Design PID Controller Tuned by Using Fuzzy Logic for 3 Link Robot Manipulator

Kamal M. H. Raheem[1], Ali Muhssen Abdul-Sadaha[*2], Nasseer K. Bachacheb[3]

[1] Faculty of Information Technology, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq.
[2] College of Technical Engineering, University of Alkafeel, Al-Najaf, 54001, Iraq.
[3] Bilad Alrafidain University College, Diyala, 32001, Iraq
Emails: iraqikamal@gmail.com, kamal_mohammed@sadiq.edu.iq; ali.muhssen@alkafeel.edu.iq; dr.naseer@bauc14.edu.iq

Abstract

Robots are commonly used in industry, but they have limitations like complex dynamics, difficulties with flexibility, and nonlinearity. This research aims to enhance the tracking performance of a three-DOF open-chain robot manipulator. So, the driven dynamic equations will be utilized to identify the nonlinear robot model. The objective of this study is to achieve the desired performance of a three-degree-of-freedom (3-DOF) robot through the implementation of a Fuzzy Logic Self-Tuning Proportional-Integral-Derivative (PID) controller. The proposed PID controller exhibits notable distinctions when compared to the traditional PID controller. In conventional PID control, model parameters are determined by a range of procedures, including Ziegler-Nichols. However, in the context of fuzzy logic self-tuning PID control, these parameters are selected utilizing intelligent methodologies. This paper presents one of the smart methods (Fuzzy logic) as a tuner to obtain the PID parameter value. After the model of the 3-DoF Robot manipulator is driven, The PID controller tuned by Fuzzy logic is created in two scenarios:

1. Using the error and error derivative.
2. Using the error and error integral.

The data obtained from the simulation indicate that the proposed controllers have the ability to enhance the overall efficiency of the 3-DoF Robot manipulator.

Keywords: 3-DOF Robot manipulator, nonlinear model, PID, Fuzzy Logic Controller, PID tuned by Fuzzy logic.

1. Introduction

Many new control approaches have been suggested in recent decades to control robotic systems [1, 2]. A new technique is employed to define the robotic system's behavior while considering the loops' interaction. Most of the controllers are constructed on rigid basis robots, which may create instability[3]. When the flexibility effect is considered, the control system will be improved. It is anticipated that employing lighter-weight mechanical frameworks for flexible robots would increase the performance of mechanical manipulators[4]. The primary goal of the control system is to position the robot manipulator precisely without vibration or bending[5]. The robot manipulator is frequently utilized for various reasons, including quicker movements, increased load capacity, and less energy [5]. The flexible links cause technological issues, such as the flexibility of the robot arms. As a result, precise control becomes much more challenging. The initial stage in designing and controlling a 3-DOF robot is committed to those who believe that the elasticity of the connection is formed in just one direction, such as the vertical axis[6]. The Linear Quadratic Gaussian controller is used to control the robot's position. Position control of single link robot rotates horizontally in [1]. In [7], an optimum

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

PD controller for two links robots is proposed. Its objective would be to position the robot link quickly. In [8], a 3-DOF robot arm is controlled with a Fuzzy Logic Controller "FLC" approach. A 3-DOF open-chain robot arm is designed and controlled using two different control methods: a traditional PID controller and a sliding mode controller (SMC) [9]. A complicated controller combining a Fuzzy controller and a Linear Quadratic Regulator (LQR) was proposed in [10, 11], which described an energy-based nonlinear control approach (using the Lyapunov function) for a 2-DOF robotic manipulator. [12] Investigates a robust control approach based on neural networks and introduces a 3-DOF Robot manipulator. [13] created a fuzzy logic controller under the uncertainty situation using a PID controller for a 2-DOF robot manipulator. Other approaches mentioned in the literature include Model Predictive Control (MPC), impedance control, and adaptive control [14,15].

This study describes a 3-DOF robot arm. Then, using dynamic relations, the modeling of the nonlinear robot is derived. The PID controller tuned by the Fuzzy Logic technique is designed for 3-DOF open-chain robot manipulator control.

After the introduction, the first section of this work explains the robot model with a 3-DOF manipulator system. A traditional PID controller is designed and examined in the next section. The third section focused on the PID controller tuned by the Fuzzy logic. The assessment of this controller is conducted in two distinct scenarios, whereby the inaccuracy in the integral and derivative is employed. The simulation results show the response of the robotic system and the comparison between the reactions; the last section represents the conclusion.

2. Modeling the Robot Manipulator

The diagram of the robot manipulator with a 3-DOF open chain is shown in Figure 1.

The Lagrange technique is used for this system, and the system equations are based on conserving energy [11,13].

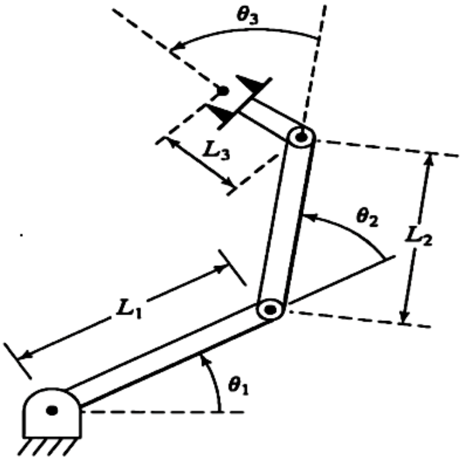

Figure 1: Open-chain three links robot manipulator

First, all system components' kinetic and potential energy are calculated [9]. So:

$$L_{(\theta,\dot{\theta})} = K_{(\theta,\dot{\theta})} - P_{(\theta)} \qquad (1)$$

Where:

$L_{(\theta,\dot{\theta})}$ It is the Lagrange energy.

