

آزمایش دهم: Fuzzy PID Controller

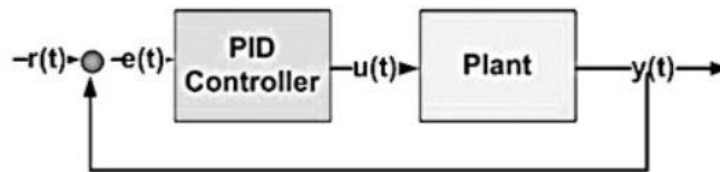
هدف آزمایش: کنترلر PID فازی

شرح آزمایش:

در این آزمایش قصد داریم عملکرد کنترلر PID را توسط یک کنترلر فازی شبیه سازی کنیم. در این قسمت ابتدا به مروری کوتاه بر ساختار کنترلر PID پرداخته می شود. سپس ساختار کنترلر فازی برای شبیه سازی عملکرد کنترلر PID بررسی می شود.

کنترل کننده PID یک کنترل کننده ی قدرتمند و در عین حال ساده، برای سیستم های کنترلی می باشد که تابع تبدیل و خروجی آن به صورت معادلات ۱.۱۲ است. ساختار کلی کنترلر در شکل یک نمایش داده شده است. ورودی این کنترلر خطا و خروجی آن ورودی سامانه است.

$$\begin{aligned} G(s) &= K_p + \frac{K_i}{s} + K_d s \\ u(t) &= K_p [e(t) + \frac{1}{T_i} \int_0^t e(r) dr + T_d \dot{e}] \\ T_i &= \frac{K_p}{K_i} \\ T_d &= \frac{K_d}{K_p} \end{aligned} \quad (1.12)$$



شکل یک، ساختار کلی PID

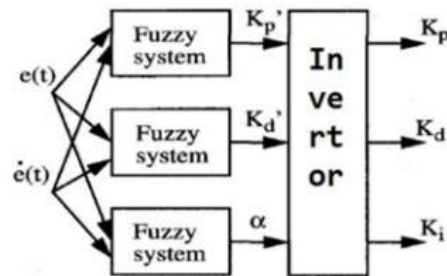
کنترلر PID فازی

برای طراحی کنترلر PID فازی ورودی های کنترلر به صورت خطا و مشتق خطا در نظر گرفته می شود. خروجی کنترلر نیز ضرایب K_p' و K_d' و α که به صوت زیر تعریف می شوند در نظر گرفته می شود. در نهایت بعد از به دست آوردن ضرایب توسط یک بلوک معکوس کننده ضرایب K_p' و K_d' و α به ضرایب K_p ، K_d و K_i تبدیل می شوند. ساختار ورودی و خروجی کنترلر فازی PID در شکل دو نمایش داده شده است.

$$K_{p'} = \frac{K_p - K_{pmin}}{K_{pmax} - K_{pmin}}$$

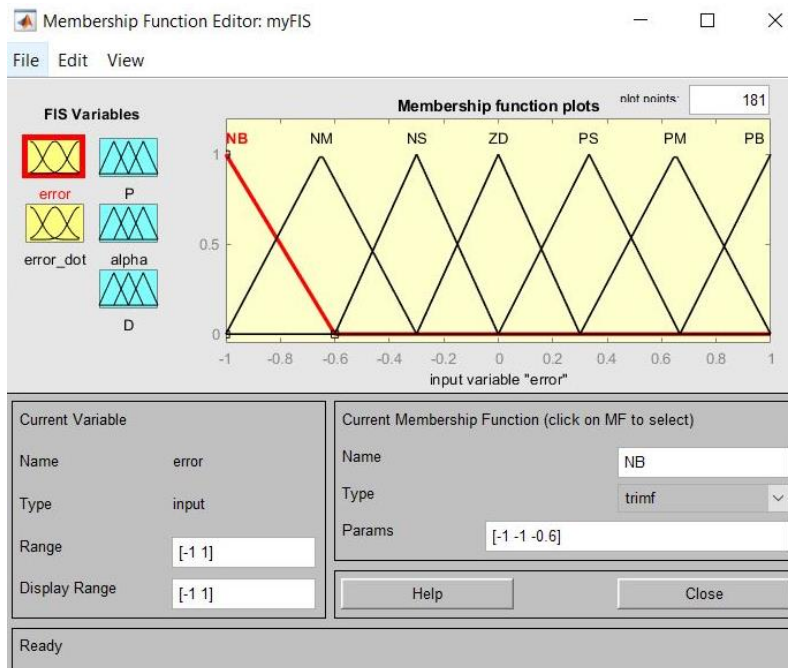
$$K_{d'} = \frac{K_d - K_{dmin}}{K_{dmax} - K_{dmin}} \quad (2.12)$$

