Readme Document for Fuzzy PID Controller in MATLAB

Introduction

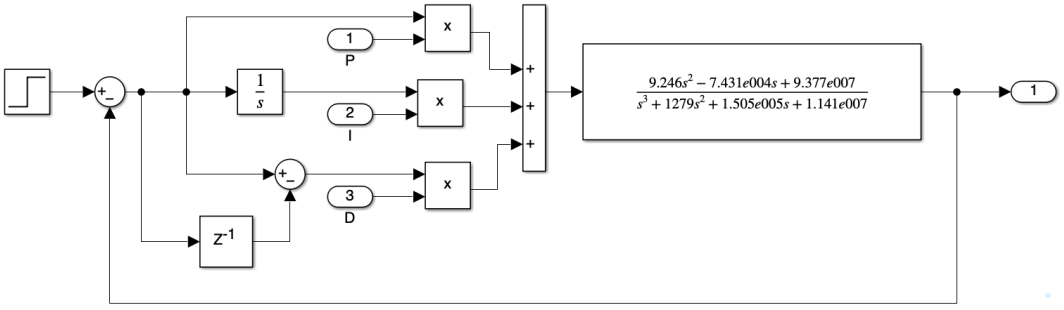
This document provides instructions for implementing a fuzzy PID controller in MATLAB using Simulink. A fuzzy PID controller is a popular control technique that combines fuzzy logic and PID control, offering robustness and effective handling of nonlinear and complex systems.

Implementation Steps

Follow the steps below to implement a fuzzy PID controller in MATLAB using Simulink:

Step 1: Create a Simulink model for a PID Control

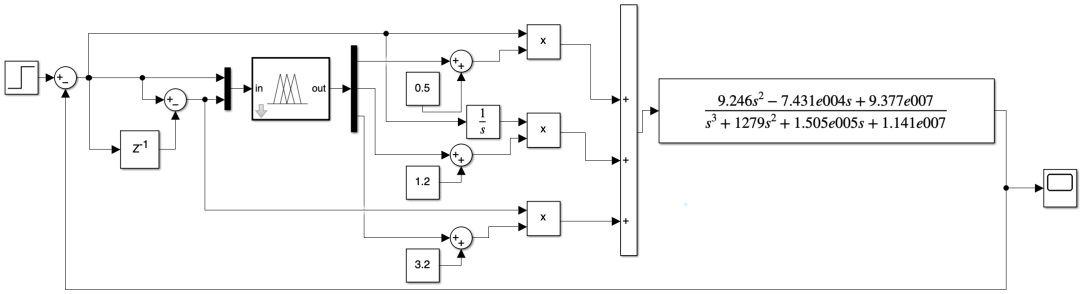
1. Open MATLAB and enter the command `simulink`.
2. In the opened Simulink model window, navigate to the Simulink Library Browser.
3. In the Simulink Library Browser, locate and expand the "Simulink" section.
4. Drag and drop the following modules into the model window: Step, Integral, Delay, Constant, Sum, Gain, Transfer Fcn, and Scope.
5. Connect the modules in the following order to build a parallel PID simulation model.