$K_{(\theta,\dot{\theta})}$ It is the Kinetic energy.

$P_{(\theta)}$ It is the Potential energy.

The equations of the required torque for each manipulator joint are given by [9]

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \tag{2}$$

Knowing that

i It is denoted as the number of links, which is 1,2 and 3.

θ_i It is denoted as the angle of the link *i*.

$\dot{\theta}_i$ It is denoted as the angular velocity of the link *i*.

τ_i It is denoted as the torque of the link *i*.

The following is a voltage-torque relationship [16]:

$$\tau_i = \frac{K_i}{R_{m_i}} v_i - \frac{{K_i}^2}{R_{m_i}} \dot{\theta}_\iota \tag{3}$$

$$K_i = K_{Gi} K_{mi} \tag{4}$$

K_{mi} is the constant of the motor of link *i*.

K_{Gi} is the Gear ratio of the motor of link *i*.

R_{mi} is the resistance of the motor of link *i*.

So, the torque equations of all three robotics links will be as

$$[\,\tau\,] = [\,M_{(\theta)}\,]\,[\ddot{\theta}] + [\,C_{(\dot{\theta},\theta)}\,] + [\,G_{(\theta)}\,] \tag{5}$$

When:

τ is a 3x1 torque vector.

M is a 3x3 mass matrix.

C is a 3x1 Coriolis vector.

G is a 3x1 Gravity vector.

Table 1 below displays the specifications of the robot manipulator.

Table 1: The parameters of 3-DOF Robot.

Parameter	Unit	Symbol	Link1	Link2	Link3
Mass	Kg	mi	0.4	0.3	0.2
Length	m	Li	0.36	0.27	0.18
Moment of inertia	Kg.m^2	Ji	0.0432	0.0182	0.0054
Gear Ratio	-------	KG	75	75	75
Constant of Motor	N/rad/sec	Km	0.008	0.008	0.005

Figure 2 shows the 3-link robot manipulator constructed using a Sim-Mechanics Environment in Matlab.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

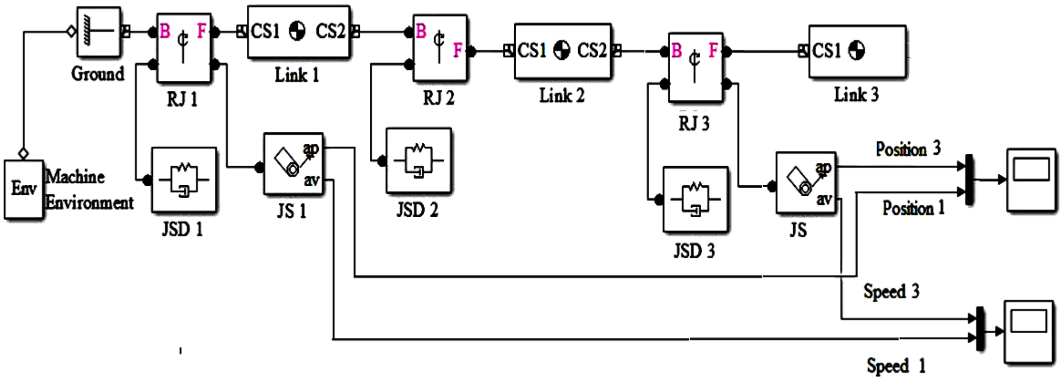

Figure 2: three links robot manipulator in sim-mechanic Environment.

3. PID Controller Design

PID is an abbreviation for Proportional Integral Derivative, representing the feedback controller. The difference between the desired and actual values is called the error. The control variable is based on the error and some variables of the measured process [17]. This type of control is very popular to use, and its formula is shown below [9], and the structure is shown in Figure 3.

$$e_i(t) = r_i(t) - y_i(t) \quad (6)$$

$$u_i(t) = K_{Pi}\, e_i(t) + K_d \frac{de_i(t)}{dt} + \frac{1}{K_i} \int e_i(t)\; dt \quad (7)$$

Knowing that

$e(t)$ is denoted as the error function.

K_P is denoted as the proportional control coefficient, which provides the control is directly proportional to the error.

K_d is denoted as The utilisation of the derivative control coefficient is employed to enhance the transient response.

K_i is denoted as the integral control coefficient is employed to mitigate steady-state errors.

$r_i(t)$ is denoted as the desired angle of link *i*.

$y_i(t)$ is denoted as the actual angle of link *i*.

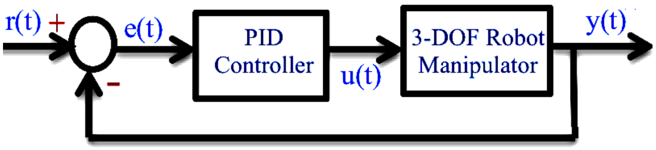

Figure 3: closed-loop control diagram.

