**Journal of Intelligent Systems and Control**
https://www.acadlore.com/journals/JISC

# Comparative Analysis of PID and Fuzzy Logic Controllers for Position Control in Double-Link Robotic Manipulators

Nor Maniha Abdul Ghani*, Aqib Othman, Azrul Azim Abdullah Hashim, Ahmas Nor Kasrudin Nasir

Faculty of Electrical & Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah, 26600 Pekan, Pahang, Malaysia

* Correspondence: Nor Maniha Abdul Ghani (normaniha@umpsa.edu.my)

**Received:** 10-01-2023 **Revised:** 11-02-2023 **Accepted:** 11-13-2023

**Citation:** N. M. A. Ghani, A. Othman, A. A. A. Hashim, and A. N. K. Nasir, "Comparative analysis of PID and fuzzy logic controllers for position control in double-link robotic manipulators," *J. Intell Syst. Control*, vol. 2, no. 4, pp. 183–196, 2023. https://doi.org/10.56578/jisc020401.

© 2023 by the authors. Licensee Acadlore Publishing Services Limited, Hong Kong. This article can be downloaded for free, and reused and quoted with a citation of the original published version, under the CC BY 4.0 license.

**Abstract:** This study presents a comprehensive evaluation of linear and non-linear control systems, specifically Proportion Integration Differentiation (PID) and fuzzy logic controllers, in the context of position control within double-link robotic manipulators. The effectiveness of these controllers was rigorously assessed in a simulated environment, utilizing MATLAB Simulink for the simulation and SOLIDWORKS for the model design. The PID controller, characterized by its *Kp*, *Ki*, and *Kd* components, was implemented both in the simulation and on the hardware. However, due to the constraints of the microcontroller's RAM and processor, which facilitate the hardware's connection with MATLAB, the application of the Fuzzy Logic concept to hardware was not feasible. In the simulated environment, the fuzzy logic controller demonstrated superior stability in comparison to the PID controller, evidenced by a lower settling time (1.0 seconds) and overshoot (2%). In contrast, the PID controller exhibited a settling time of 0.2 seconds and an overshoot of 32%. Additionally, the fuzzy logic controller showcased a 44% reduction in steady-state error relative to the PID controller. When applied to hardware, the PID controller maintained stable results, achieving a settling time of 0.6 seconds and an overshoot of 2%. The steady-state errors for Link 1 and Link 2 were recorded as 3.6° and 1.4°, respectively. The findings highlight the fuzzy logic controller's enhanced stability, rendering it more suitable for ensuring the accuracy and protection of the manipulator system. As a non-linear controller, the fuzzy logic controller efficiently addresses various potential errors through its intelligent control mechanism, which is embedded in its fuzzy rules. Conversely, the PID controller, a linear controller, responds rapidly but may lack flexibility in complex scenarios due to its inherent linearity. This study underscores the importance of selecting an appropriate controller based on the specific requirements of robotic manipulator systems, with a focus on achieving optimal performance and stability.

**Keywords:** Proportion integration differentiation controller; Fuzzy logic controller; MATLAB Simulink; Double link robotic manipulator system; Hardware in loop application

## 1 Introduction

Robotic manipulator systems, known for their structural flexibility and capacity for versatile movement, require precise control mechanisms to ensure accurate trajectory and velocity [1, 2]. In these systems, two predominant types of controllers are employed: linear and non-linear. PID controllers, proposed as linear controllers, are favored for their simplicity, stability, and adaptability in systems with linear dynamics [3]. In contrast, fuzzy logic controllers, representing non-linear control systems, excel in handling complex, unpredictable scenarios and integrate human expertise effectively, making them suitable for non-linear system environments [4].

Historically, the range of motion in most robotic manipulators was constrained by the capabilities of actuators and the control algorithms employed. Such limitations often prevented robots from reaching certain positions or executing specific tasks. Mechanical inaccuracies in the robot's design further compounded these challenges, impacting overall performance [5, 6]. Additionally, environmental disturbances such as vibrations or wind have been observed to induce unpredictable movements in robotic manipulators, potentially causing deviations from intended positions.

In response to these challenges, this study evaluates the adaptability and efficiency of both linear and non-linear controllers in maintaining the precision of robotic manipulators under various disturbances. The influence of both internal and external disturbances, including human interaction and inertial effects on the movement of links, has been a critical focus. The rate at which each controller type corrects disturbances-induced errors is a key metric for assessing their efficiency.

To mitigate disturbances in robotic manipulators, a simulated environment was developed, utilizing PID and fuzzy logic controllers within a Double Link Robotic Manipulator System (DLRMS). This simulation, conducted using Simscape in MATLAB Simulink, aimed to scrutinize the effectiveness of these control systems against diverse disturbances applied to DLRMS. Further, the optimal structure of the PID controller was corroborated through Hardware in Loop (HIL) testing.

This comprehensive examination of PID and fuzzy logic controllers seeks to elucidate their respective strengths and limitations in managing the complexities inherent in robotic manipulator systems. The study's findings are intended to contribute to the advancement of control system design, enhancing the precision and adaptability of robotic manipulators in dynamic environments.

## 2 Methodology

### 2.1 System Flowchart

The methodology encompasses three distinct approaches: simulation studies of both PID and fuzzy logic controllers, and hardware implementation utilizing only the PID controller within a HIL system, as delineated in the flowchart (Figure 1).

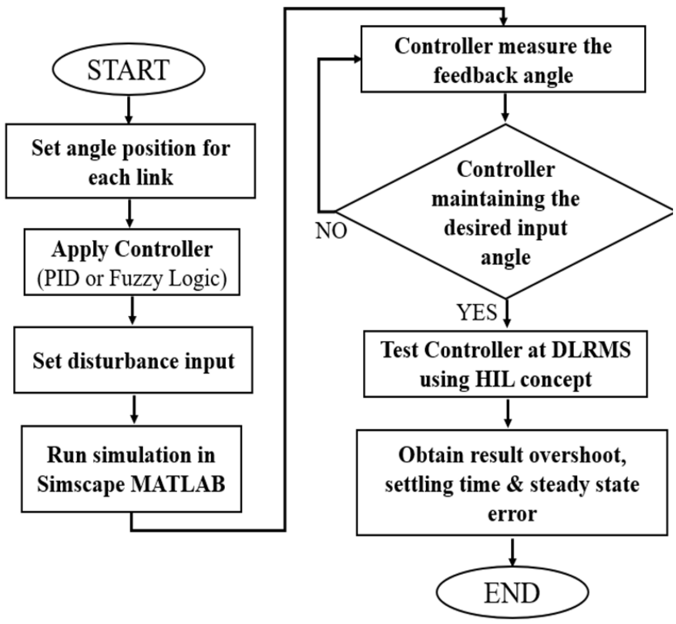


**Figure 1.** Flowchart of simulation and hardware implementation procedure

### 2.2 Linear Controller (PID Controller)

In robotic systems, PID controllers play a crucial role in regulating arm position and mitigating disturbances. These controllers adjust the output by quantifying the discrepancy between the desired and actual positions of the robotic arm, employing proportional control (P) [7]. Integral control (I) is utilized to counteract steady-state errors, while derivative control (D) aims to curtail overshoot and oscillation [8]. The efficiency of the PID controller in double-link robotic manipulator systems is evaluated through two methodologies: simulation using 3D animation from Simscape (Figure 2) and hardware implementation via a HIL concept. Both approaches focus on reducing steady-state errors and enhancing the flexibility of the manipulator link system [8, 9].

The PID values for each link are distinct, reflecting the unique roles of the links. Consequently, different *Kp*, *Ki*, and *Kd* values are assigned to each link, as outlined in Table 1. The *Kp* value for Link 2 is higher to accommodate the additional inertia encountered during the movement of Link 1, necessitating a more robust evaluation of the angle discrepancy. These values were derived from system identification references, using motor specifications akin to those employed in this study. A minor modification in the *Kd* value was implemented to optimize error reduction.