$$T_i = \alpha T_d$$

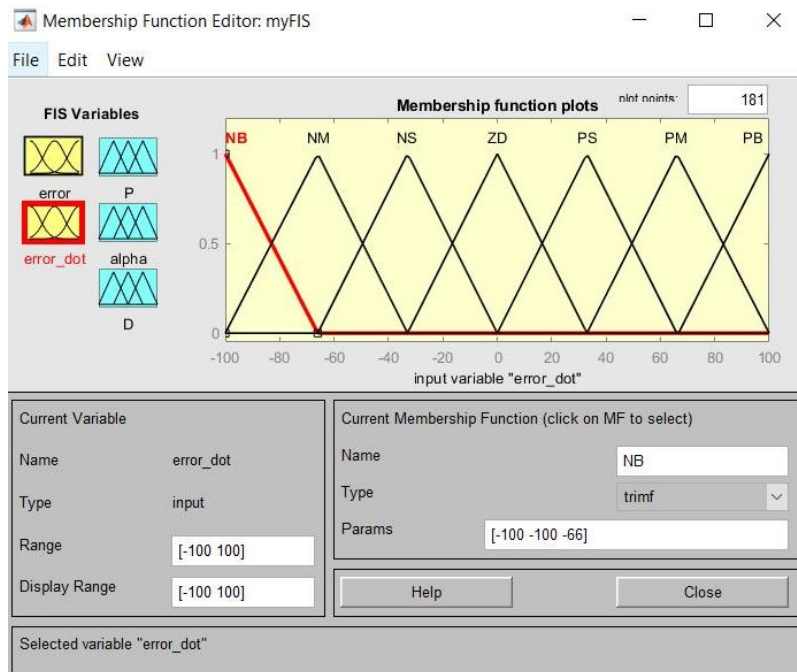


شکل دو، رابطه بین ورودی و خروجی های کنترلر PID فازی

ابتدا Fuzzy Logic Designer را باز میکنیم و ورودی ها و خروجی ها را به صورت زیر را تنظیم می کنیم.