The following diagram shows the behavior of a PID controller:

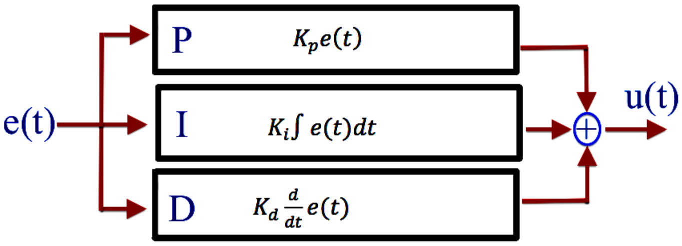

Figure 4: The PID controller's architecture.

The impact of the PID parameter is demonstrated in Table 2.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

Table 2: The impact of the proportional-integral-derivative (PID) parameter on system responses.

Parameter	**Rise Time**	**Settling Time**	**Overshoot**	**SS Error**
Kp	Decrease	Minor Increase	Increase	Decrease
Ki	Minor Decrease	Decrease	Decrease	Minor Change
Kd	Minor Increase	Increase	Increase	Major Decrease

The controller must possess the capability to stabilize the system in order to achieve a suitable response. Figure 5 is the result of the simulation.

The overshoot, settling time, and rise time for each Link in the 3-DOF robot manipulator under the PID controller is illustrated in Table 3, and the error in the angle of each Link is shown in Figure 4.

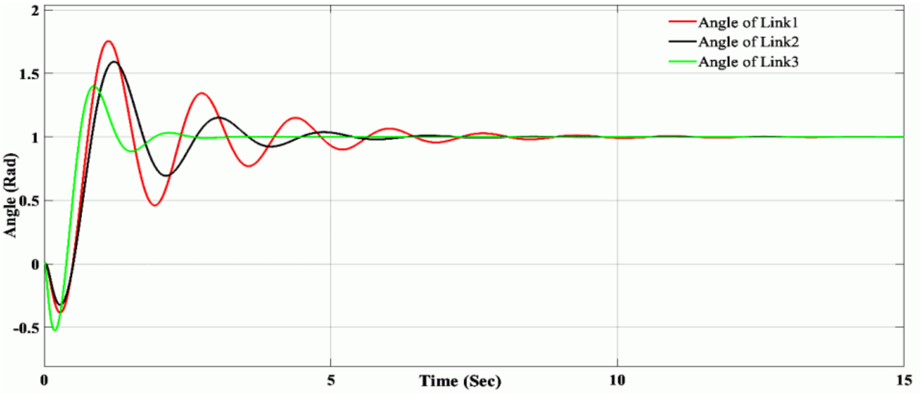

Figure 5: Response of the 3-DOF robot manipulator using PID controller.

Table 3: The effect of the PID parameter on system responses.

Parameter	**Link 1**	**Link 2**	**Link 3**
Rise Time	1 sec	1.2 sec	1.1 sec
Settling Time	4 sec	8 sec	12.5 sec
Max. Overshoot	45%	55%	75%

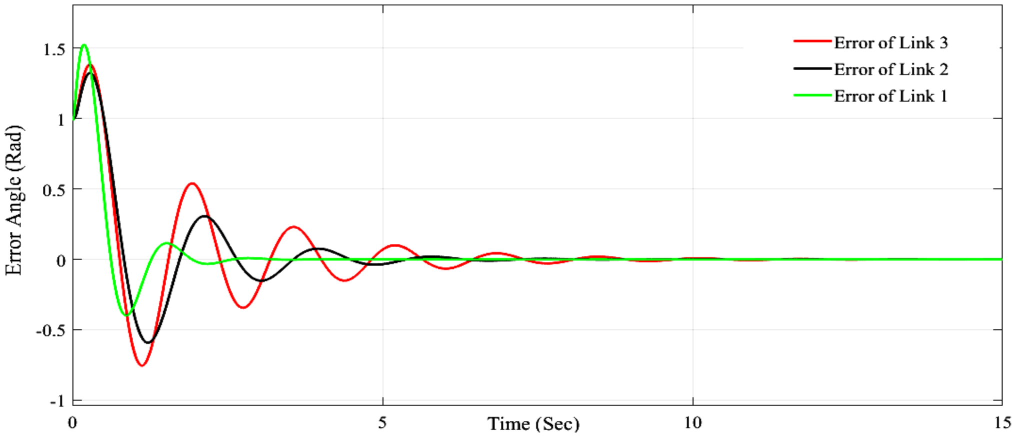

Figure 6: Error of the Link's angle for 3-DOF robot arm using PID controller.