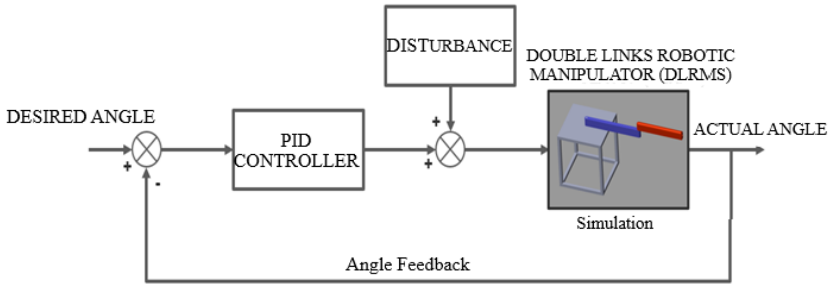


**Figure 2.** PID controller of DLRMS

**Table 1.** PID value of simulation and hardware for Link 1 and Link 2

| Parameter | Simulation | | Hardware | |
|---|---|---|---|---|
| | **Link 1** | **Link 2** | **Link 1** | **Link 2** |
| $Kp$ | 70 | 90 | 70 | 90 |
| $Ki$ | 0.005 | 0.005 | 0.005 | 0.005 |
| $Kd$ | 2.5 | 2.5 | 0.5 | 0.5 |

The *Kp* and *Ki* values remained constant across both methodologies, while a slight variation in *Kd* was deemed necessary. This adjustment was intended to enhance the reduction of overshoot and settling time in hardware applications.

### 2.3 Non-Linear Controller (Fuzzy Logic Controller)

The fuzzy logic controller, a non-linear control system, demonstrates versatility and efficacy in managing complex systems. Inputs for this controller are typically derived from sensor readings or other system variables, and the outputs manifest as control signals or actions within the system [10, 11]. Each rule and variable within the fuzzy logic system is designed to counter disturbances affecting the performance of the robotic manipulator's links. The fuzzy logic block diagram and its implementation in MATLAB Simulink using the Simscape plugin are illustrated in Figure 3, respectively.

Prior to the implementation of the fuzzy logic controller, the linguistic variables and terms relevant to the system are defined. This step is critical for the construction of an appropriate membership function. The knowledge base, comprising these defined rules, undergoes a process of defuzzification to maintain the position of the links and to mitigate any disturbances, thereby ensuring the attainment of the desired angle [12].

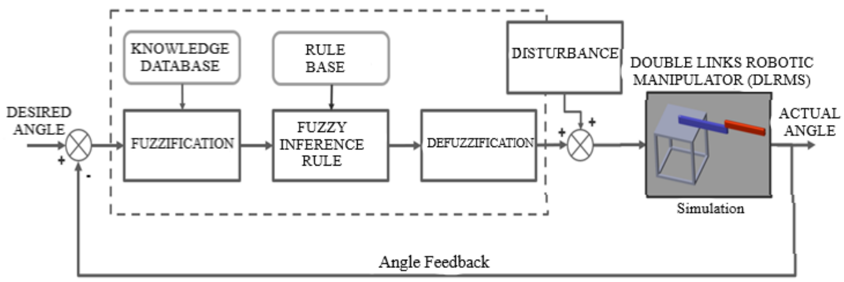


**Figure 3.** Fuzzy logic control block diagram

Table 2 outlines the fuzzy rules for the Mamdani fuzzy inference system. These rules are categorized based on the input error (e) and the change in error ($\Delta$e), each varying from negative big (NB), negative small (NS), zero (Z), positive small (PS), to positive big (PB). The output of the fuzzy logic controller is then formulated to counterbalance these errors, aiming for a zero error rate [13]. The implementation of Gaussian membership functions within the controller enhances its flexibility, while the Mamdani system in the fuzzy inference system is designed to handle errors ranging from small to large without inconsistencies. Figure 4 displays the fuzzy surface utilized in this research, representing the 5x5 Mamdani membership rules.

**Table 2.** Fuzzy rule of Mamdani fuzzy inference system

| $e/\Delta e$ | $NB$ | $NS$ | $Z$ | $PS$ | $PB$ |
|---|---|---|---|---|---|
| $NB$ | $PB$ | $PB$ | $PB$ | $PS$ | $Z$ |
| $NS$ | $PB$ | $PB$ | $PS$ | $Z$ | $NS$ |
| $Z$ | $PB$ | $PS$ | $Z$ | $NS$ | $NB$ |
| $PS$ | $PS$ | $Z$ | $NS$ | $NB$ | $NB$ |
| $PB$ | $Z$ | $NS$ | $NB$ | $NB$ | $NB$ |

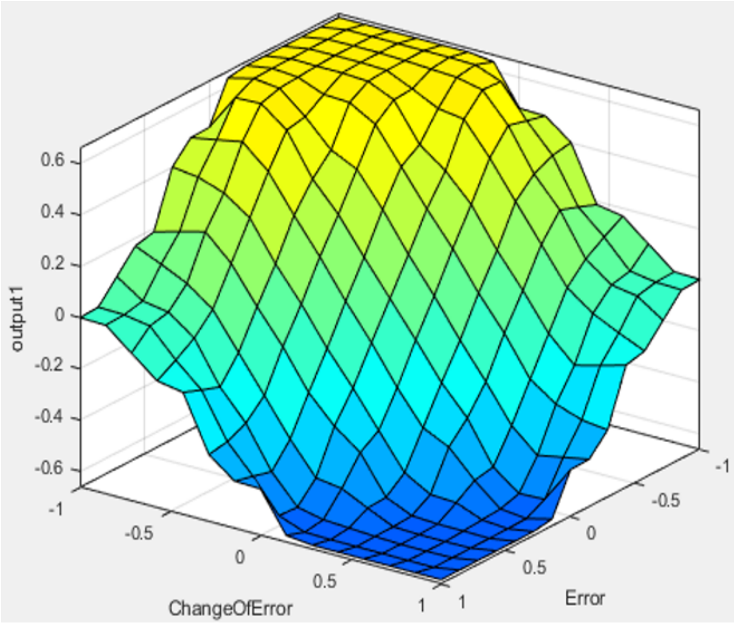


**Figure 4.** Fuzzy surface of 5×5 fuzzy Mamdani membership rules

### 2.4 Simulation Setup

The simulation was structured to evaluate the controllers' response under four distinct conditions, with the input angles set at 45°, 90°, 180° and 270° for both Link 1 and Link 2. These conditions are visualized in Figure 5 and Figure 6. Additionally, a movement sequence akin to a "pick & place robot" application was also simulated to further test the controllers' efficacy in a dynamic environment. The associated figures (Figure 5 and Figure 6) illustrate the signal builders for the input signals of Links 1 and 2, respectively.

Internal disturbances were applied to each link, utilizing different step function signals as depicted in Figure 7. This approach enabled an analysis of the controllers' ability to handle internal system disruptions. Moreover, Figure 8 displays the initial position of both Link 1 and Link 2, set at a 0-degree angle.