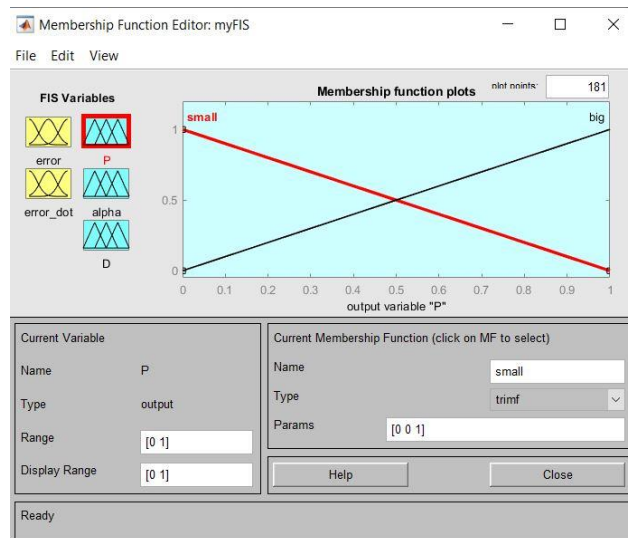


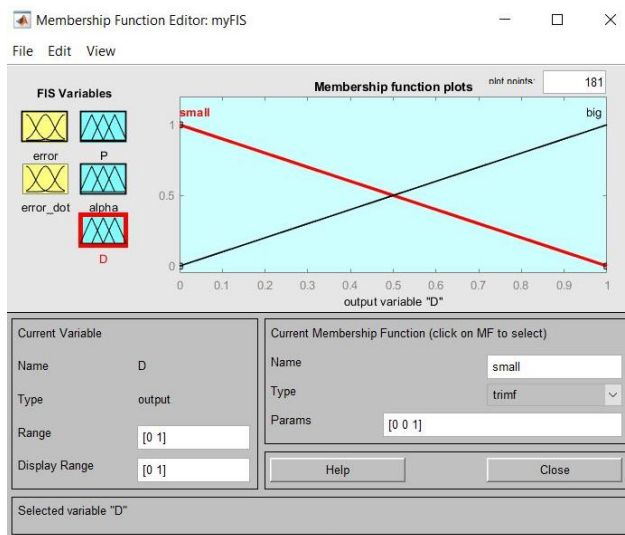
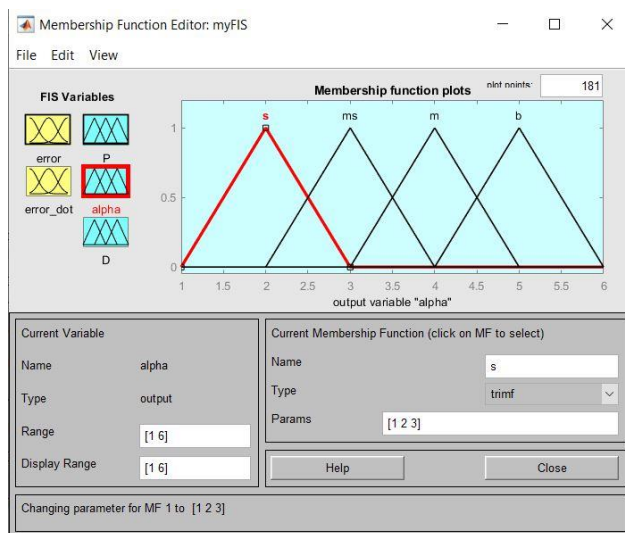
شکل سه، تنظیم ورودی های فازی



شکل چهار، تنظیم ورودی های فازی

سپس خروجی فازی را به صورت زیر تنظیم میکنیم. سه خروجی داریم که P و alpha و D است.





شکل پنچ، تنظیم خروجی فازی

بعد از مشخص شدن توابع عضویت برای متغیرهای مختلف سیستم فازی برای کامل شدن ساختار باید به تعریف قوانین پرداخته شود. قوانین سیستم فازی را می توان به گونه های مختلفی تعریف کرد. یک نمونه از قوانین برای K_p و α به ترتیب در شکل های شش، هفت و هشت نمایش داده شده است. با تعریف شدن قوانین و مشخص شدن توابع عضویت با انتخاب یک فازی ساز و یک ضد فازی ساز طراحی کنترلر فازی PID پایان می پذیرد.

		$\dot{e}(t)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

شکل شش، قوانین K_p'

