Design Fuzzy, PID and Fuzzy-PID Controller for 3P-Cartesian Robot

Alireza Amiri
Department of Mechatronics Engineering
KNTU University of Technology
Tehran, Iran
ali.amiri@email.kntu.ac.ir

***Abstract*— This project explores the control strategies for a 3P-Cartesian robot by implementing and comparing a fuzzy controller and a PID controller. The study evaluates key control specifications such as response time, rise time, and steady-state error through simulations conducted in Simulink. The PID controller parameters were initially tuned using Simulink's tuner and subsequently adjusted manually. The fuzzy controller, designed using the Fuzzy Logic Designer toolkit in MATLAB, employs seven membership functions each for error and its derivative, along with seven membership functions for its single output. Each controller was independently applied to the x, y, and z axes to ensure individual axis control without cross-axis interference. The comparative analysis reveals that both controllers exhibit satisfactory performance, with the fuzzy controller demonstrating slightly higher steady-state error compared to the PID controller.**

Keywords*— *Fuzzy controller, PID controller, Simulink, Robotics, Control

I. Introduction

3P_cartesian robots, characterized by their three degrees of freedom (DOF) in x, y, and z axes, play a crucial role in precision manipulation tasks. While lacking rotational movements, these robots ensure precise spatial control, vital for various industrial applications. Central to their operation is the implementation of effective controllers, which maintain positional accuracy and stability during operation. The design of controllers for 3P_cartesian robots is based on the fundamental assumption that the three links (corresponding to x, y, and z axes) operate independently. This independence allows for the individual design and implementation of controllers tailored to each axis. Notably, the z-axis, influenced directly by gravity, requires a more refined control strategy due to its distinct dynamics compared to the other axes.

The dynamical equations governing the motion of these links are expressed as follows:

$$\ddot{x} = 0.0625\, F\_1$$

$$\ddot{y} = 0.0323\, F\_2$$

z=0.1667(F3−58.8)

where F_1, F_2 and F_3 represent the forces applied to the respective links, and x, y, x denote their positions.

In this project, closed-loop controllers are employed, beginning with a trajectory planner that orchestrates the robot's movement from initial to target points. The trajectory planner function efficiently generates smooth and controlled motion profiles for the tool or end effector. By incorporating gradual acceleration and deceleration phases during the specified blending periods, it ensures minimal jerk and stable transitions between positions. This capability is essential for maintaining precision in robotic operations, enabling accurate trajectory tracking from initial to target positions within the defined time frame. Effective parameter tuning of the blending period blend and total movement time T optimizes the planner's performance, enhancing overall operational efficiency and motion quality in robotic applications.

The closed-loop control section integrates two distinct control strategies: the Proportional-Integral-Derivative (PID) controller and the Fuzzy Logic controller. The PID controller regulates the system based on proportional, integral, and derivative actions, providing a well-established method for precise and stable control in robotics. In contrast, the Fuzzy Logic controller utilizes linguistic variables and fuzzy rules to accommodate nonlinearities and uncertainties, offering robust performance in varied operating conditions.

Tuning parameters in the PID controller and defining membership functions and rules in the Fuzzy Logic controller are pivotal tasks to ensure optimal performance and responsiveness in real-world scenarios.

The actuator, a pivotal component known as the 3P_manipulator, applies controlled forces to the robot's links based on the output from the controllers. This process incorporates dynamic considerations such as the weights of the links and the gravitational effects in the z-direction, ensuring accurate and efficient movement of the robot.

To accurately determine the actual position of the end effector following the inverse kinematics calculations, an End Effector Position block is utilized. This block reverses the initial transformation to derive the true spatial coordinates of the end effector, essential for verifying the efficacy of the trajectory planning and control strategies.

In conclusion, this report investigates the application and performance analysis of PID and Fuzzy Logic controllers within the context of a 3P_cartesian robot system. By integrating precise trajectory planning, robust closed-loop control strategies, and effective actuation mechanisms, the study aims to enhance the operational efficiency and accuracy of such robotic systems in industrial settings.