PID Controller Model

Step 2: Add Fuzzy Logic module to the PID Control model

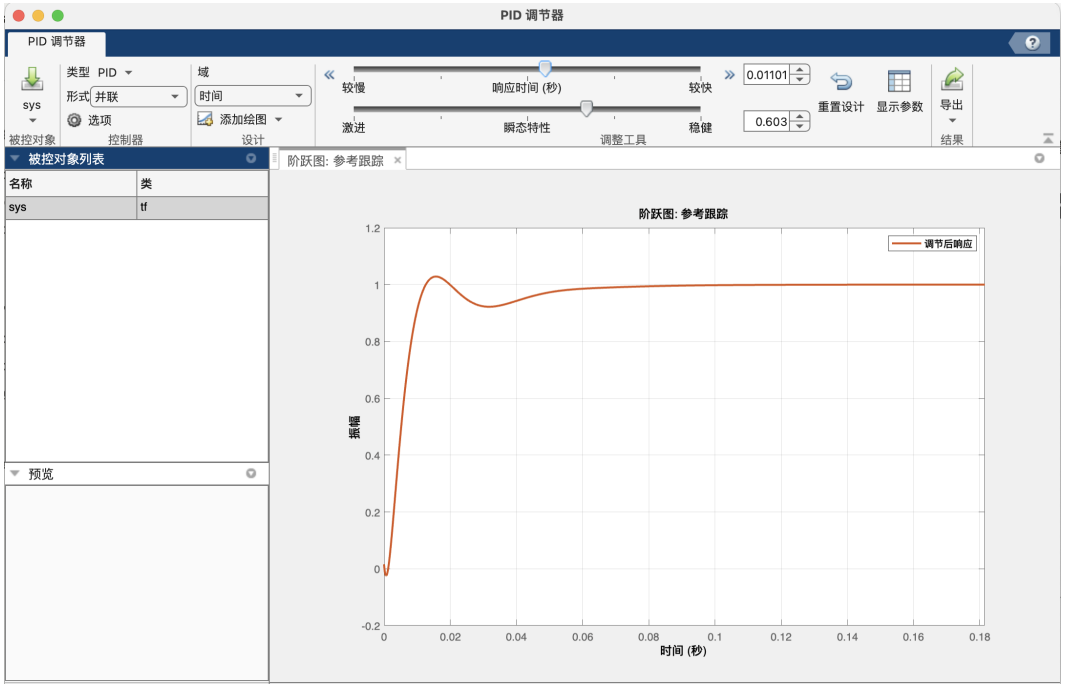
1. In the Simulink Library, find the Fuzzy Logic module and insert it into the PID Control model.
2. Use the ux module to connect e, ec as inputs and p, i, d parameters as outputs, creating the basic model of the Fuzzy PID Controller.



Fuzzy PID Controller

Step 3: Tune the basic PID parameters

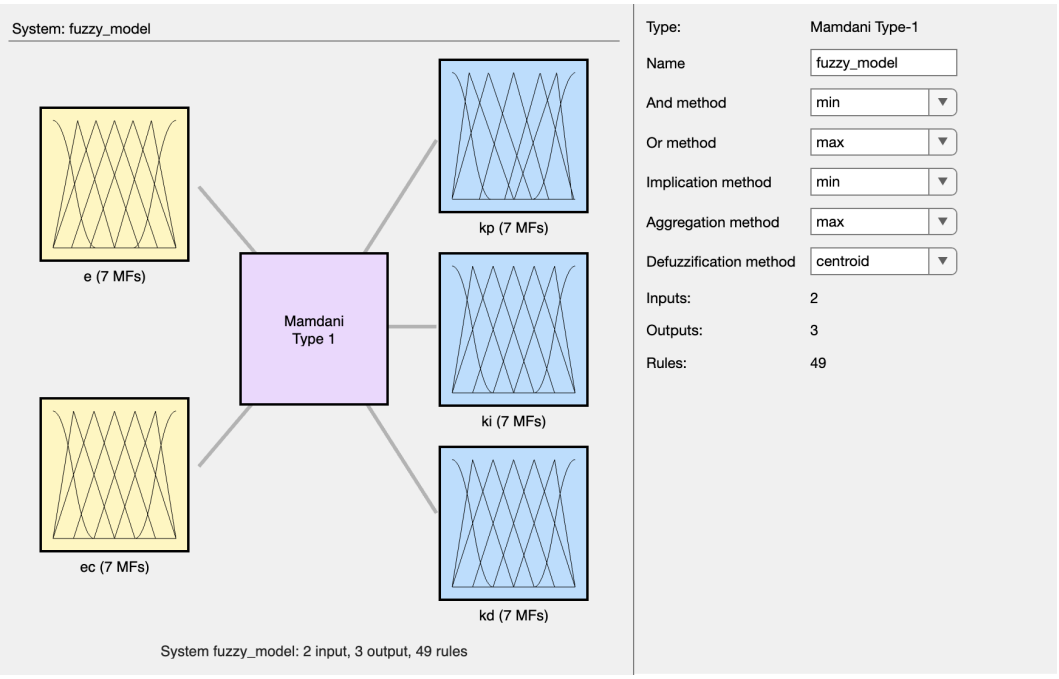
1. Open the PID Tuner App in MATLAB.
2. Import the system model to be controlled.
3. The PID Tuner will provide reference values for Kp, Ki, and Kd. When Kp=0.5, Ki=12, and Kd=3.2, the system can achieve a good response.