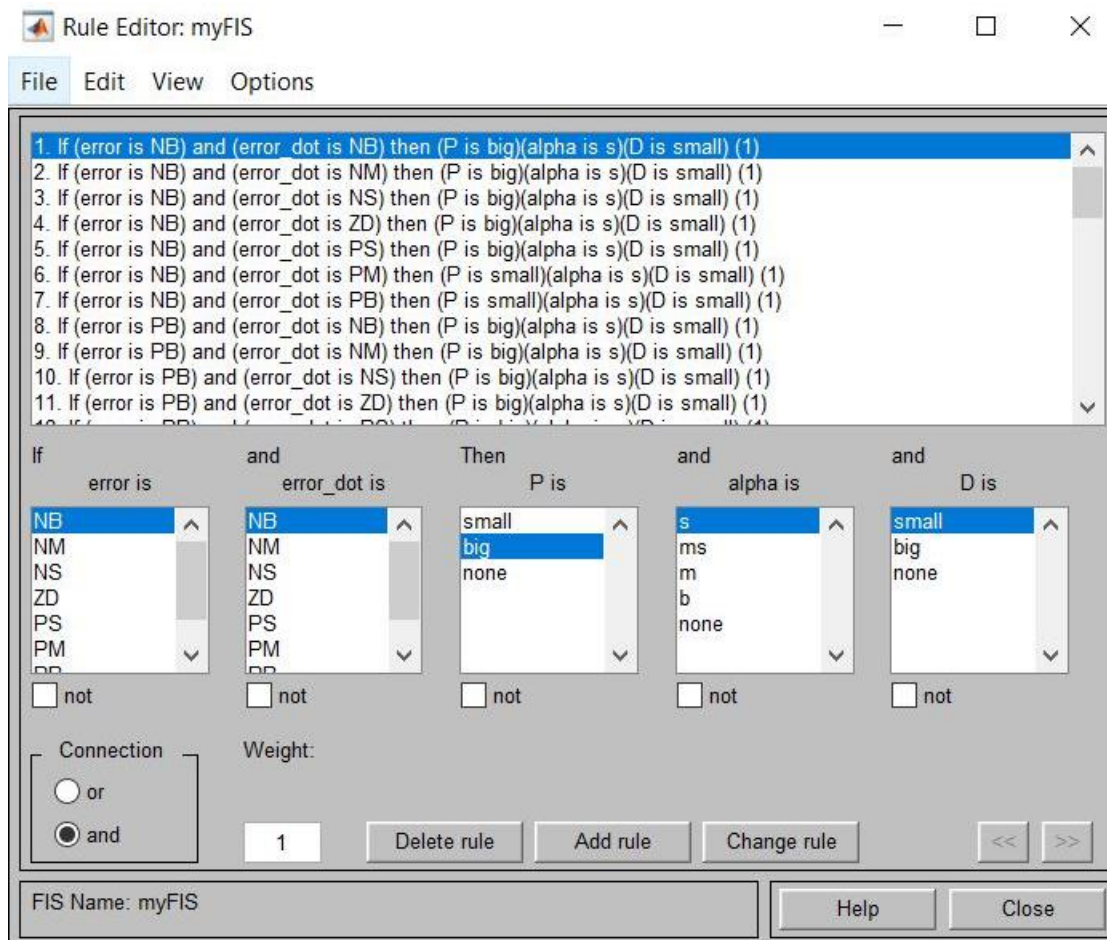
		$\dot{e}(t)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

شکل هفت، قوانین K_d'

		$\dot{e}(t)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	2	2	2	2	2	2	2
	NM	3	3	2	2	2	3	3
	NS	4	3	3	2	3	3	4
	ZO	5	4	3	3	3	4	5
	PS	4	3	3	2	3	3	4
	PM	3	3	2	2	2	3	3
	PB	2	2	2	2	2	2	2

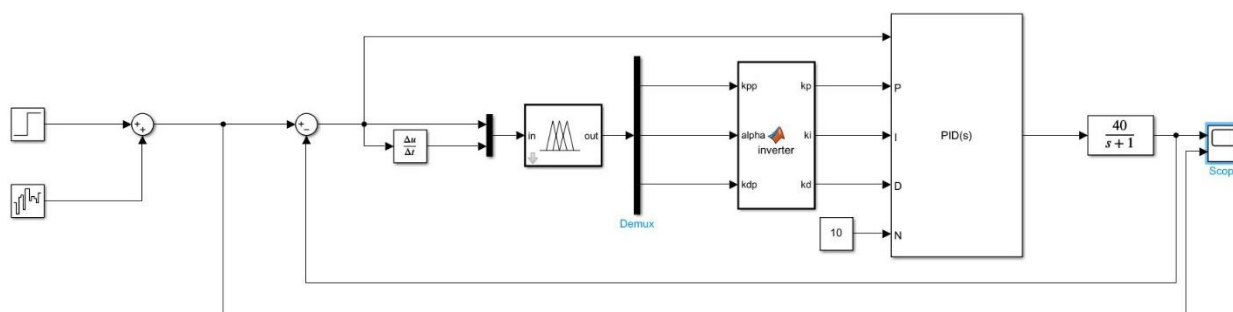
شکل هشت، قوانین α

در قسمت ممدانی، قوانین IF, THEN را تعریف میکنیم که به صورت زیر است.



شکل نه، قوانین FIS

حال به سیمولینک انتقال می دهیم.