II. PID CONTROLLER DESIGN

After understanding the different components of the whole system, the control loop is defined as shown in the image below. Figure 1

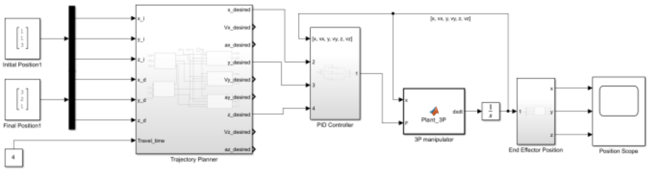

Figure 1 3P-Cartesian Robot Control Model

Initially, a PID controller was defined for the system. The PID controller block inputs the desired positional trajectory profile that each of the links must travel as a reference, along with the traveled path as inputs, and then outputs a control command for each link. The output is an array of three values, each corresponding to an axis (x, y, z) and utilized by the manipulator.

Inside the PID controller block, as shown inFigure 2, there are three separate and independent PID controllers, one for each axis. The parameters for the proportional (P), integral (I), and derivative (D) gains were initially tuned using the Simulink tuner. However, the initial values were too large, resulting in noisy and high-amplitude outputs. Consequently, the PID parameters were manually re-tuned to achieve stable outputs.

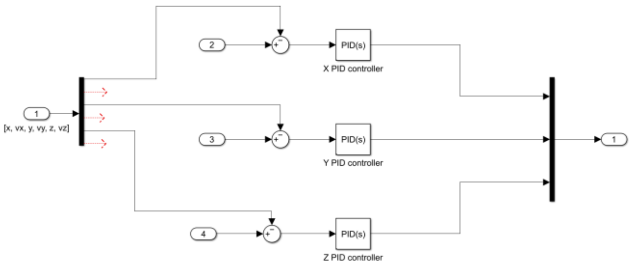

Figure 2 PID Controlelr Block

The finalized values for these parameters, along with control specifications such as rise time, settling time, and steady-state error, are presented in Table 1.

Table 1 PID controller parameters

Axis	**Kp**	**Ki**	**Kd**	**Rise time**	**Settling time**	**Overshoot %**
X	107	10	255	0.23	2.99	1
Y	107	10	255	0.23	2.99	1
Z	700	272	400	0.03	0.24	2.34

Additionally, the step response of each of these controllers is depicted in *Figure 3 PID Controllers step responses*

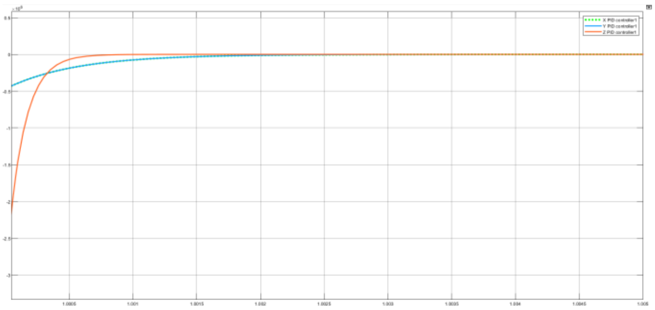

Figure 3 PID Controllers step responses

With these controllers tuned, we can assess the system's performance. The system is given two points as the initial and final desired positions, [1,1,3] and [3,2,1], respectively. These points are fed into the trajectory planner block, which generates the trajectories shown inFigure 4 Trajectory planner output. The controller outputs and the final trajectory traveled by the end effector are presented in