The step response of the PID controller .could be better As a result, the proportional-integral-derivative (PID) controller exhibits a greater degree of overshoot and a longer duration for the settling process. So, the PID controller could be more optimal. The signal of the PID controller is shown in the figure 7.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

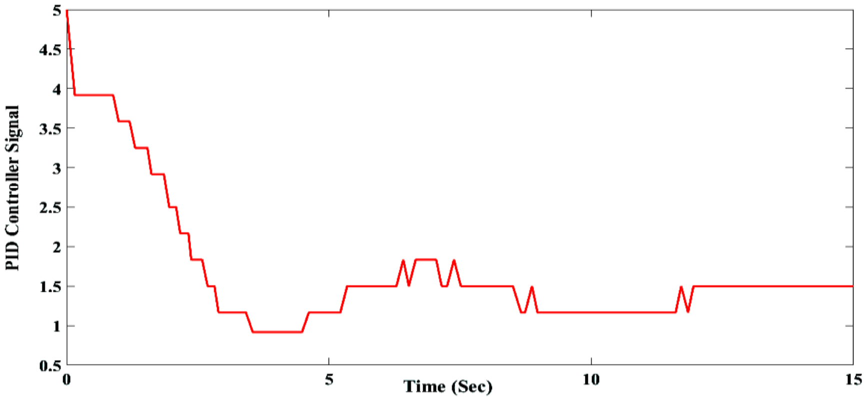

Figure 7: PID Controller Signal.

4. PID Controller Tuned by Using Fuzzy Logic

The controller's design is suboptimal in the event of a change in the system's condition. As a result, the system responsiveness will be changed. Several intelligence strategies may be utilized to adjust the PID controller to get a steady and appropriate response. One of these strategies is fuzzy logic based on human strategy and experiences. The figure 8 illustrates the control procedure.

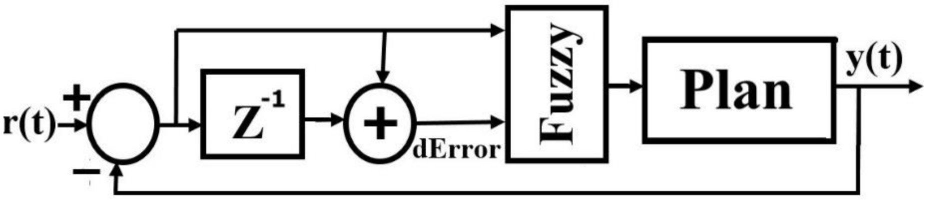

Figure 8: Fuzzy control process [18].

Fuzzy logic has main and vital parts, as shown in Figure 9:

1. Interface of fuzzification.
2. Base of knowledge.
3. Interface mechanism.
4. Defuzzification.

The fuzzification transforms the input to a suitable value for the subsystem. The input variable value has been transferred into the appropriate linguistic value by performing the fuzzification function [19]. During the process of fuzzification, a scalar value is converted into a fuzzy value. The process of defuzzification entails transforming a fuzzy inference activity into a crisp, non-fuzzy outcome. The method of defuzzification involves converting fuzzy outputs into a singular or crisp output value. A rule base is a decision-making unit replicating a human decision-making process based on an understanding of the control algorithm. Each rule is (if - then) with a condition. The rule of control is evaluated when the inference mechanism uses the fuzzy input variable.

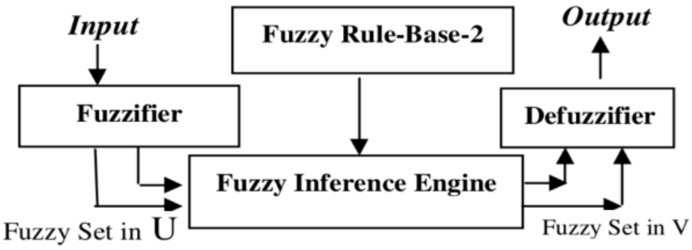

Figure 9: Structure of fuzzy control design [9].

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

In the initial design phase, it is necessary to establish the appropriate range for both the Input Membership Function (IMF) and the Output Membership Function (OMF). In the suggested controller, the parameters of the PID controller (KP, KI, KD) are tuned and predicted by the Fuzzy Controller, as shown in Figure 10.

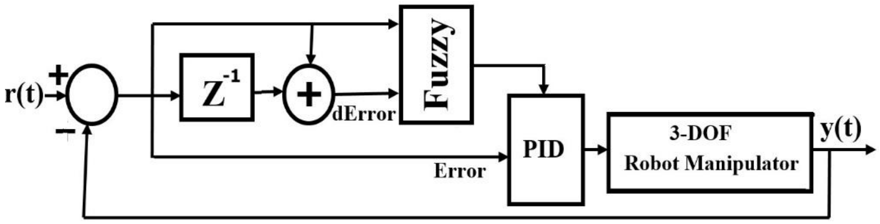

Figure 10: Fuzzy PID Controller Structure [18].

This controller's most significant component is its Fuzzy Logic rules [20,21]. The fuzzy rules depending on [9]:

Maintaining the output value if ess =0, and the value of output = set value.

According to the sign and quantity of the error changes in location and velocity, the output value = specified value.

The error *e(n)* is mathematically defined as the discrepancy between the desired or goal value and the observed or actual value, as indicated by the following equation. [9]

$$e(n) = \theta_d(n) - \theta_{fuzzy}(n) \tag{8}$$

$$\Delta e(n) = e(n) - e(n-1) \tag{9}$$

In most cases, this assumption is satisfied

Case (1) $e(n) < 0 \;\;\&\;\; \Delta e(n) > 0 \; Then \quad \theta_d(n) < \theta_{fuzzy}(n)$

Case (2) $e(n) > 0 \;\;\&\;\; \Delta e(n) < 0 \; Then \quad \theta_d(n) > \theta_{fuzzy}(n)$

Where $\theta_d(n)$ *is the desired angle.*

$\theta_{fuzzy}(n)$ is the signal of the Fuzzy Logic Controller

$e(n)$ is the error signal.