شکل ده، سیمولینک

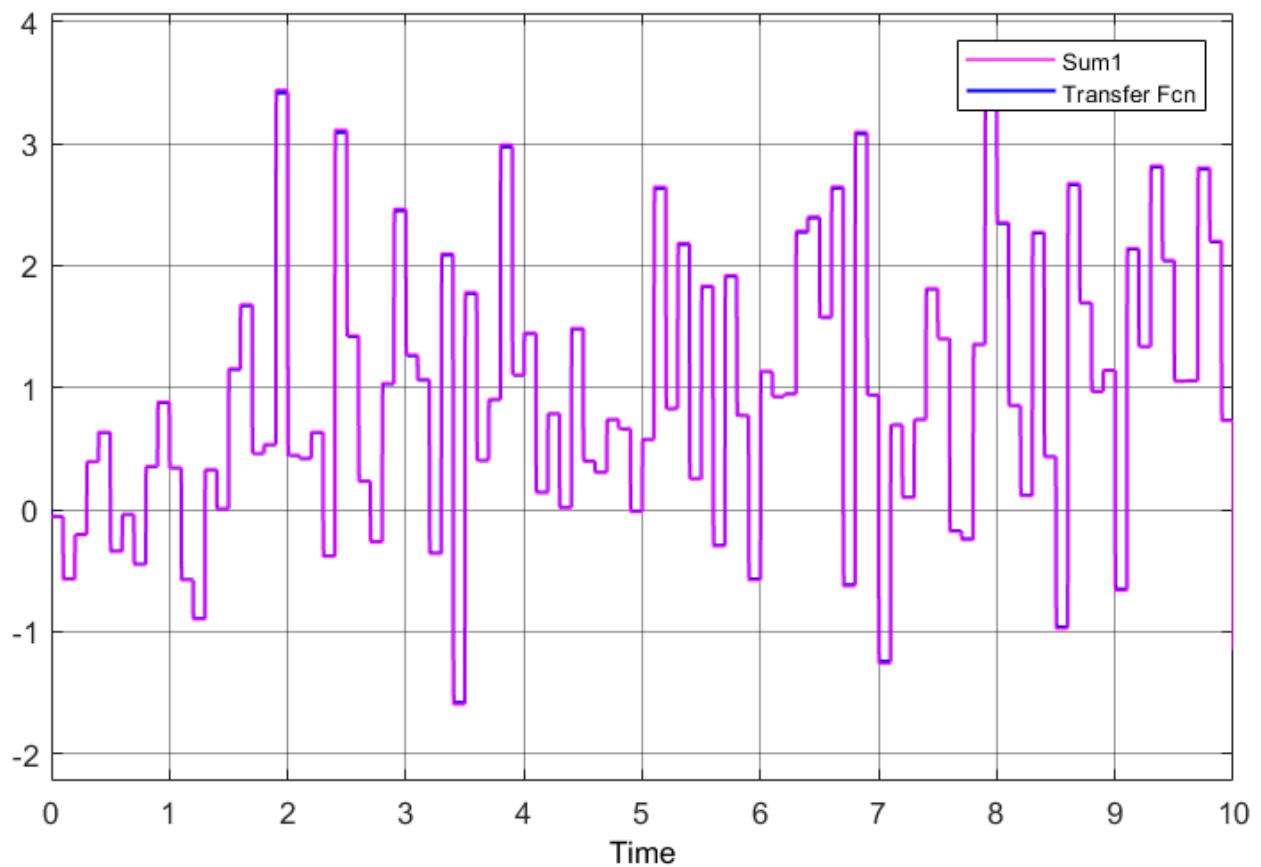
سورس PID را به صورت external قرار می دهیم. ورودی های آن را از تابع ساخته شده زیر دریافت می کنیم. علاوه بر آن درجه فیلتر را ده قرار داده و ورودی باقی مانده آن error است.

```
untitled ▶ MATLAB Function
1 function [kp,ki,kd] = inverter(kpp,alpha,kdp)
2   kp = 5*kpp;
3   kd = 5*kdp;
4   ki = (kp*kp)/(alpha*kd);
5   end
6
```

شکل یازده، تابع inverter

خروجی PID را از plant داده شده در دستورکار عبور می دهیم.

خروجی شبیه سازی به صورت زیر می باشد.



شکل دوازده، خروجی سیمولینک