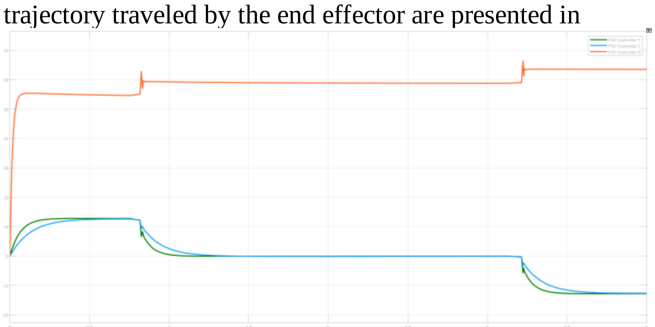

Figure 5 PID controller output, Figure 6 Trajectory traveled by the end effector

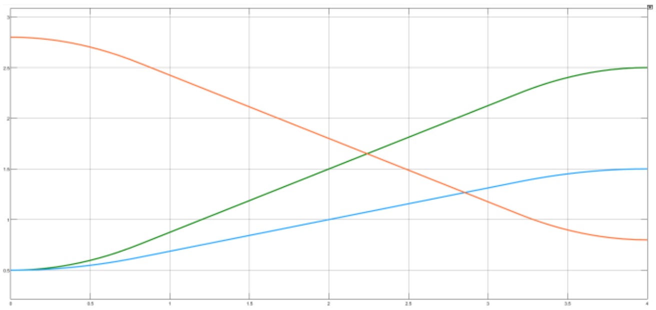

Figure 4 Trajectory planner output

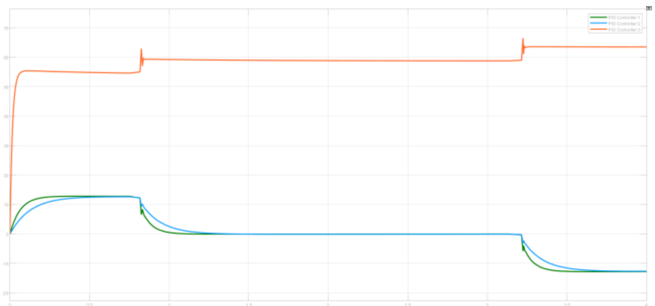

Figure 5 PID controller output

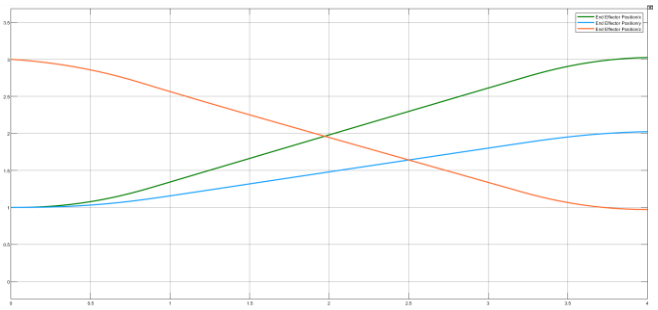

Figure 6 Trajectory traveled by the end effector

III. FUZZY CONTROLLER DESIGN

In this section, a fuzzy controller is designed to control the 3P-cartesian robot. Fuzzy controllers leverage fuzzy logic, a form of multi-valued logic derived from fuzzy set theory to handle reasoning that is approximate rather than precise. This makes fuzzy controllers particularly suitable for complex, nonlinear systems where traditional control methods may not perform well.

Fuzzy logic controllers are based on the concept of fuzzy sets, where an element's membership in a set is gradual rather than binary. This allows for handling of uncertainty and imprecision. In a fuzzy controller, the inputs are usually the error (difference between desired and actual output) and the derivative of the error (rate of change of error). These inputs are fuzzified into linguistic variables, which are then processed using a set of rules to produce a control output.

The fuzzification process involves mapping input values (error and its derivative) into fuzzy sets characterized by membership functions. For this project, the fuzzy controller uses two inputs: error and its derivative (edot). Each input is associated with seven membership functions, labeled as Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Big (PB). These membership functions range from -1 to 1, providing a comprehensive representation of possible input values.Figure 9 Fuzzy Controlled System Diagram,

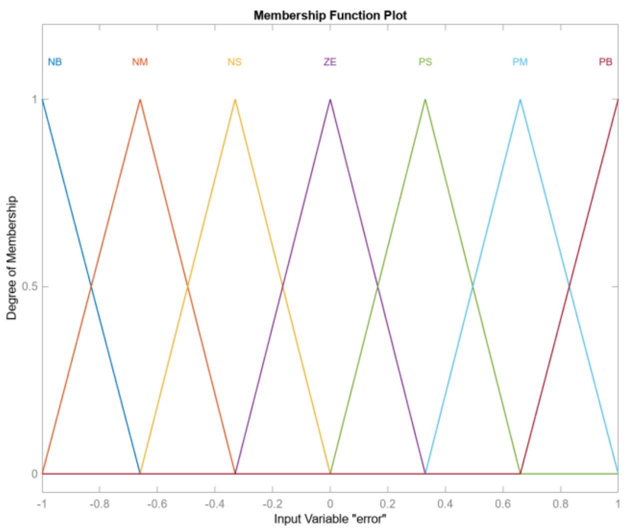

Figure 7 Membership function of FLC

Membership functions define how each point in the input space is mapped to a membership value between 0 and 1. In this project, we used triangular membership functions for simplicity and computational efficiency. The shape and range of these functions are crucial for accurately representing the linguistic variables and ensuring smooth control behavior.