$\Delta e(n)$ is the changing of the error signal.

The Fuzzy control component within the Fuzzy Logic Self-Adjusting PID system is tasked with the adjustment of the proportional gain (KP), integral gain (KI), and derivative gain (KD) parameters. Consequently, the fuzzy system possesses a total of a pair of inputs and three outputs. The functions of the membership are depicted in Figures 11-15.

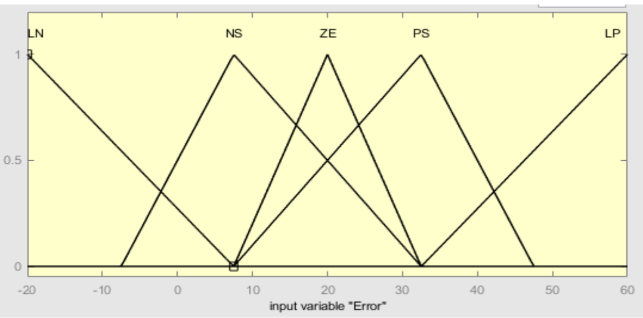

Figure 11: Input (Error) Function membership.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

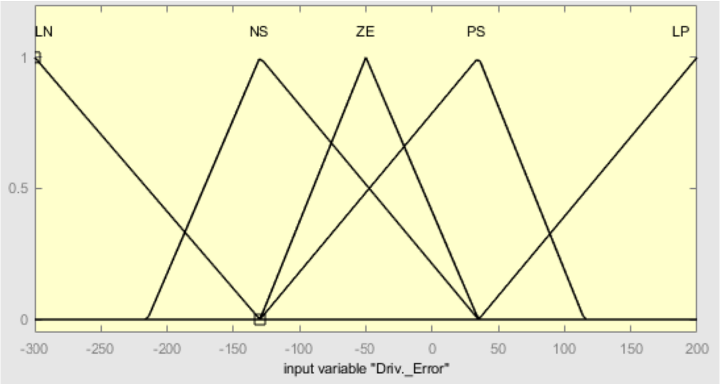

Figure 12: Input (Derivative of Error) Function membership.

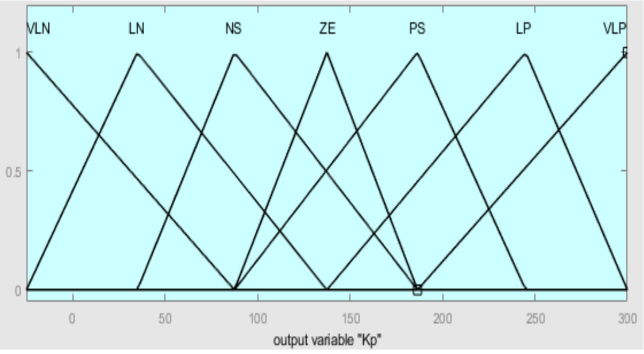

Figure 13: Output (KP) Function membership.

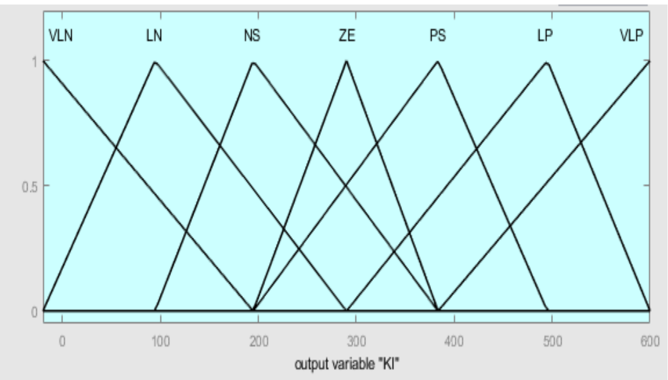

Figure 14: Output (KI) Function membership.

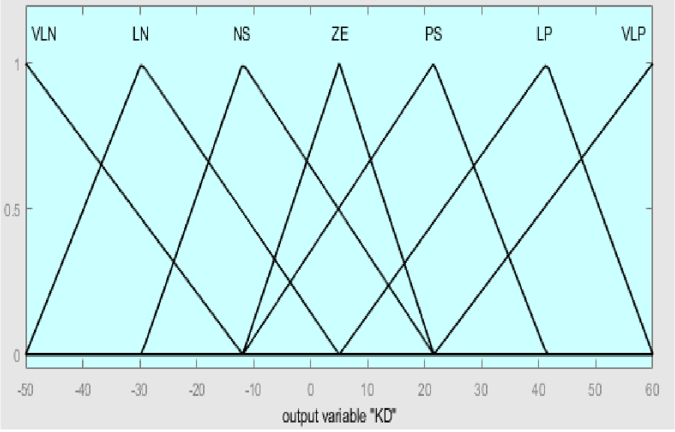

Figure 15: Output (KD) Function membership.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

The fuzzy PID control rules are presented in Tables 4, 5, and 6.