PID Tuner APP

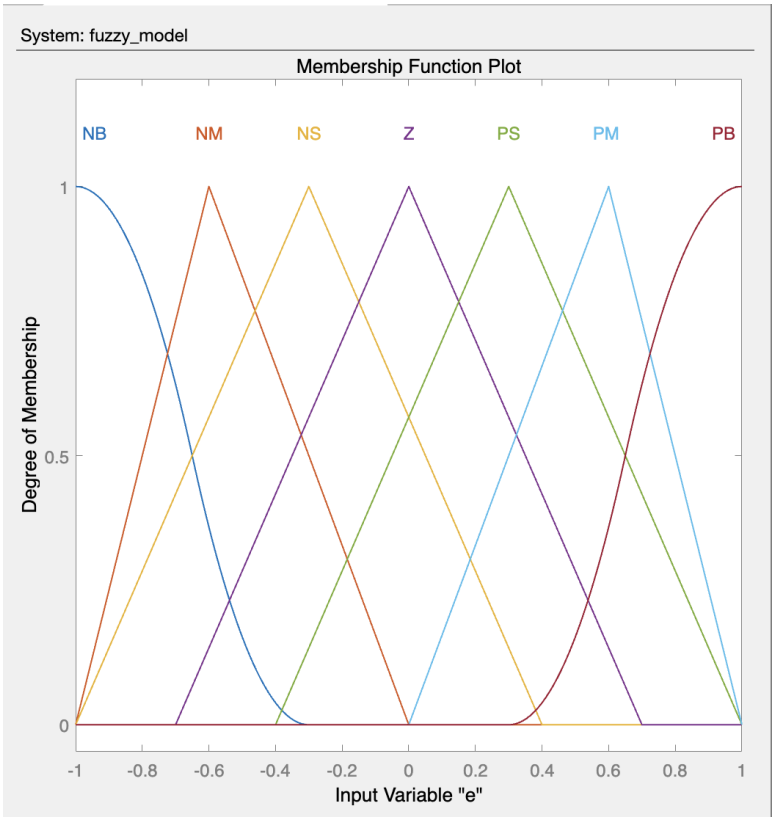
Step 4: Create a Fuzzy Logic Controller object

1. Open the Fuzzy Logic Designer App in MATLAB and create a new Fuzzy Inference System (FIS) object.



fuzzy logic model

2. In the "Inputs" and "Output" tabs, add input variables e, ec, and output variables Kp, Ki, and Kd.
3. Specify the range for input and output variables based on the PID parameter values and error range.
4. Select appropriate fuzzy subsets for the variables. To achieve more precise fuzzy values, select seven linguistic variables: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), and Positive Big (PB).
5. Choose suitable membership functions for the variables. In this case, all variables select "S-shaped," "Z-shaped," and "Triangular" as the membership functions.



Membership Function of “e”

Step 5: Define fuzzy rules

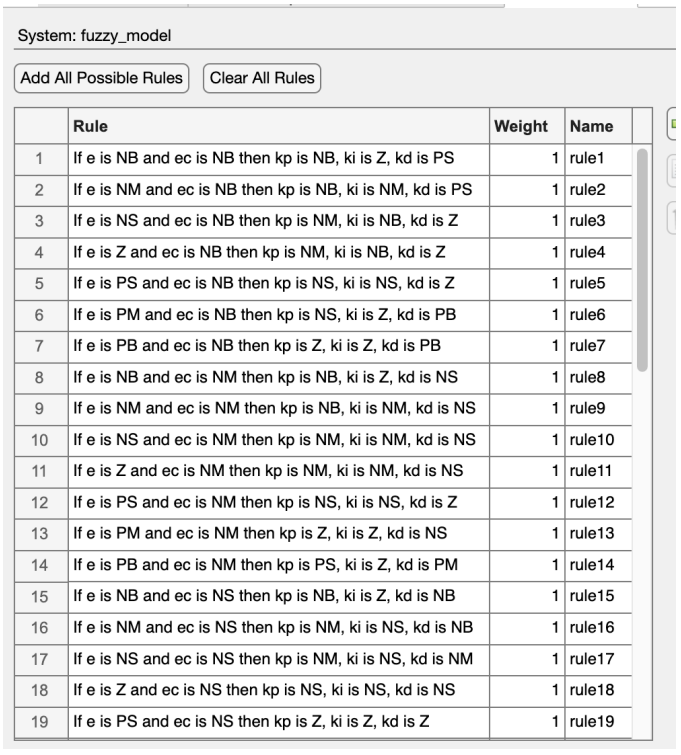
1. In the "Rules" tab, define fuzzy rules to describe the relationship between input and output variables.
2. Based on the Expert knowledge about PID gain tuning, establish fuzzy rules as follows:
    a. When |e| is large, regardless of the error change trend, the controller's output should be maximized to quickly adjust the error and reduce its absolute value at the maximum speed. To prevent integral saturation, larger Kp, smaller Ki, and Kd values should be used.
    b. When e * ec (error change) > 0, it indicates that the error is increasing in the direction of increasing absolute value.
        i. If the error is large, a stronger control action should be implemented by the controller to reverse the change in error towards a decrease in absolute value and quickly reduce the absolute value. In this case, larger Kp should be used, Kd should not be too large, and a smaller Ki value should be used.