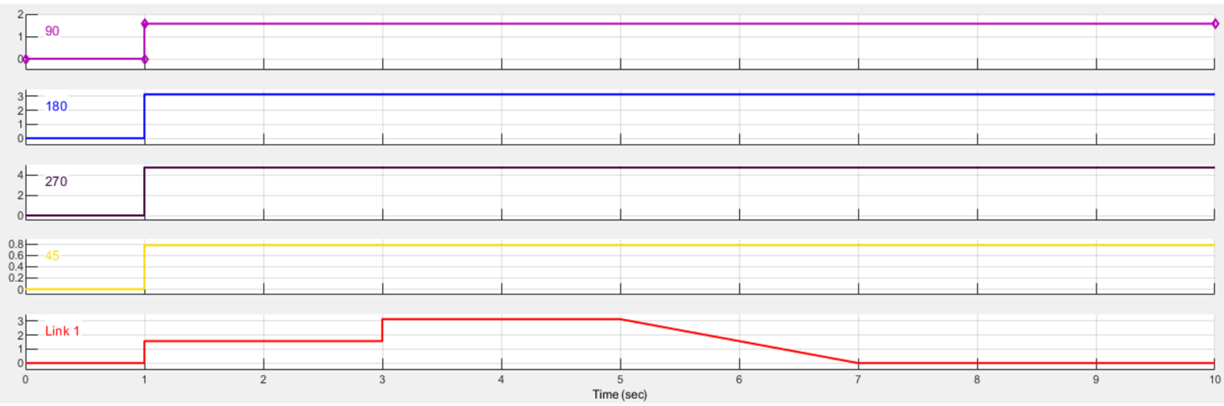


**Figure 5.** Signal builder of input signal for Link 1

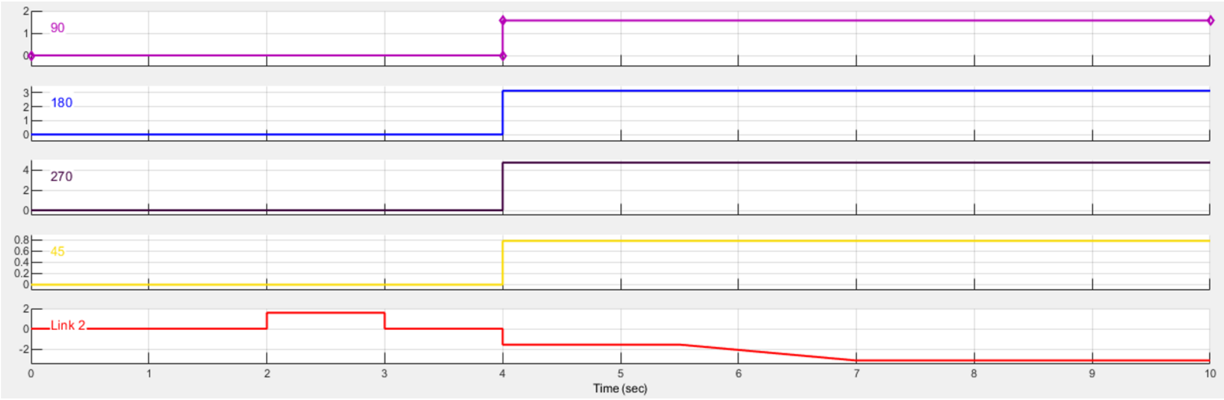


**Figure 6.** Signal builder of input signal for Link 2

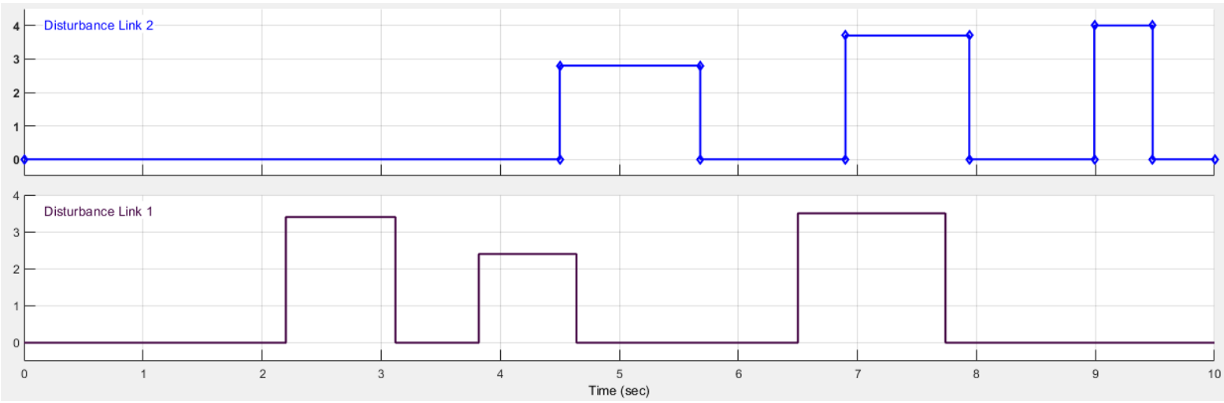


**Figure 7.** Signal builder disturbance signal for Link 1 and Link 2

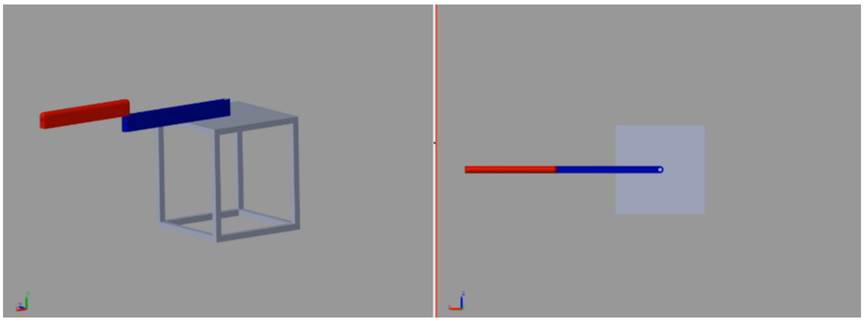

**Figure 8.** Initial condition of double link at 0 degree

The step function signal generated by the signal builder, with a maximum value reaching 250°, was injected into the system at predetermined intervals. The primary objective of introducing this disturbance signal was to assess the controllers' proficiency in minimizing the steady-state error induced by such disturbances.

### 2.5 HIL

The HIL system integration is depicted in Figure 9, showcasing the connection diagram between MATLAB Simulink and the hardware system. The hardware utilized in this study is powered by two 12V DC motors equipped with encoders, entirely controlled via MATLAB Simulink. An Arduino Mega functions as an intermediary between MATLAB Simulink and the DC motor encoders. This choice was informed by the compatibility of the Arduino Mega with Simulink's external plugins, offering seamless integration with the 12V Brushless DC motors and their encoders. Specifically, interrupt pins 2, 3, 18, and 19 on the Arduino Mega facilitate this connection, aligning with the requirements of the hardware setup. Figure 10 and Figure 11 illustrate the double link robotic manipulator hardware and its integration with MATLAB Simulink, respectively.