For this project, the fuzzy rule base was constructed based on previously approved papers and expert knowledge. These rules are designed to handle various scenarios, ensuring smooth and accurate control of the robot's movements.

Once the inputs are fuzzified and the rules are evaluated, the inference engine computes the fuzzy output by combining the results of the rules. The most common inference method is the Mamdani method, which uses the minimum operator for rule evaluation and the maximum operator for output aggregation.Figure 8 Fuzzy rules based on e and edot

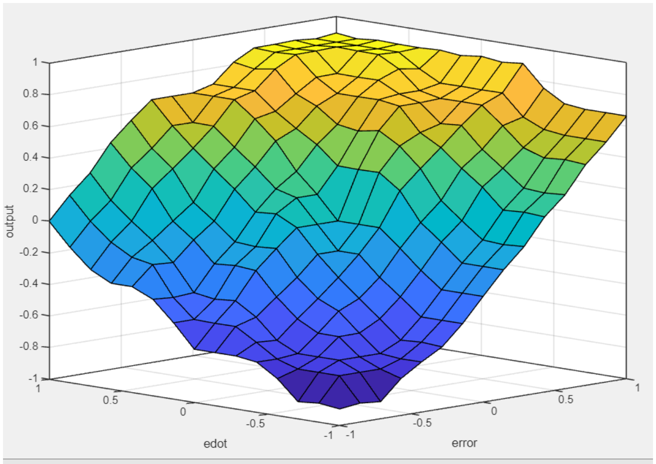

Figure 8 Fuzzy rules based on e and edot

The final step is defuzzification, where the fuzzy output is converted back into a crisp value to be used by the actuator. The centroid method is commonly used for defuzzification, as it provides a balanced and smooth output.

For this project, the fuzzy controller was designed using MATLAB's Fuzzy Logic Designer. The controller consists of two inputs (error and its derivative) and one output, each with seven membership functions. The input-output relationship is defined by the rule base, which was constructed based on empirical data and expert knowledge. The controller diagram is shown in Figure 9 Fuzzy Controlled System Diagram, illustrating the structure and flow of the fuzzy logic control system.

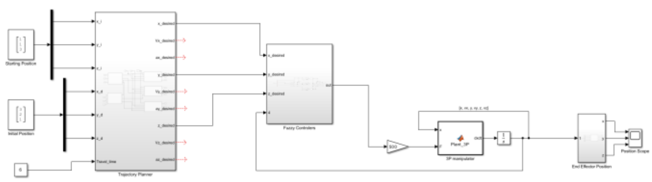

Figure 9 Fuzzy Controlled System Diagram

To ensure proper scaling, the inputs to the fuzzy block are normalized using gain blocks. This normalization process involves downscaling the inputs to fit within the range of the

membership functions and then upscaling the output to match the required control signal levels.
One challenge in using fuzzy controllers is the small magnitude of the raw output, which is often insufficient for direct application to the actuator. To address this, a gain block is used to upscale the output to a practical level. The fuzzy controller block diagram of the system is presented in Figure 10 Fuzzy block diagram

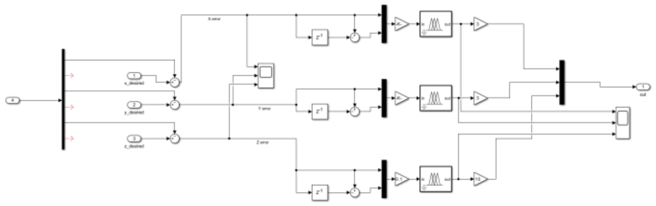

Figure 10 Fuzzy block diagram

To evaluate the performance of the fuzzy controller, the same initial and final positions used for the PID controller ([1,1,3] and [3,2,1]) were applied. The trajectory planner generated the desired trajectories, which were then used as reference inputs for the fuzzy controller.
The outputs of the fuzzy controller and the resulting path traveled by the end effector are presented in Figure 11 Fuzzy controller output,Figure 12 Trajectory traveled by end effector controlled by FLC

