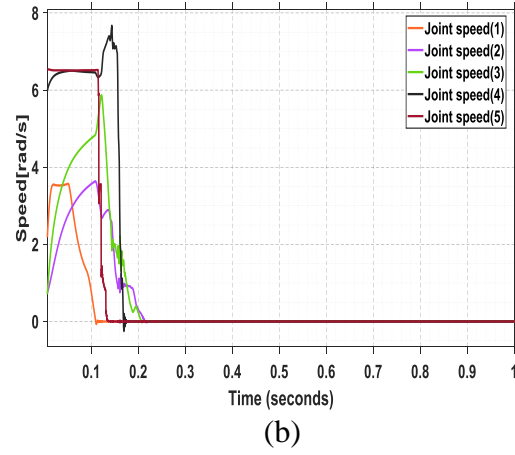
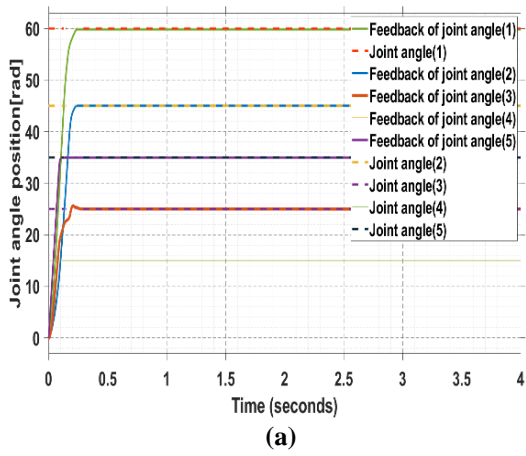


Figure 16: The simulation results of Experiment 3(b): (a) Input and output values of joints angles, (b) output values of joints speed, (c) output values of joints acceleration, (d) output values of joints torque (e) end effector position.

The findings reveal that all controllers can complete the intended movement of the robot arm's servo motors. The time response parameters, including rise time (rt), settling time, and steady-state error (SSE) of the PID controller and FLC system of the higher-order system transfer function of the DC servo motor of the processor, were also obtained. The FLC resulted in reduced rise time, stabilization time, SSE, and bypass than the PID controller. The comparison between the two controllers is shown in Table 5.

Table 6 displays the results of several defuzzification procedures. Table 6 shows that the BOA, MOM, and SOM techniques yield almost identical results; however, the COG strategy yields a broad range of outcomes.

Implementing a simple defuzzification approach resulted in system optimization due to the complexity of processes like fuzzification and defuzzification. In other words, the CG method should be avoided. when comparing our results to the results given in [31], we found that the performance of our FLC gave better results in terms of Rise time (rt) , Settling time (st) , The results obtained using the four defuzzification strategies are shown in Table 7. This table shows that the BOA, mean of maximum, and SOM strategy yield approximately the same results. In contrast, there are wide variations in the COG approach results. Due to

complex operations such as fuzzification and particularly defuzzification, implementing a simplified defuzzification strategy optimizes the system. This means that the COG strategy must be avoided. Overshoot, Steady State Errors (SSE) as shown in Table 6.

6.ROBOT ARM PROTOTYPE AND CONTROL CIRCUIT

In the mechanical characterization of the robotic arm, an appropriate design solution was studied in terms of cost and the ability to 17(a), and a schematic diagram of the control circuit is shown in Figure 17(b). Table 8 shows the control circuit components and their specifications.

7.RESULTS AND DISCUSSION

Robotics has recently become an exciting area of research. This paper studies the robot manipulator from two aspects: modeling and control. The modeling process includes kinematic analysis and DC motor modeling. This process is important before controlling a robot to save it from being damaged. Using a control

Table 5: compared between PID and FLC controllers' parameters.