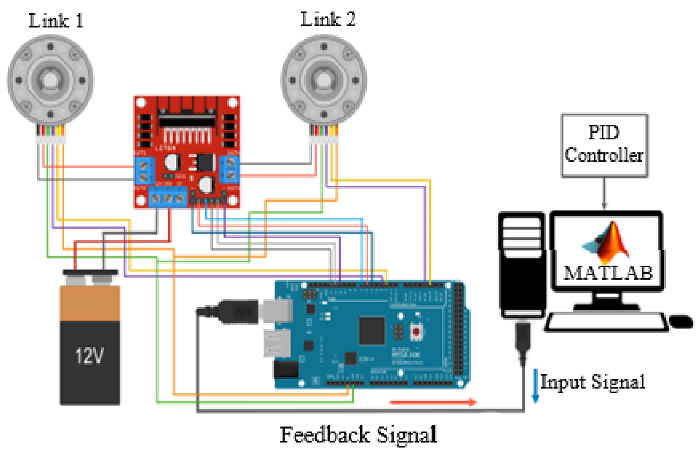


**Figure 9.** Connection diagram for HIL between MATLAB Simulink and hardware system

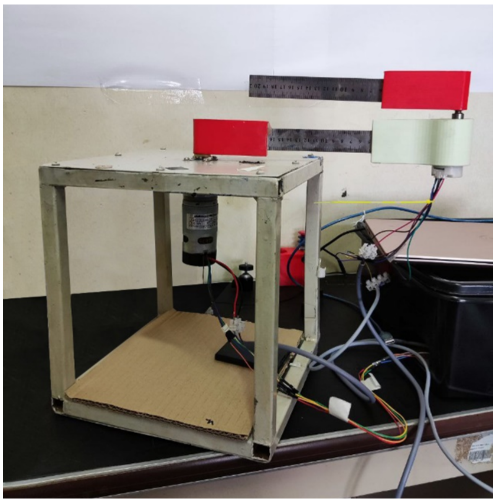

**Figure 10.** Double link robotic manipulator hardware

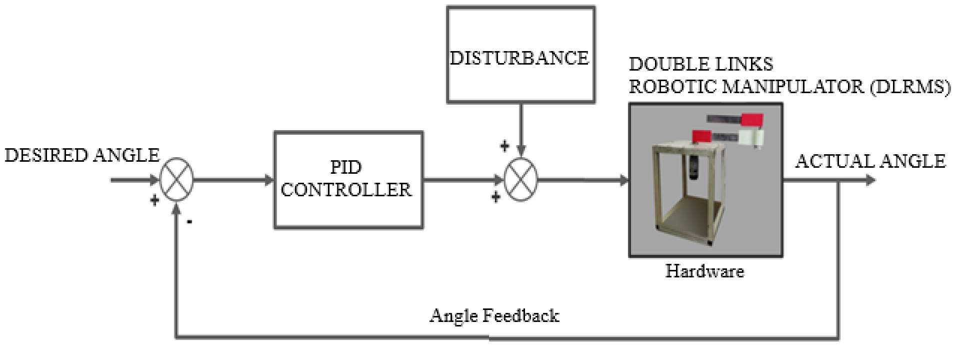


**Figure 11.** PID controller of double link robotic manipulator integrated with MATLAB Simulink

The PID values for the hardware implementation of the double-link robotic manipulator slightly diverge from those in the simulation. To mitigate overshoot within the system, the *Ki* value in the hardware setting was reduced compared to that in the simulation [14]. The objective was to replicate the control behavior observed in the simulation, maintaining consistency between the PID controller's simulated and actual performance. However, it was observed that a minor adjustment in the *Kd* value was necessary to enhance the system's response, ensuring optimal control efficacy [9, 15].

This part of the methodology emphasizes the critical importance of adapting control parameters when transitioning from simulation to hardware implementation. The hardware setup, especially the integration of the Arduino Mega, exemplifies the need for flexibility and compatibility in hardware choices to ensure the effective application of control strategies developed in simulated environments.

## 3 Results and Discussion

### 3.1 Simulation of the PID Controller

In the conducted simulation, a set point of 180° was established for both links, which encountered signal disturbances at specific intervals. It was observed that the PID controller effectively managed to maintain the position of the links while concurrently reducing internal disturbances and minimizing steady-state errors.

As illustrated in Figure 12, a notable overshoot was initially observed at the commencement of the step function (angle of 180°) for both links. A noteworthy phenomenon was the slight negative movement of Link 1, induced by the inertia effect on Link 2. However, the PID controller promptly rectified this disturbance, restoring Link 1 to its original position expeditiously. Both links exhibited rapid settling times and negligible steady-state errors, demonstrating the PID controller's efficiency in position control and disturbance reduction.

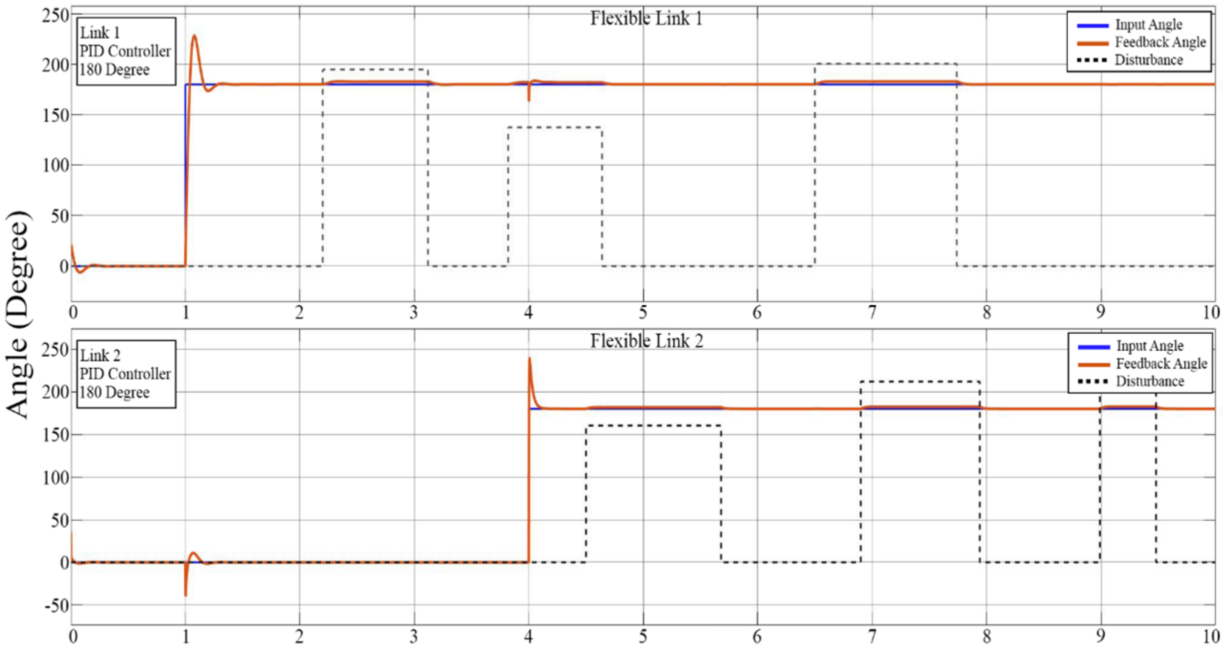


**Figure 12.** Performance of double link robotic manipulator using PID controller

### 3.2 Simulation of the Fuzzy Logic Controller

In this simulation scenario, similar to the PID setup, both links were subjected to signal disturbances at a predetermined time and operated with a set point of 180°. The fuzzy logic controller was then evaluated for its capability to sustain link positions while minimizing steady-state errors and internal disturbances.

Referencing Figure 13, it was discerned that the settling time for the system to attain the desired angle was elongated compared to the PID controller's instantaneous response. Notably, the settling time with the fuzzy logic controller varied proportionally with the set point angle; higher angles resulted in extended settling times. Minimal overshoot was observed in the movements of both links, attributed to the extended settling period. Furthermore, when internal disturbances were introduced, the steady-state error remained minimal.

### 3.3 Hardware Implementation of the PID Controller

Due to limitations in applying the fuzzy logic controller to actual hardware, the study focused on the PID controller for the hardware implementation. The interaction between the PID controller and the DLRMS was meticulously analyzed. Similar to the simulation, a set point angle of 180° was employed, along with the highest input disturbance value.

The results, as demonstrated in Figure 14, revealed that the PID controller facilitated a highly responsive settling time on both links, accompanied by minimal overshoot. The steady-state error, when disturbances were introduced, was found to be significantly low. The reduced overshoot was influenced by the PID controller's prompt response to the set point angle. Additionally, the utilization of geared DC motors and efficient encoders synergistically contributed to the enhanced control of the link positions [16, 17].