Table 4: Fuzzy Rules pertaining to the concept of KP.

		Error Derivative				
		LN	**NS**	**ZE**	**PS**	**LP**
Error	**LN**	VLP	VLP	VLP	VLP	VLP
	NS	PS	PS	PS	PS	PS
	ZE	VLN	VLN	LN	NS	NS
	PS	PS	PS	PS	ZE	ZE
	LP	VLP	VLP	VLP	VLP	VLP

Table 5: Fuzzy Rules pertaining to the concept of KI.

		Error Derivative				
		LN	**NS**	**ZE**	**PS**	**LP**
Error	**LN**	ZE	ZE	ZE	ZE	ZE
	NS	NS	NS	NS	NS	NS
	ZE	LN	LN	LN	LN	LN
	PS	NS	NS	NS	NS	NS
	LP	ZE	ZE	ZE	ZE	ZE

Table 6: Fuzzy Rules pertaining to the concept of KD.

		Error Derivative				
		LN	**NS**	**ZE**	**PS**	**LP**
Error	**LN**	VLP	VLP	VLP	VLP	VLP
	NS	NS	NS	NS	NS	NS
	ZE	LN	LN	LN	LN	LN
	PS	NS	NS	NS	NS	NS
	LP	ZE	ZE	ZE	ZE	ZE

Based on the provided charts, the inputs consist of LN (Large Negative), NS (Negative Small), ZE (Zero), PS (Positive Small), and LP (Large Positive), while the outputs include VLN (Very Large Negative), LN (Large Negative), NS (Negative Small), ZE (Zero), PS (Positive Small), LP (Large Positive), and VLP (Very Large Positive). According to the rules, the Fuzzy Logic Controller Surfaces are shown in Figures 16, 17, and 18 for KP, KI, and KD, respectively.

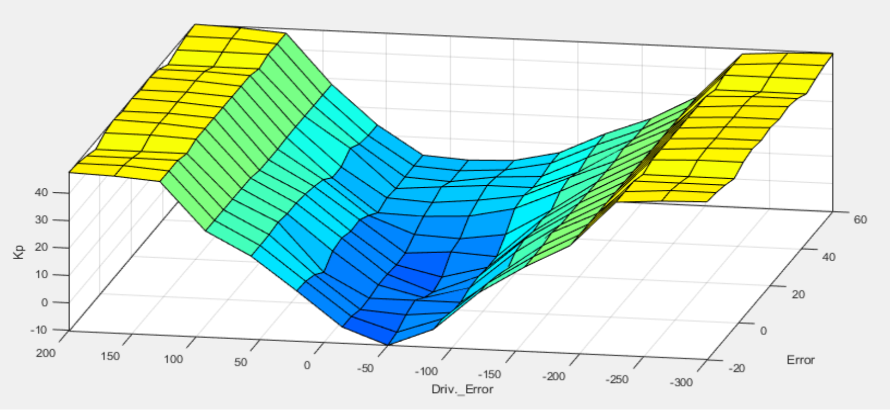

Figure 16: Fuzzy Logic Controller Surface for KP.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

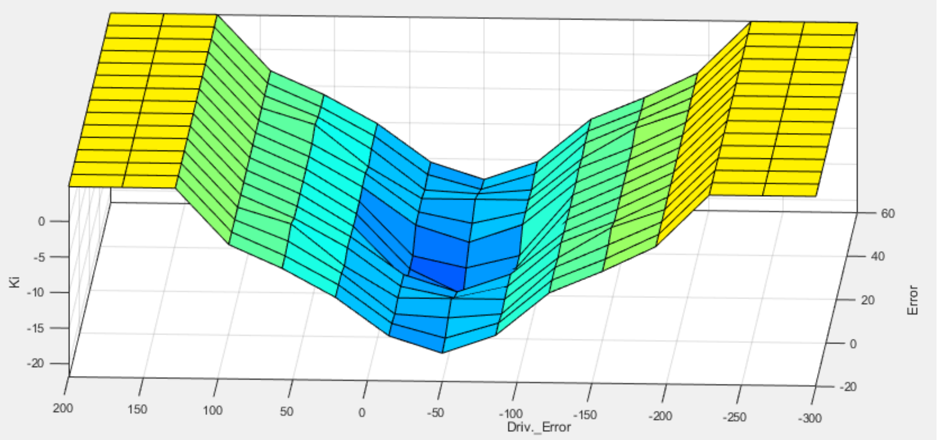

Figure 17: Fuzzy Logic Controller Surface for KI.