Motors (joint No.)	System properties				
	Controller Type	Rise time (rt)	Settling time (st)	Steady State Errors (SSE)	Overshoot
Motor1 (at joint 1)	PID	0.26514	0.015685	0.03125	0.03
	FLC	0.146253	0.011897	0.00612	0.016446
Motor2 (at joint 2)	PID	0.246084	0.030299	0.00108	0.008109
	FLC	0.225239	0.023342	0.000807	0.0023701
Motor3 (at joint 3)	PID	0.274915	0.026104	0.001985	0.069705
	FLC	0.107747	0.024641	0.001256	0.0015471
Motor4 (at joint 4)	PID	0.2714389	0.025763	0.014598	0.0028283
	FLC	0.094743	0.012818	0.00125	0.011673
Motor5 (at joint 5)	PID	0.212219	0.020554	0.00234	0.0176497
	FLC	0.0064693	0.012827	0.00254	0.0013509

Table 6: Comparison of the FLC parameters of the developed robot arm and the FIC in study of [31].

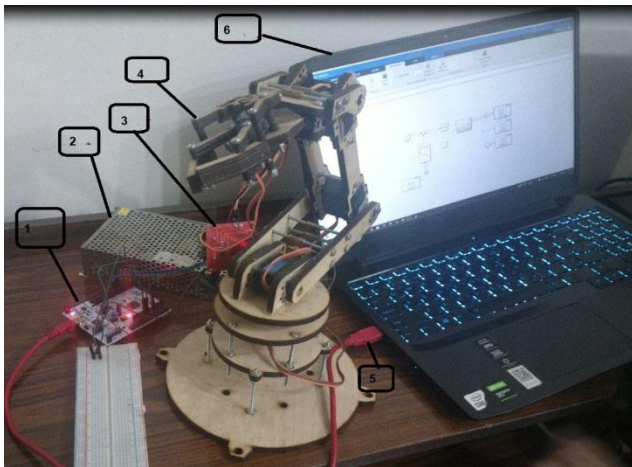
System output characteristics	Controllers					
	Link 1 (FLC) to robot arm	Link 1 (FLC) in study	Link 2 (FLC) to robot arm	Link 2 (FLC) in study	Link 3 (FLC) to robot arm	Link 3 (FLC) in study
Rise time (rt)	0.146253	1.8507	0.225239	1.1373	0.107747	0.6444
Settling time (st)	0.011897	3.3064	0.02342	2.4064	0.024641	1.4413
Overshoot	0.016446	2.2594e-04	0.0023701	7.2402e-04	0.0015471	1.2071e-04
Steady State Errors (SSE)	0.00612	-0.001	0.00807	-0.002	0.001256	-0.003

Table 7: compared between different defuzzification strategies of FLC controllers.

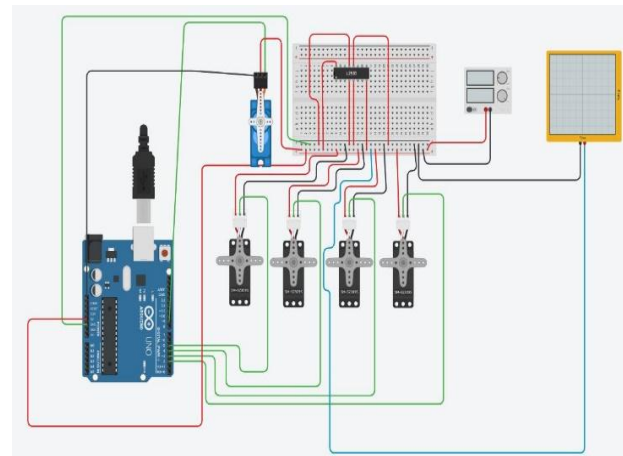
System properties				
Defuzzification method	Rise time (rt)	Steady State Errors (SSE)	Overshoot	Settling Time (st)
COG	0.0086253	1.9507e-05	0.00044694	0.011897
SOM	0.0063461	0.01309	7.8937	0.99954
MOM	0.0066253	0.01304	0.00044694	0.011897
BOA	0.0064803	0.0033492	3.1163	0.99958

Table 8: Specifications of control circuit components.