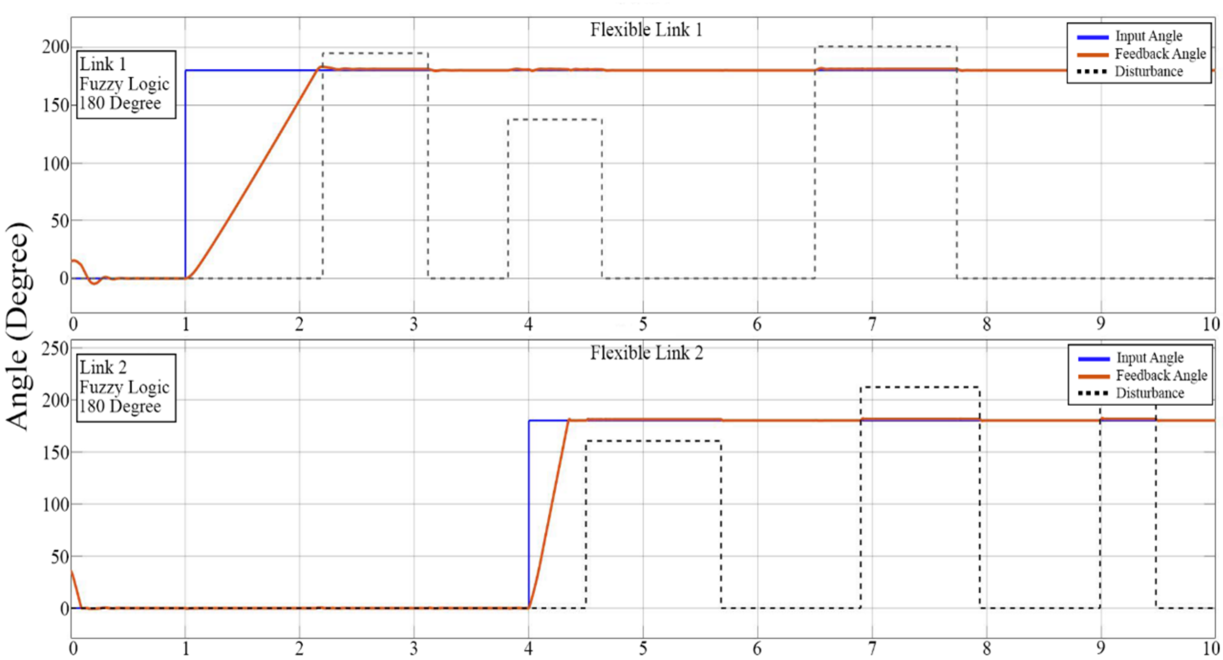


**Figure 13.** Performance of double link robotic manipulator using fuzzy logic controller

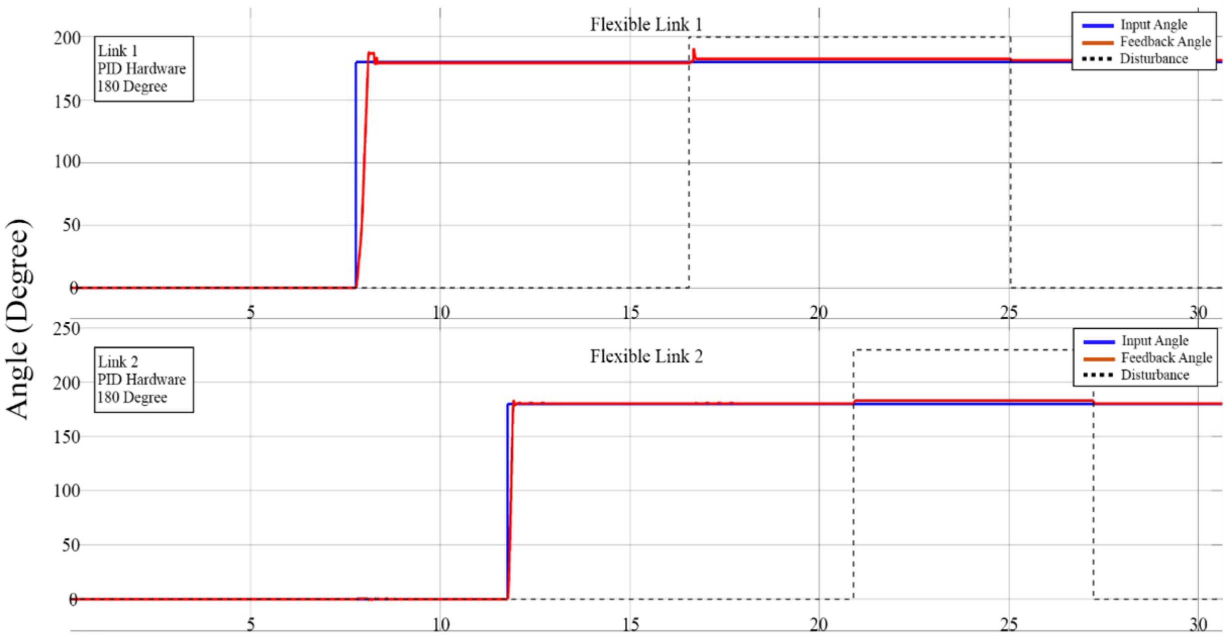


**Figure 14.** Hardware results of double link robotic manipulator using PID controller

### 3.4 Robot Pick and Place Routine Demonstration Result

DLRMS, analogous to robotic arms used in industrial applications such as robotic welding and medical scanning, was employed to demonstrate a pick & place routine. This routine, simulating the movement of a ball from Box A to Box B and back to a resting position, was executed using both PID and fuzzy logic controllers. The performance of these controllers during this routine is depicted in Figures 15-17.