ii. If the absolute value of the error is small, the controller should implement a general control action to only reverse the trend of the error and make it decrease.

c. When e * ec < 0 or e = 0, it indicates that the absolute value of the error is decreasing or has reached a balance state. In this case, the controller's output should remain unchanged.

d. When e * ec = 0 and e ≠ 0, it indicates that the system's curve is parallel or consistent with the theoretical curve. To ensure good steady-state performance and avoid oscillation near the setpoint, larger Kp and Ki values should be used, while considering the system's disturbance resistance, an appropriate Kd value should be selected.

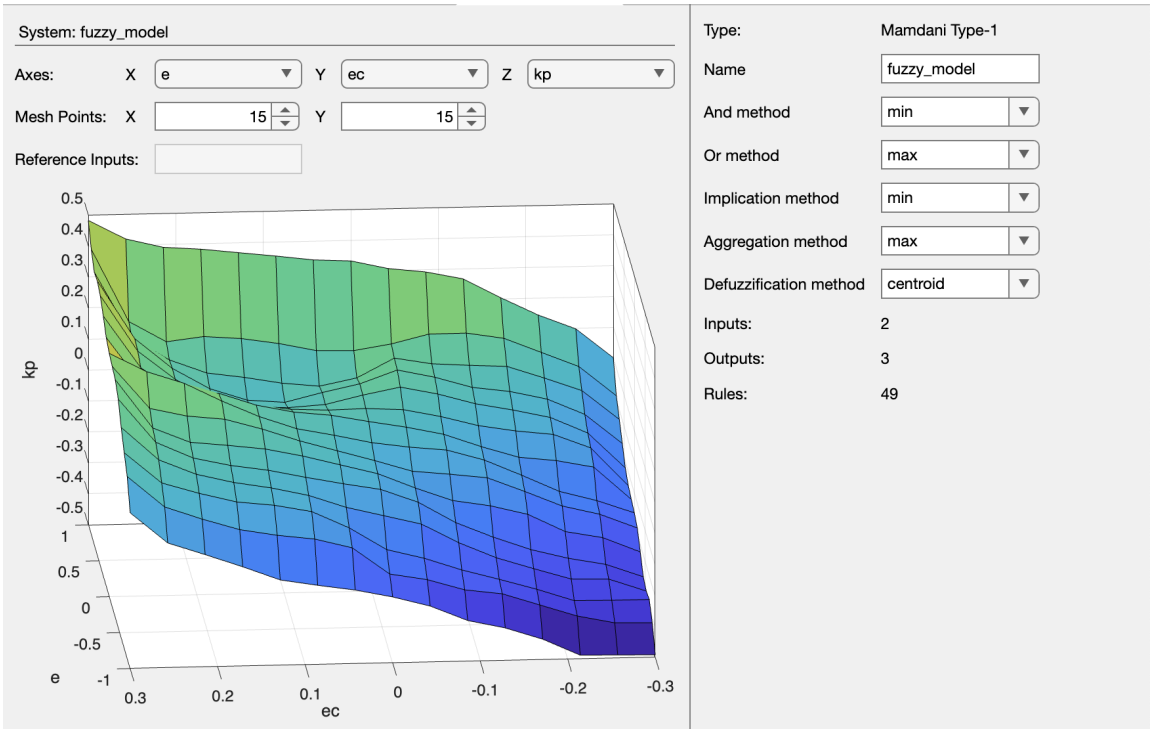
3. Enter the above Expert Knowledge fuzzy rules into the Rule Editor of the Fuzzy Logic Designer App.