No.	Part	No. of	Specifications	Function
1	Servo Motor) 360)	5	Torque: 17 kg.cm Current: 130mA step angel: 1.8 Motor speed: 800 r.p.m	-Robot forward and backward motion, -Robot cleaning motion
2	Arduino Mega 2560 board	1	A microcontroller board based on the AT mega 2560, with 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator.	Control program development.
3	Dual H-bridge motor driver using L298N	5	L298N Dual H Bridge DC Stepper Motor Drive Controller Board Module for Arduino the L298 Stepper Controller makes it easy to drive either two dc motor or a bipolar stepper motor. This is a very high quality board and is very compact for designs where space really matters.	Contact the motor to the control circuit
4	Power supply	1	5 DC volt, 10 mA	DC power source
5	USB 2.0 cable	1	Data transfer speed is 480 Mb/sec	Connect the Arduino board to PC.
6	Adaptor	1	Input 100-240V, 50/60 Hz, 0.2A Output 5V	Connect the Arduino board to power source



(a)



(b)

(b)Figure 17:(a) Real photo of robot system with control circuit;1: Arduino Mega 2560 board, 2: Power supply, 3: Servo motor driver, 4: Robot arm, 5: USB 2.0 cable, 6: Laptop;(b)Schematic diagram of control circuit .

technique is important to guarantee high efficiency and less error for the motion of a robot. The desired tasks were accomplished using three stages: The first stage provided systematic rules for analyzing forward and IK solutions for robotic manipulators with revolute and prismatic joints using DH parameters. We then analyzed the mathematical model of the DC motor. In the second stage, we discussed the problem of control techniques. A PID controller was used to control a robotic manipulator. FLC was then implemented as a second choice to control the robot. with different defuzzification strategies was employed. First, an FLC was used to enhance the nonlinearity of the robot arm. FLC was designed based on Mamadani (pro Max) inference. Five different defuzzification methods were

used and compared to obtain the control signal. These are (1) center of gravity (COG), (2) bisector of area (BOA), (3) denotes of maximum (MOM), (4) smallest of maximum (SOM), and (5) largest of maximum (LOM). In the third stage, we compared the results of using the two controllers for controlling the robot manipulator. All simulations were performed using MATLAB and SIMULINK, both widely used in control applications.

8.CONCLUSIONS

A 5 DOF arm robot was designed and motion space was simulated using solidworks software. The forward kinematics of the arm robot was inferred mathematically and checked by Simmechanics software in addition to the robot dynamics. PID controller was designed and

This study aimed to control a 5-DOF robot arm to reach a specified position with minimal error while meeting certain specifications. This paper did not consider the tracking path from the initial to the final position. The final position for each motor was set using an independent joint control method. A feed-forward method was

employed to overcome the disturbances loaded on each motor. The system is a 5-DOF model, i.e., it has five motors, each of which can control its position independently. In this paper, we applied the Mamdani method to FLC. This method was applied with 49 rules to control a robot arm.Both controllers used the center of area defuzzification method and the min-max inference mechanism.

This paper presents the simulation and numerical results of the controllers so far. One of the most problems with fuzzy controller is that the computing time is longer than that for PID. The results prove that the FLC is more efficient in time response behavior than the PID controller. Performance comparisons were made using transient and

tuned using Simmechanics instead of using conventional tuning methods ZN. FLC was designed based on Mamadani (pro Max) inference. Five different defuzzification methods were used and compared to obtain the control signal. These are (1) center of gravity (COG), (2) bisector of area (BOA), (3) denotes of

maximum (MOM), (4) smallest of maximum (SOM), and (5) largest of maximum (LOM). From results BOA, MOM, and SOM techniques yield almost identical results; however, the COG strategy yields a broad range of outcomes. Implementing a simple defuzzification approach resulted in system optimization due to the complexity of processes like fuzzification and defuzzification. In other words, the CG method should be avoided.