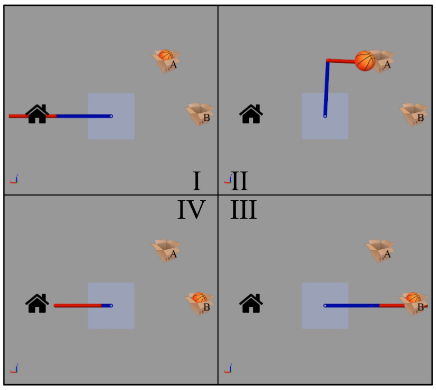


**Figure 15.** Pick & place robot demonstration via DLRMS

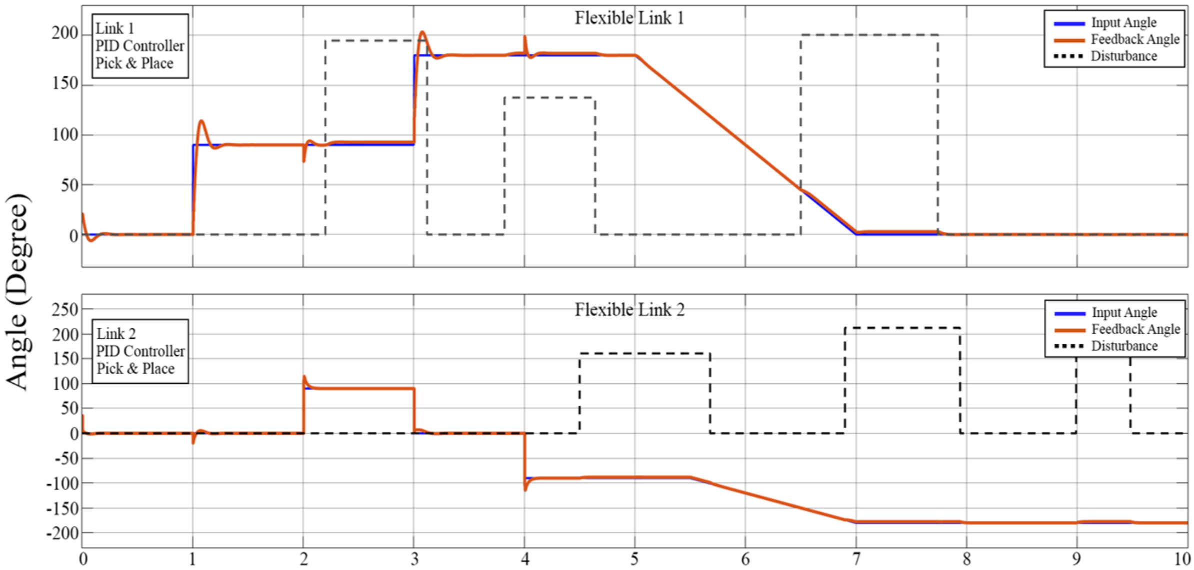


**Figure 16.** Position control of PID controller for pick & place robot demonstration

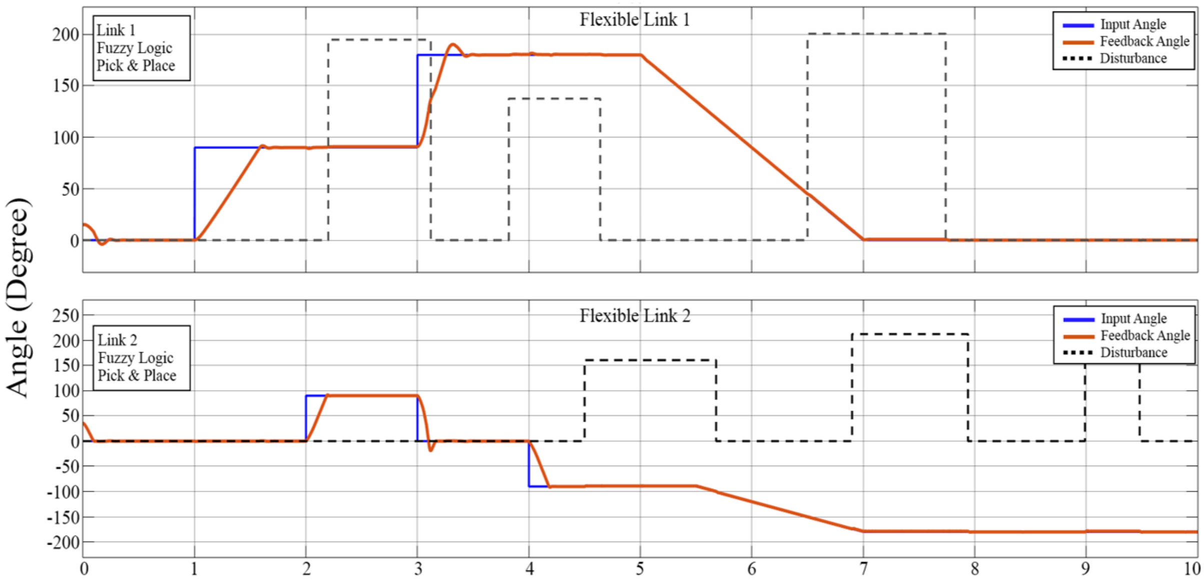


**Figure 17.** Position control of fuzzy logic controller for pick & place robot demonstration

Observations from Figures 16 and 17 reveal that the PID controller exhibits rapid settling time and minimal steady-state error, yet it is prone to significant overshoot due to its quick response. Conversely, the fuzzy logic controller, while demonstrating slower settling times for each set point, ensures minimal overshoot and a steady-state error approaching zero. These findings suggest that both controllers possess distinct advantages and are capable of performing effectively within the given parameters.

A notable limitation of the PID controller is its susceptibility to unexpected overshoots when faced with out-of-range signals, potentially leading to system failure. The fuzzy logic controller, in contrast, smoothly manages various inputs and deliberately extends settling time to reduce abrupt movements, thereby minimizing overshoot during the rise time. This extended settling time demonstrates a commitment to maintaining position accuracy, even in the face of external disturbances.

It was further observed that the fuzzy logic controller more effectively repels external disturbances compared to the PID controller, exhibiting a higher error reduction rate. This controller's ability to minimize inertia effects on link positions contributes to its robustness, particularly in scenarios involving free-form movements of manipulator links. Such attributes render the fuzzy logic controller more adaptable and resilient to unexpected system failures, highlighting its suitability for dynamic and complex tasks.

### 3.5 Combined Results

The comprehensive results encompassing the simulation method and hardware implementation are consolidated in a unified table. This table presents a comparative analysis of overshoot, steady-state error, and settling time for the PID and fuzzy logic controllers in simulation, as well as the PID controller in hardware.

#### 3.5.1 Overshoot

Figure 18 and Table 3 reveal a pronounced disparity in overshoot values among the different control methods. The Simulink PID exhibited overshoot values ranging from 11% to 32%. In stark contrast, both the Fuzzy Logic Simulink and Hardware PID controllers demonstrated significantly lower overshoot values, ranging from 0.3% to 7.5%. These results indicate the superior performance of the fuzzy logic controller in Simulink and the PID controller in hardware environments in terms of minimizing overshoot.

**Table 3.** Comparative overshoot results for three controller methods

| | **Overshoot, %** | | | | | |
|---|---|---|---|---|---|---|
| **Set Point Angle** | **PID Controller** | | **Fuzzy Logic** | | **Hardware PID** | |
| | Link 1 | Link 2 | Link 1 | Link 2 | Link 1 | Link 2 |
| 45° | 27.47 | 11.33 | 5.91 | 1.24 | 3.65 | 7.41 |
| 90° | 27.56 | 23.91 | 3.65 | 1.36 | 3.65 | 4.74 |
| 180° | 26.62 | 31.54 | 1.53 | 0.36 | 1.53 | 1.53 |
| 270° | 26.28 | 25.74 | 1.53 | 0.51 | 0.51 | 0.51 |

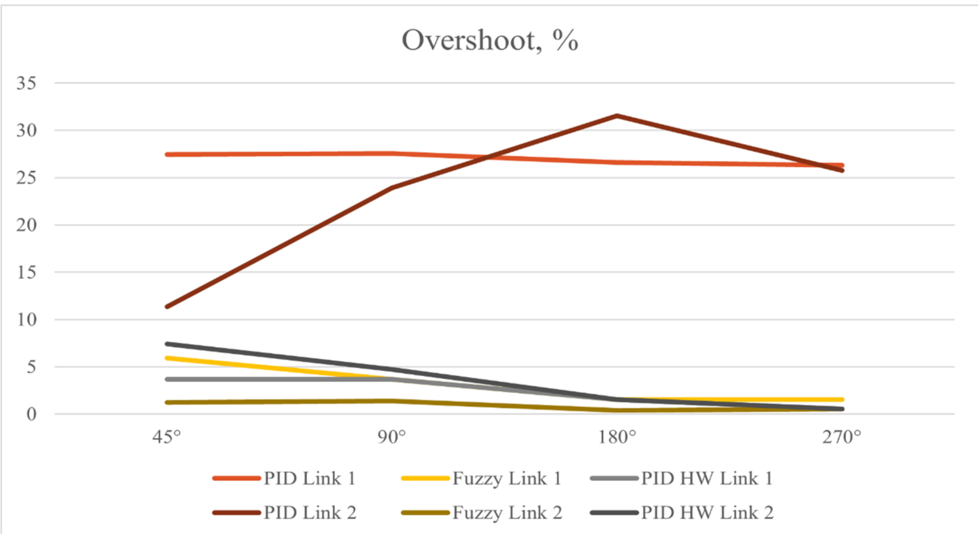


**Figure 18.** Graphical representation of overshoot for three controller methods

### 3.6 Steady-State Error

As depicted in Figure 19 and summarized in Table 4, all three methods achieved remarkably minimal steady-state errors, confined within the range of 0° to 5°. This outcome underscores the efficiency of both the PID and fuzzy logic controllers in mitigating internal disturbances within the system. Despite disturbances, these controllers were adept at maintaining the set point angle with negligible steady-state errors.

The fuzzy logic controller demonstrated the lowest steady-state error, showcasing its flexibility in handling disturbances related to angle and inertia. The PID controller, both in simulation and hardware settings, exhibited a slightly higher steady-state error compared to the fuzzy logic controller.