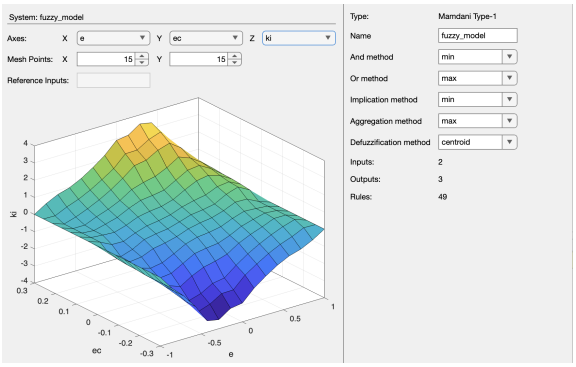
	Rule	Weight	Name
1	If e is NB and ec is NB then kp is NB, ki is Z, kd is PS	1	rule1
2	If e is NM and ec is NB then kp is NB, ki is NM, kd is PS	1	rule2
3	If e is NS and ec is NB then kp is NM, ki is NB, kd is Z	1	rule3
4	If e is Z and ec is NB then kp is NM, ki is NB, kd is Z	1	rule4
5	If e is PS and ec is NB then kp is NS, ki is NS, kd is Z	1	rule5
6	If e is PM and ec is NB then kp is NS, ki is Z, kd is PB	1	rule6
7	If e is PB and ec is NB then kp is Z, ki is Z, kd is PB	1	rule7
8	If e is NB and ec is NM then kp is NB, ki is Z, kd is NS	1	rule8
9	If e is NM and ec is NM then kp is NB, ki is NM, kd is NS	1	rule9
10	If e is NS and ec is NM then kp is NM, ki is NM, kd is NS	1	rule10
11	If e is Z and ec is NM then kp is NM, ki is NM, kd is NS	1	rule11
12	If e is PS and ec is NM then kp is NS, ki is NS, kd is Z	1	rule12
13	If e is PM and ec is NM then kp is Z, ki is Z, kd is NS	1	rule13
14	If e is PB and ec is NM then kp is PS, ki is Z, kd is PM	1	rule14
15	If e is NB and ec is NS then kp is NB, ki is Z, kd is NB	1	rule15
16	If e is NM and ec is NS then kp is NM, ki is NS, kd is NB	1	rule16
17	If e is NS and ec is NS then kp is NM, ki is NS, kd is NM	1	rule17
18	If e is Z and ec is NS then kp is NS, ki is NS, kd is NS	1	rule18
19	If e is PS and ec is NS then kp is Z, ki is Z, kd is Z	1	rule19

fuzzy rules

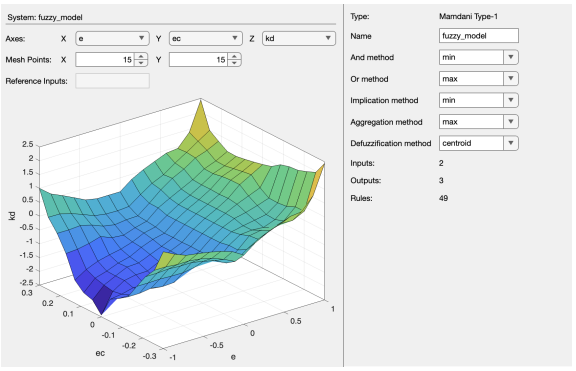
4. After define the fuzzy rule of this model, we can get the control surface of the output variables.