Because of MATLAB's slow execution time, it is recommended that a high-level computer programming language be used to run the software. Future research can focus on different topics, such as the development of different types of controllers to be applied to the developed platform and the selection of the best control strategy for this type of manipulator. This would improve the results obtained and minimize errors between the actual arm and the simulation. As with other robots, future developments, such as path planning and computer vision, are expected for this robotic arm.

Credit Authorship Contribution Statement

9. REFERENCES

- [1] L. Tsai, "Robot analysis: the mechanics of serial and parallel manipulators," *The Mechanics of Serial and Parallel Manipulators*. p. 520, 1999. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Robot+Analysis#2%5Cnhttp://books.google.com/books?hl=en&lr=&id=PK_N9aFZ3ccC&oi=fnd&pg=PR11&dq=Robot+Analysis:+The+Mechanics+of+Serial+and+Parallel+Manipulators&ots=acRJiEF57G&sig=MXyiO77O7H09ox
- [2] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.
- [3] Z.-Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Trans. Syst. Man. Cybern.*, vol. 23, no. 5, pp. 1392–1398, 1993.
- [4] X. Wang, X. Chen, W. Jia, Y. Sun, and H. Pu, "Forward kinematics analysis and 3-dimmission gait simulation of a MiniQuad walking robot," in *2007 International Conference on Mechatronics and Automation*, 2007, pp. 1932–1937.
- [5] M. A. Qassem et al., "Efficient kinematic transformations for the PUMA 560 robot," *2010 2nd Int. Conf. Comput. Autom. Eng.*, vol. 5, no. 3, pp. 357–369, 2018.
- [6] A. C. Soh, E. A. Alwi, R. Z. A. Rahman, and L. H. Fey, "Effect of fuzzy logic controller implementation on a digitally controlled robot movement," *Kathmandu Univ. J. Sci. Eng. Technol.*, vol. 4, no. 1, pp. 28–39, 2008.
- [7] G. M. Khoury, M. Saad, H. Y. Kanaan, and C. Asmar, "Fuzzy PID control of a five DOF robot arm," *J. Intell. Robot. Syst.*, vol. 40, no. 3, pp. 299–320, 2004.
- [8] S. G. Anavatti, S. A. Salman, and J. Y. Choi, "Fuzzy + PID controller for robot manipulator," *CIMCA 2006 Int. Conf. Comput. Intell. Model. Control Autom. Jointly with IAWTIC 2006 Int. Conf. Intell. Agents Web Technol. ...*, 2006, doi: 10.1109/CIMCA.2006.103.
- [9] B. W. Bekit, L. D. Seneviratne, J. F. Whidborne, and K. Althoefer, "Fuzzy PID tuning for robot manipulators," in *IECON'98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No. 98CH36200)*, 1998, vol. 4, pp. 2452–2457.
- [10] A. Delibaşı, T. Türker, and G. Cansever, "Real-Time DC Motor Position Control by Fuzzy Logic and PID Controllers Using Labview." *Yildiz Technical University*.
- [11] A. Visioli, "Tuning of PID controllers with fuzzy logic," *IEE Proceedings-Control Theory Appl.*, vol. 148, no. 1, pp. 1–8, 2001.
- [12] D. Dulaidi, "Modeling and Control of 6 DOF Industrial Robot Using Fuzzy Logic Controller." *Universiti Tun Hussein Onn Malaysia*, 2014.
- [13] V. K. Banga, J. Kaur, R. Kumar, and Y. Singh, "Modeling and simulation of robotic arm movement using soft computing," *World Acad. Sci. Eng. Technol.*, vol. 51, pp. 616–619, 2011.
- [14] Y. Tao et al., "Fuzzy PID control method of deburring industrial robots," *J. Intell. Fuzzy Syst.*, vol. 29, no. 6, pp. 2447–2455, 2015, doi: 10.3233/IFS-151945.
- [15] J. Tavoosi, A. S. Jokandan, and M. A. Daneshwar, "A new method for position control of a 2-DOF robot arm using neuro-fuzzy controller," *Indian J. Sci. Technol.*, vol. 5, no. 3, pp. 2253–2257, 2012.