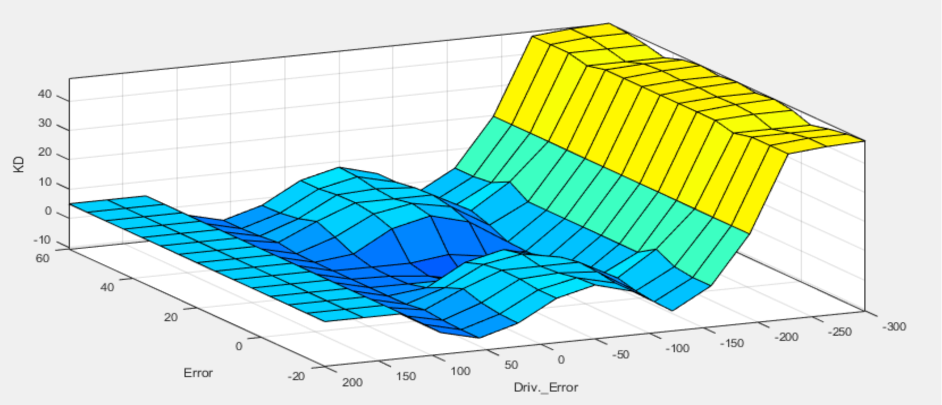

Figure 18: Fuzzy Logic Controller Surface for KD.

The controller must possess the capability to achieve system stabilisation in order to provide a good response.

After determining the ideal PID settings with the Fuzzy Logic Auto Tunning controller, the step response is plotted to calculate the angle. The following in Figure 19 is a MATLAB simulation of a PID controller response:

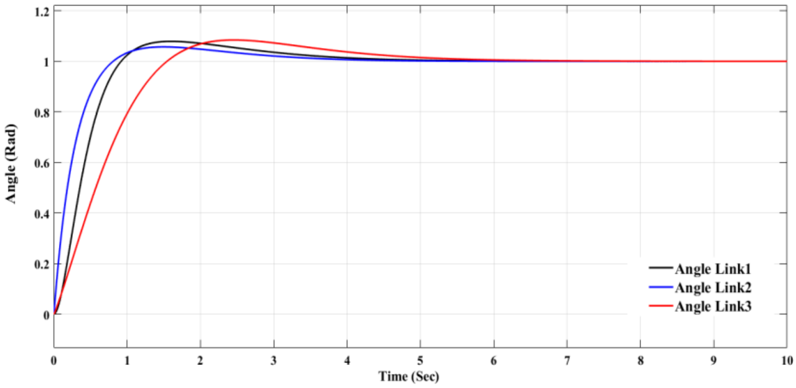

Figure 19: Response of the 3-DOF robot arm using PID Auto Tuning controller.

The error in the rotation of the link is explained in the figure 20.

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

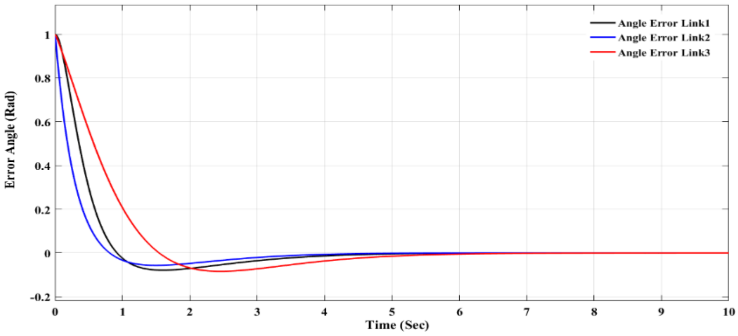

Figure 20: Error of the Link’s angle for 3-DOF robot arm using PID Auto Tuning controller.

The PID auto-tuning by fuzzy logic control signal will be as in Figure 21.

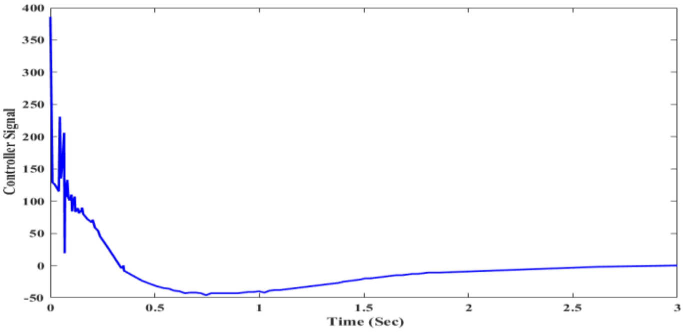

Figure 21: PID Auto Tunning Controller Signal.

The overshoot, settling time, and rise time for each link in the 3-DOF robot arm using the PID Auto Tunning controller is illustrated in Table 7.

Table 7: PID Auto Tunning Controller Signal.

Parameter	**Link 1**	**Link 2**	**Link 3**
Rise Time	1 sec	0.8 sec	1.5 sec
Settling Time	5 sec	4.5 sec	6 sec
Max. Overshoot	10%	7%	13%

Depending on the data in Tables 3 and 7, the Fuzzy Logic auto-tuned PID controller is better than the conventional PID controller regarding overshoot and settling time. As a result, it has a faster response time.

5. Conclusion

This study presents the design of the Traditional PID controller and the PID auto-tuning method using Fuzzy logic. for a 3-DOF open-chain robot manipulator, considered a nonlinear system. The preceding discussion attempts to propose an auto-tuning PID based on fuzzy logic for controlling the position of a three-series link robot manipulator. Performing control aims to track desired trajectories for three links. While using the PID controller, noticed that the response has a significant amount of overshoot and a longer settling time. Then, PID tuning by fuzzy logic is designed to obtain high speed and low overshoot values compared with traditional PID controllers. The simulation results show that the suggested controller has a lower overshoot and less setting time corresponding to the PID controller, so the recommended controller is better than the classical PID controller. The Robotic manipulators' performance will be improved as expected.