Kp



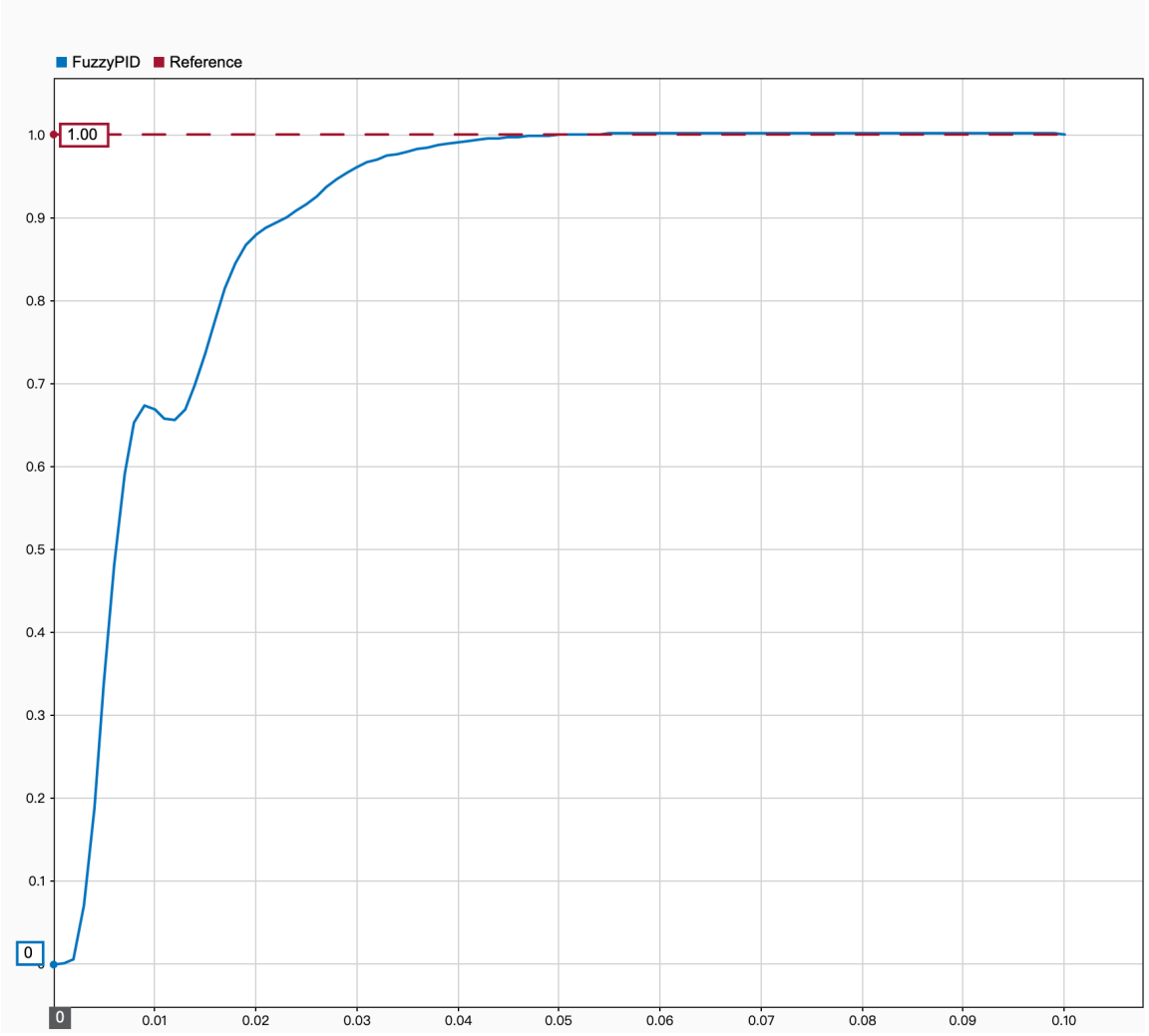
Ki



Kd

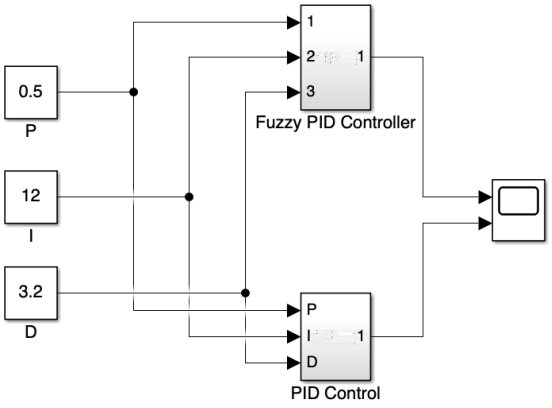
Step 6: Analyze the simulation results of the Fuzzy PID Controller