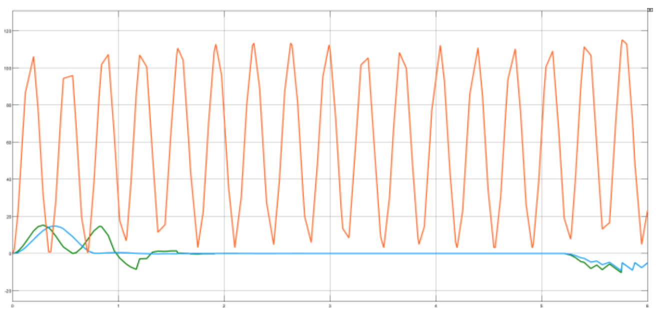

Figure 11 Fuzzy controller output

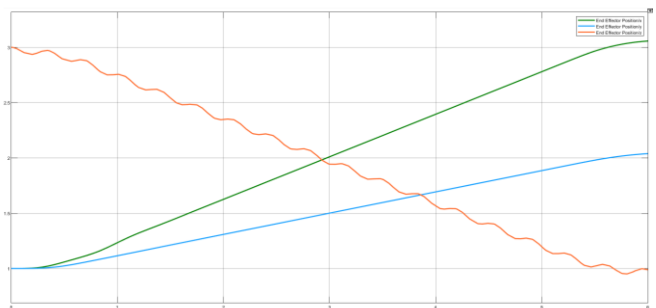

Figure 12 Trajectory traveled by end effector controlled by FLC

As observed in the above figure, one can recognize that the performance of controller in z-axis is not as stable as the other axes. This due to the inherit difference of dynamic equations of the mover on this axis, since in includes the gravitational force applied on the link.

IV. FUZZY-PID CONTROLLER

A Fuzzy-PID controller is basically a PID controller in which its parameters are dynamically tuned by a Fuzzy Logic Controller (FLC). This method, enables the system to overcome nonlinear behavior of systems. In Figure 13 Fuzzy-PID controller, a design for Fuzzy-PID controller is illustrated.

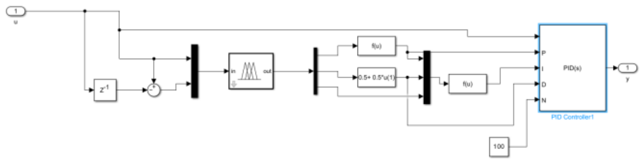

Figure 13 Fuzzy-PID controller

In this design, five membership function were considered for error and error derivative, two membership functions for Kp and Kd, and 4 mfs for alpha. These membership functions are presented in