Funding: The present study did not get any financial support from external sources.

Conflicts of Interest: “The authors declare no conflict of interest.”

Doi : https://doi.org/10.54216/IJNS.xxxxxx
Received: July 20, 2021 Accepted: Jan 04, 2022

References

[1] M. Benosman and G. Le Vey, “Control of flexible manipulators: A survey,” *Robotica*, vol. 22, no. 5, pp. 533–545, 2004.

[2] S. Ozgoli and H. D. Taghirad, “A survey on the control of flexible joint robots,” *Asian J. Control*, vol. 8, no. 4, pp. 332–344, 2006.

[3] A. Fenili and J. M. Balthazar, “The rigid-flexible nonlinear robotic manipulator: Modeling and control,” *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, no. 5, pp. 2332–2341, 2011.

[4] Y. Li, S. Tong, and T. Li, “Adaptive fuzzy output feedback control for a single-link flexible robot manipulator driven DC motor via backstepping,” *Nonlinear Anal. Real World Appl.*, vol. 14, no. 1, pp. 483–494, 2013.

[5] S. Ulrich, J. Z. Sasiadek, and I. Barkana, “Modeling and direct adaptive control of a flexible-joint manipulator,” *J. Guid. Control. Dyn.*, vol. 35, no. 1, pp. 25–39, 2012.

[6] R. H. Cannon Jr and E. Schmitz, “Initial experiments on the end-point control of a flexible one-link robot,” *Int. J. Rob. Res.*, vol. 3, no. 3, pp. 62–75, 1984.

[7] J.-H. Park and H. Asada, “Integrated structure/control design of a two-link nonrigid robot arm for high-speed positioning,” in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 735–736.

[8] M. A. J. Maktoof, A. J. M. Jawad, H. M. Abd, A. Husain, and A. Majdi, “Using a fuzzy logic integrated machine learning algorithm for information fusion in smart parking,” *Fusion Pract. Appl.*, vol. 11, no. 1, pp. 114–128, 2023, doi: 10.54216/FPA.110109.

[9] A. Green and J. Z. Sasiadek, “Dynamics and trajectory tracking control of a two-link robot manipulator,” *J. Vib. Control*, vol. 10, no. 10, pp. 1415–1440, 2004.

[10] X. Bo, K. Fujimoto, and Y. Hayakawa, “An energy-based nonlinear control for a two-link flexible manipulator,” *Trans. Jpn. Soc. Aeronaut. Space Sci.*, vol. 48, no. 160, pp. 92–101, 2005.

[11] Y. Li, G. Liu, T. Hong, and K. Liu, “Robust control of a two-link flexible manipulator with quasi-static deflection compensation using neural networks,” *J. Intell. Robot. Syst.*, vol. 44, pp. 263–276, 2005.

[12] 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)*, 2015, pp. 186–191.

[13] I. H. Akyuz, E. Yolacan, H. M. Ertunc, and Z. Bingul, “PID and state feedback control of a single-link flexible joint robot manipulator,” in *2011 IEEE International Conference on Mechatronics*, 2011, pp. 409–414.

[14] .. N. K., M. Prateek, N. Singh, and A. Saxena, “An Implicit Controlling of Adaptive Neuro Fuzzy Inference System Controller for The Grid Connected Wind Driven PMSG System,” *Fusion Pract. Appl.*, vol. 12, no. 2, pp. 193–205, 2023, doi: 10.54216/fpa.120216.

[15] K. M. H. Raheem, O. O. Hassoon, and A. M. Abdul-Sadah, “Modeling 3-Degree of freedom robotics manipulator with PID and sliding mode controller,” in *AIP Conference Proceedings*, 2022. doi: 10.1063/5.0066824.

[16] K. Groves and A. Serrani, “Modeling and nonlinear control of a single-link flexible joint manipulator,” *Appendices*, 2004.

[17] M. F. Rahmat and M. M. Ghazaly, “Performance comparison between PID and fuzzy logic controller in position control system of DC servomotor,” *J. Teknol.*, vol. 45, no. 4, pp. 1–17, 2006.

[18] H. W. A. Darminto, Y. G. Yusuf, A. Yodinata, and L. Haryanto, “Performance analysis fuzzy-PID versus fuzzy for quadcopter altitude lock system,” *J. Teknol.*, vol. 77, no. 22, pp. 33–38, 2015.

[19] M. J. Nekooei, Jaswar, and A. Priyanto, “A simple fuzzy logic diagnosis system for control of internal combustion engines,” *J. Teknol.*, vol. 74, no. 5, pp. 121–124, 2015.

[20] R. Adnan, M. Tajjudin, N. Ishak, H. Ismail, and M. H. F. Rahiman, “Self-tuning fuzzy PID controller for electro-hydraulic cylinder,” in *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, 2011, pp. 395–398.

[21] R. Akbari-Hasanjani, S. Javadi, and R. Sabbaghi-Nadooshan, “DC motor speed control by self-tuning fuzzy PID algorithm,” *Trans. Inst. Meas. Control*, vol. 37, no. 2, pp. 164–176, 2015.