1. Apply the “fis” file to the fuzzy logic model, run the simulation. It can be get a new response result of system.

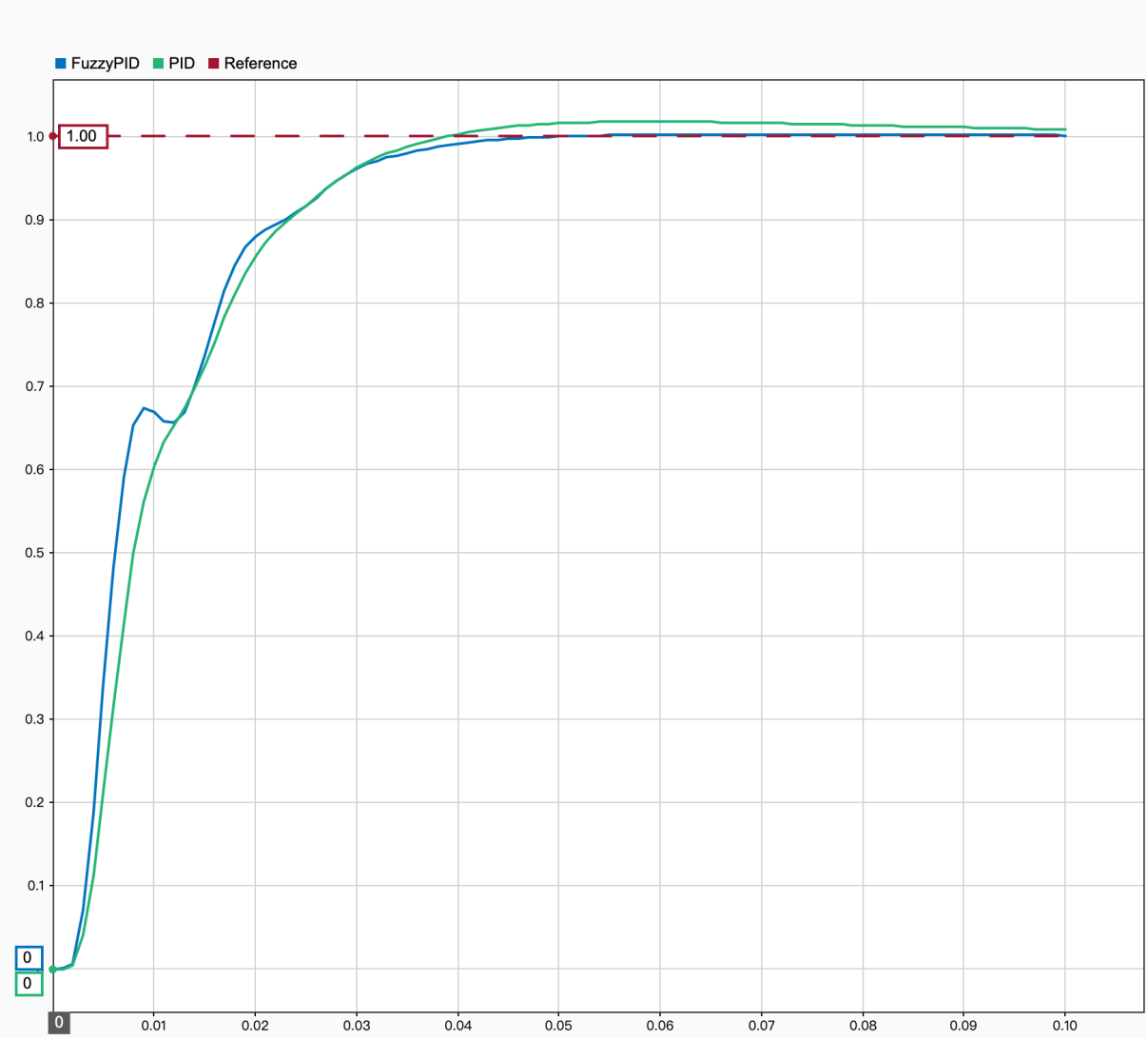


Fuzzy PID Controller's Response

2. Connect the outputs of the PID Controller and Fuzzy PID Controller to the Scope module.



3. Compare the performance of both controllers. It can be observed that the Fuzzy PID Controller has better response time and steady-state error compared to traditional PID control.



Conclusion

By following the steps in this document, you can implement a fuzzy PID controller in MATLAB using Simulink. The fuzzy PID controller combines the advantages of fuzzy logic and PID control, providing more effective control for systems. Try using different fuzzy logic configurations and PID controller parameters to achieve the desired control performance.