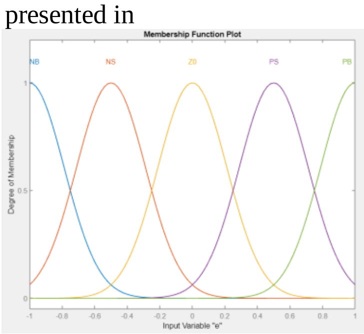

Figure 14 error and derivative of error mf

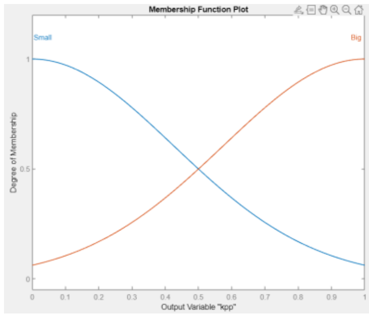

Figure 15 Kp and Kd mf

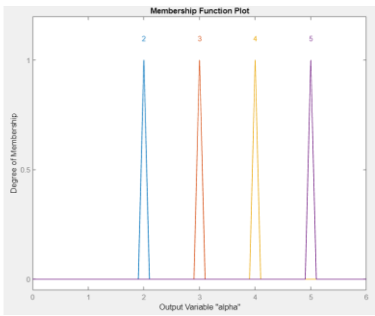

Figure 16 alpha mf

The rules defined for these fuzzy controller is as shown in figures below

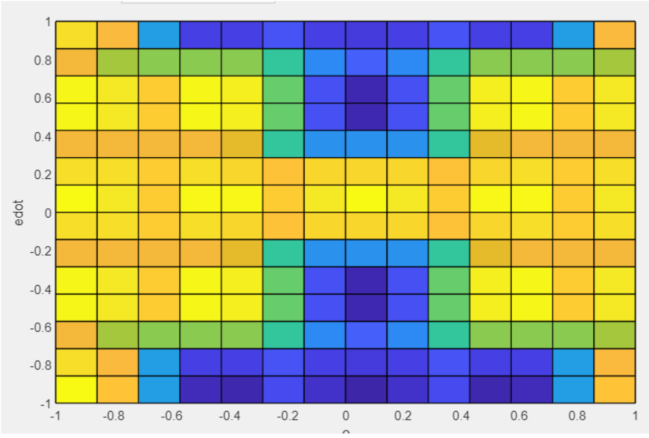

Figure 17Kp rules

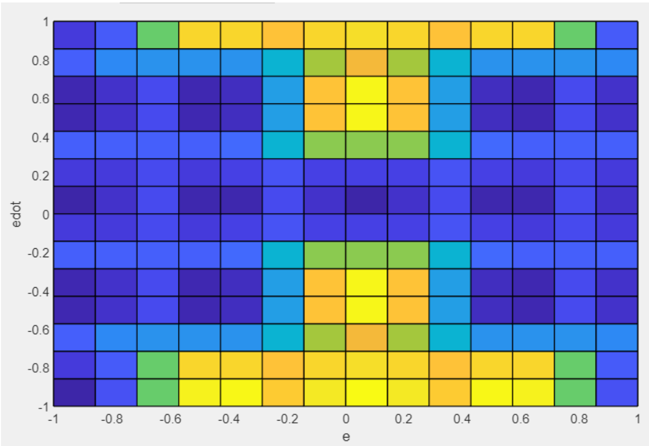

Figure 18 Kd rules

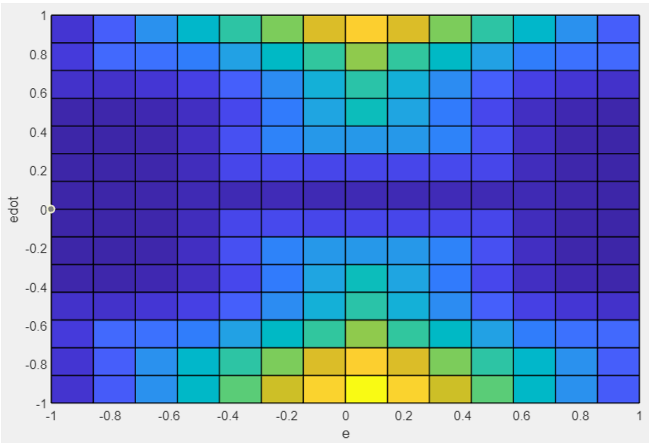

Figure 19 alpha rules

In the simulation, the responses for this controller were unstable and could not track the trajectory. A sample of its performance is presented in the below figure.

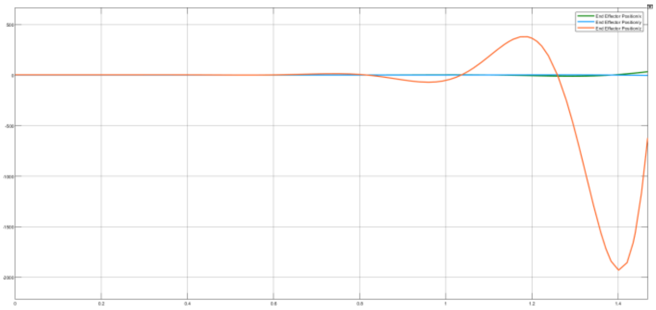

Figure 20 Performance of Fuzzy-PID controller