Declaration of competing Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Declaration of Funding

The authors did not receive support from any organization for the submitted work.

steady-state characteristics. All controllers could follow the set point with slight SSE. We proved that the performance of the FLC was better than the PID performance for controlling a robot manipulator in terms of reducing overshoot size, enhancing rising time, and minimizing SSEs. For example, the SSE of Motor 1 in the FLC was 0.006, while it was 0.03 in the PID controller. This means that the FLC SSE is 80% less than the PID controller. Furthermore, the rise time in the FLC is 44% less than that in the PID controller. Finally, the overshoot of FLC is 45% less than that of the PID controller. The FLC controller yielded better performance in rise time and settling and reduced steady-state error and overshoot.

- [16] U. Kabir, M. F. Hamza, A. Haruna, and G. S. Shehu, "Performance analysis of PID, PD and fuzzy controllers for position control of 3-DOF robot manipulator," arXiv Prepr. arXiv1910.12076, 2019.
- [17] A. Z. Alassar, I. M. Abuhadrous, and H. A. Elaydi, "Modeling and control of 5 DOF robot arm using supervisory control," in 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, vol. 3, pp. 351–355.
- [18] J. Tavoosi, M. Alaei, and B. Jahani, "Neuro-Fuzzy Controller for Position Control of Robot Arm," Pap. Ref., no. 0113–795, pp. 12–17, 2011.
- [19] Q. Zhu, "Real-time DC motor position control by (FPID) controllers and design (FLC) using labview software simulation," in 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010, vol. 2, pp. 417–420.
- [20] Y.-L. Kuo and S.-M. Liu, "Position Control of a Serial Manipulator Using Fuzzy-PID Controllers," Int. J. Autom. Smart Technol., vol. 5, no. 1, pp. 18–26, 2015.
- [21] N. Dersarkissian, R. Jia, and D. L. Feitosa, "Control of a Two-link Robotic Arm using Fuzzy Logic," in 2018 IEEE International Conference on Information and Automation (ICIA), 2018, pp. 481–486.
- [22] F. Z. Baghli and L. El Bakkali, "Design and simulation of robot manipulator position control system based on adaptive fuzzy PID controller," in Robotics and Mechatronics, Springer, 2016, pp. 243–250.
- [23] A. Ali, S. F. Ahmed, K. A. Kadir, M. K. Joyo, and R. N. S. Yarooq, "Fuzzy PID controller for upper limb rehabilitation robotic system," 2018 IEEE Int. Conf. Innov. Res. Dev. ICIRD 2018, no. May, pp. 1–5, 2018, doi: 10.1109/ICIRD.2018.8376291.
- [24] M. Solouki, M. Ansarin, M. Torabi, A. Nematia, and Y. Bakhshizadeh, "Optimization of PID Controller with Supervisory Fuzzy Control for Industrial Robots," J. Artif. Intell. Electr. Eng., vol. 7, no. 28, pp. 27–36, 2019.
- [25] F. Cao, X. Wang, and X. Li, "Fuzzy adaptive PID control of a mobile assistive robot platform," in 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, 2014, pp. 502–506.
- [26] L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta," ISA Trans., vol. 79, pp. 172–188, 2018.
- [27] S. Pedrammehr, B. Danaei, H. Abdi, M. T. Masouleh, and S. Nahavandi, "Dynamic analysis of Hexarot: axis-symmetric parallel manipulator," Robotica, vol. 36, no. 2, pp. 225–240, 2018.
- [28] D. Jinjun, G. Yahui, C. Ming, and D. Xianzhong, "Symmetrical adaptive variable admittance control for position/force tracking of dual-arm cooperative manipulators with unknown trajectory deviations," Robot. Comput. Integr. Manuf., vol. 57, pp. 357–369, 2019.