**Table 4.** Steady-state error results for three controller methods

| | Steady State Error, ° | | | | | |
|---|---|---|---|---|---|---|
| **Set Point Angle** | **PID Controller** | | **Fuzzy Logic** | | **Hardware PID** | |
| | Link 1 | Link 2 | Link 1 | Link 2 | Link 1 | Link 2 |
| 45° | 4.7 | 2.6 | 1.6 | 1.4 | 0.6 | 3.6 |
| 90° | 3.2 | 2.6 | 1.0 | 1.5 | 1.6 | 2.9 |
| 180° | 3.0 | 2.0 | 0.7 | 1.5 | 1.4 | 3.6 |
| 270° | 3.2 | 2.5 | 1.2 | 1.5 | 0.8 | 3.0 |

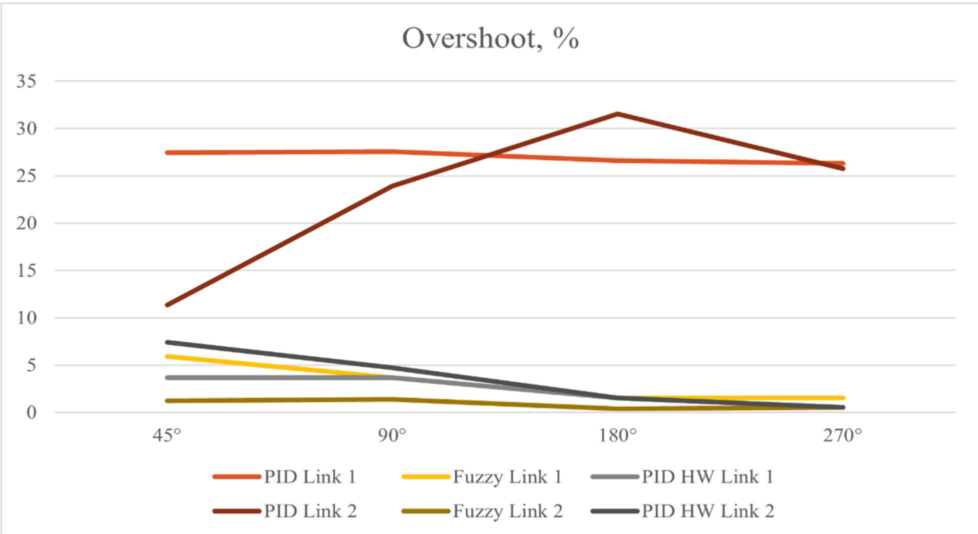


**Figure 19.** Graph depicting steady-state error across three methods

#### 3.6.1 Settling time

Figure 20 and Table 5 present the settling time comparisons among the three controller methods. A significant difference was noted, particularly between the fuzzy logic controller and the hardware PID, in comparison to the simulation PID. The maximum settling time recorded was 1.5 seconds, a value considered exemplary for controller efficiency. The simulation PID exhibited the shortest settling time, attributed to its rapid response in controlling the 3D Simscape model, which is highly sensitive to movements. The fuzzy logic controller, characterized by its extensive membership range, adeptly manages both standard and atypical disturbances through its specified rules, showcasing responsiveness across various disturbance scenarios.

#### 3.6.2 Outcome results

The results elucidate the distinct advantages and limitations of both the PID and fuzzy logic controllers in robotic system control. The PID controller, with its simplicity, stability, and adaptability, is particularly effective in linear systems. In contrast, the fuzzy logic controller, known for its versatility and robustness in managing unpredictable scenarios, excels in complex and nonlinear systems. The choice of controller is contingent upon factors such as system complexity, nonlinearity, and availability of specialized expertise. An optimal approach may involve leveraging the strengths of both controllers in tandem, depending on the specific requirements of the task at hand.

**Table 5.** Settling time results for three controller methods

| Set Point Angle | Settling Time, s | | | | | |
|---|---|---|---|---|---|---|
| | PID Controller | | Fuzzy Logic | | Hardware PID | |
| | Link 1 | Link 2 | Link 1 | Link 2 | Link 1 | Link 2 |
| 45° | 0.3 | 0.1 | 0.4 | 0.2 | 0.4 | 0.2 |
| 90° | 0.3 | 0.1 | 0.6 | 0.3 | 0.6 | 0.3 |
| 180° | 0.4 | 0.2 | 1.0 | 0.4 | 0.6 | 0.4 |
| 270° | 0.4 | 0.2 | 1.5 | 0.6 | 1.11 | 0.5 |

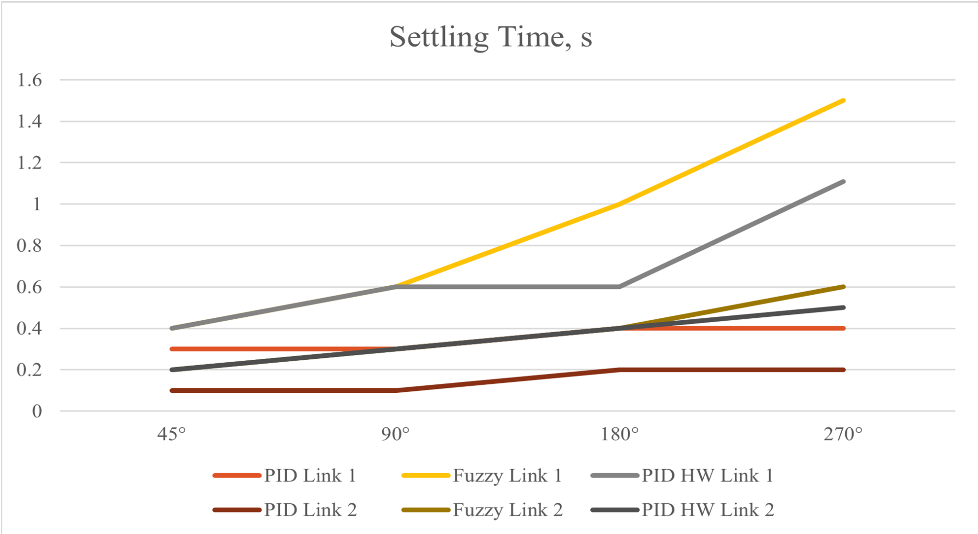


**Figure 20.** Graph of settling time for three controller methods

## 4 Conclusions

In summary, the study demonstrated that both PID and fuzzy logic controllers are compatible with the DLRMS, effectively maintaining manipulator link positions and mitigating disturbances. The PID controller, a linear controller, exhibits high reactivity but faces limitations in handling unforeseen situations and is prone to overshooting [18, 19]. Its lack of adaptability in responding to sudden adjustments potentially results in system failures.

Conversely, the fuzzy logic controller excels in precision maintenance of manipulator link positions and disturbance reduction. Its ability to adapt system parameters based on a relevant knowledge base enables it to effectively respond to unexpected routines or actions [20]. This flexibility is instrumental in minimizing errors and maintaining system consistency, thus averting potential failures.

Moreover, the PID and fuzzy logic controllers complement each other within the DLRMS, enhancing responsiveness and efficiency in achieving specific objectives swiftly [21]. While the PID controller is suited for simpler systems, the fuzzy logic controller is more appropriate for complex environments.

The application of these systems in the industrial sector, particularly in heavy industries like automotive manufacturing and sorting centers, is notable. Tasks such as ”pick & place robot” operations and spot-welding are common applications. The integration of a non-linear controller and intelligent control within these systems can lead to more efficient operations, benefiting the industry significantly.