- [29] N. M. Ghaleb and A. A. Aly, "Modeling and control of 2-DOF robot arm," Int. J. Emerg. Eng. Res. Technol., vol. 6, no. 11, pp. 24–31, 2018.
- [30] N. Dersarkissian, R. Jia, and D. L. Feitosa, "Control of a Two-link Robotic Arm using Fuzzy Logic," in 2018 IEEE International Conference on Information and Automation (ICIA), 2018, pp. 481–486.
- [31] U. Kabir, M. F. Hamza, A. Haruna, and G. S. Shehu, "Performance analysis of PID, PD and fuzzy controllers for position control of 3-DOF robot manipulator," arXiv, vol. 8, no. 1, pp. 18–25, 2019.
- [32] M. F. El-Khatib and S. A. Maged, "Low level position control for 4-DOF arm robot using fuzzy logic controller and 2-DOF PID controller," 2021 Int. Mobile, Intelligent, Ubiquitous Comput. Conf. MIUCC 2021, pp. 258–262, 2021, doi: 10.1109/MIUCC52538.2021.9447617.
- [33] A.-A. S. Abdel-Salam and I. N. Jleta, "Fuzzy logic controller design for PUMA 560 robot manipulator," IAES Int. J. Robot. Autom., vol. 9, no. 2, p. 73, 2020, doi: 10.11591/ijra.v9i2.pp73-83.
- [34] P. Rocco, "Stability of PID control for industrial robot arms," IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 606–614, 1996.
- [35] Z.-Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," IEEE Trans. Syst. Man. Cybern., vol. 23, no. 5, pp. 1392–1398, 1993.
- [36] C. R. Carignan, G. G. Gefke, and B. J. Roberts, "Intro to space mission design: space robotics," in Seminar of Space Robotics, University of Maryland, Baltimore, 2002, vol. 26.
- [37] R. Boyce and J. Mull, "Complying with the occupational safety and health administration: Guidelines for the dental office," Dent. Clin. North Am., vol. 52, no. 3, pp. 653–668, 2008.
- [38] A. Rahman, A. H. Khan, T. Ahmed, and M. Sajjad, "Design, Analysis and Implementation of a Robotic Arm- The Animator," no. 10, pp. 298–307, 2013.
- [39] A. Elfasakhany, E. Yanez, K. Baylon, and R. Salgado, "Design and Development of a Competitive Low-Cost Robot Arm with Four Degrees of Freedom," Mod. Mech. Eng., vol. 01, no. 02, pp. 47–55, 2011, doi: 10.4236/mme.2011.12007.
- [40] S. Yu, X. Yu, B. Shirinzadeh, and Z. Man, "Continuous finite-time control for robotic manipulators with terminal sliding mode," Automatica, vol. 41, no. 11, pp. 1957–1964, 2005.

- [41] P. Singh, A. Kumar, and M. Vashisth, "Design of a robotic arm with gripper & end effector for spot welding," *Univers. J. Mech. Eng.*, vol. 1, no. 3, pp. 92–97, 2013.
- [42] M. W. Spong and M. Vidyasagar, *Robot dynamics and control*. John Wiley & Sons, 2008.
- [43] The MathWorks Inc. MATLAB 7.0 (R14SP2). The MathWorks Inc., 2005.
- [44] J. J. Craig, *Introduction to robotics: mechanics and control*. Pearson Educacion, 2005.
- [45] "Corke2017_Book_RoboticsVisionAndControl.pdf.crdownload."
- [46] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*, vol. 3. Wiley New York, 2006.
- [47] D. H. Kim, "Robust tuning of embedded intelligent PID controller for induction motor using bacterial foraging based optimization," in *International Conference on Embedded Software and Systems*, 2004, pp. 137–142.
- [48] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, 2006.
- [49] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404–418, 1990, doi: 10.1109/21.525