### 4.1 Recommendation

#### 4.1.1 Simulation improvement

The current simulation serves as an analog for the hardware, yet it does not reflect specific aspects like DC motor encoder or actuator specifications. Presently, the 3D model settings are limited to the total weight of each link. It is recommended to incorporate the transfer function of each DC motor encoder or actuator using system identification. This would optimize the simulation, aligning it more closely with the hardware application. Post-implementation of hardware parameters into the 3D model, simulation results are expected to align more accurately with hardware outcomes. Additionally, further analysis on link vibrations could provide insights into their impact on the controllers' ability to maintain positions.

#### 4.1.2 Hardware improvement

To enhance the implementation of the fuzzy logic controller with the Arduino Mega interface, it is advised to upload the fuzzy inference system file onto the Arduino Mega board. This will facilitate smoother processing of fuzzy rules. The current limitation is that the MATLAB Simulink-managed fuzzy logic system is not fully processed by the Arduino Mega's memory system, due to the large size of fuzzy set data. Uploading the 'fis' file to the Arduino Mega would potentially enable more effective processing of the fuzzy set, improving signal transmission to the encoder.

## Funding

The author would like to thank the Ministry of Higher Education for providing financial support under Fundamental Research Grant Scheme (Grant No.: FRGS/1/2021/ICT02/UMP/02/2) with a university reference number (RDU210110).

## Data Availability

The data used to support the research findings are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

[1] W. Chen, Y. Yu, X. Zhao, L. Zhao, and Q. Sun, "Position control of a 2DOF underactuated planar flexible manipulator," in *2011 IEEE International Conference on Mechatronics and Automation, Beijing, China*, 2011, pp. 464–469. https://doi.org/10.1109/ICMA.2011.5985702

[2] U. K. Sahu, D. Patra, and B. Subudhi, "Network based control of 2-DOF serial flexible link manipulator," in *TENCON 2017-2017 IEEE Region 10 Conference, Penang, Malaysia*, 2017, pp. 333–338. https://doi.org/10.1109/TENCON.2017.8227885

[3] R. K. Mandava and P. R. Vundavalli, "Design of pid controllers for 4-DOF planar and spatial manipulators," in *2015 International Conference on Robotics, Automation, Control and Embedded Systems (RACE), Chennai, India*, 2015, pp. 1–6. https://doi.org/10.1109/RACE.2015.7097269

[4] L. Kong, W. He, Y. Dong, C. Yang, and G. Li, "Bounded control of robotic manipulators via fuzzy logic systems," in *2017 36th Chinese Control Conference (CCC), Dalian, China*, 2017, pp. 140–144. https://doi.org/10.23919/ChiCC.2017.8027334

[5] P. K. Dharavath and J. Ohri, "Position and vibration control of single link flexible joint manipulator," in *2020 First IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India*, 2020, pp. 1–6. https://doi.org/10.1109/ICMICA48462.2020.9242807

[6] M. Sayahkarajy, E. Supriyanto, and Z. Mohamed, "Principal vibration modes of a rigid-flexible manipulator," in *2017 International Conference on Robotics, Automation and Sciences (ICORAS), Melaka, Malaysia*, 2017, pp. 1–5. https://doi.org/10.1109/ICORAS.2017.8308060

[7] J. Annisa, I. M. Darus, M. Tokhi, and S. Mohamaddan, "Implementation of PID based controller tuned by evolutionary algorithm for double link flexible robotic manipulator," in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), Kuching, Malaysia*, 2018, pp. 1–5. https://doi.org/10.1109/ICASSDA.2018.8477615

[8] B. A. M. Zain, M. O. Tokhi, and S. F. Toha, "PID-based control of a single-link flexible manipulator in vertical motion with genetic optimisation," in *2009 Third UKSim European Symposium on Computer Modeling and Simulation, Athens, Greece*, 2009, pp. 355–360. https://doi.org/10.1109/EMS.2009.86

[9] M. Hong, W. Yao, Z. Zhu, and Y. Guo, "A hybrid PID controller for flexible joint manipulator based on state observer and singular perturbation approach," in *2020 39th Chinese Control Conference (CCC), Shenyang, China*, 2020, pp. 3599–3603. https://doi.org/10.23919/CCC50068.2020.9189446

[10] F. M. Botsali, M. Tinkir, M. Kalyoncu, and Ü. Önen, "Fuzzy logic trajectory control of flexible robot manipulator with rotating prismatic joint," in *2010 the 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore*, 2010, pp. 35–39. https://doi.org/10.1109/ICCAE.2010.5451849

[11] A. Dehghani and H. Khodadadi, "Fuzzy logic self-tuning PID control for a single-link flexible joint robot manipulator in the presence of uncertainty," in *2015 15th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea*, 2015, pp. 186–191. https://doi.org/10.1109/ICCAS.2015.7364904

[12] S. Ling, H. Wang, and P. X. Liu, "Adaptive fuzzy tracking control of flexible-joint robots based on command filtering," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 4046–4055, 2019. https://doi.org/10.1109/TIE.2019.2920599

[13] R. Sharma, P. Gaur, and A. Mittal, "An enhanced hybrid fuzzy logic controller for robotic manipulator," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India*, 2016, pp. 2465–2470.

[14] J. S. Milind, G. Arunkumar, and T. Manjunath, "PID control of a double link (2-link) flexible robotic manipulator (2-DOF) in the 3 DE space," in *2018 4th International Conference for Convergence in Technology (I2CT), Mangalore, India*, 2018, pp. 1–7. https://doi.org/10.1109/I2CT42659.2018.9058215

[15] J. Annisa, I. M. Darus, M. Tokhi, and S. Mohamaddan, "Implementation of PID based controller tuned by evolutionary algorithm for double link flexible robotic manipulator," in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), Kuching, Malaysia*, 2018, pp. 1–5. https://doi.org/10.1109/ICASSDA.2018.8477615

[16] W. J. Tang, Z. T. Liu, and Q. Wang, "DC motor speed control based on system identification and PID auto tuning," in *2017 36th Chinese Control Conference (CCC), Dalian, China*, 2017, pp. 6420–6423. https://doi.org/10.23919/ChiCC.2017.8028376

[17] H. S. Hameed, "Brushless DC motor controller design using MATLAB applications," in *2018 1st International Scientific Conference of Engineering Sciences-3rd Scientific Conference of Engineering Science (ISCES), Diyala, Iraq*, 2018, pp. 44–49. https://doi.org/10.1109/ISCES.2018.8340526

[18] S. K. Pradhan and B. Subudhi, "Position control of a flexible manipulator using a new nonlinear self-tuning PID controller," *IEEE/CAA J. Autom. Sin.*, vol. 7, no. 1, pp. 136–149, 2018. https://doi.org/10.1109/JAS.2017.7510871

[19] P. Shah and R. Sekhar, "Closed loop system identification of a DC motor using fractional order model," in *2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE), Bali, Indonesia*, 2019, pp. 69–74. https://doi.org/10.1109/MoRSE48060.2019.8998744

[20] H. Salim, B. Rahma, and B. H. A. N. Saloua, "Comparative assessment between pd and fuzzy controllers for a rigid-flexible manipulator robot," in *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET), Hammamet, Tunisia*, 2017, pp. 227–232. https://doi.org/10.1109/ASET.2017.7983696

[21] D. Vyas, J. Ohri, and A. Patel, "Comparison of conventional & fuzzy based sliding mode PID controller for robot manipulator," in *2013 International Conference on Individual and Collective Behaviors in Robotics (ICBR), Sousse, Tunisia*, 2013, pp. 115–119. https://doi.org/10.1109/ICBR.2013.6729269

**Nomenclature**

| | |
|---|---|
| $\theta$ | Angle |
| $^\circ$ | Degree of Angle |
| $\Delta e$ | Change of Error |
| $E$ | Error |
| $Kp$ | Proportional gain |
| $Ki$ | Integral gain |
| $Kd$ | Derivative gain |
| $L1$ | Link 1 |
| $L2$ | Link 2 |
| $q_1$ | Angle of Link 1 |
| $q_2$ | Angle of Link 2 |