



بسمه تعالی



ازمایشگاه هوش محاسباتی

موضوع : Report Fuzzy PID

دانشکده مهندسی برق
دانشگاه صنعتی امیر کبیر

عباس علی کوچک زاده

بهمن ۱۳۹۷

به نام خدا

۱۳۹۷/۱۰/۲

گزارش آزمایش هفتم هوش محاسباتی

عنوان آزمایش: کنترل کننده PID به صورت فازی

اعضای گروه: علی گواهی - علی شجاعی - امیرحسین زمانی (نویسنده)

پیاده سازی کنترلر PID به صورت فازی در متلب:

دقیقا همانند مراحل گفته شده در Lecture درس و دستورکار، به سادگی با Fuzzy toolbox متلب به صورت زیر پیاده می شود.

ورودی های سیستم PID، خطا و مشتق خطاست و قرار است برای بدست آوردن ضرایب K_p ، K_i و K_d ، سه موتور فازی قرار داد تا هر کدام از این ضرایب را با قوانین فازی ای به صورت زیر پیاده سازی کند. اما در شکل های ۱ تا ۳ قوانین فازی k_p' و k_d' و α داده شده است.

		$\dot{e}(t)$						
		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

		$\dot{e}(t)$						
		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

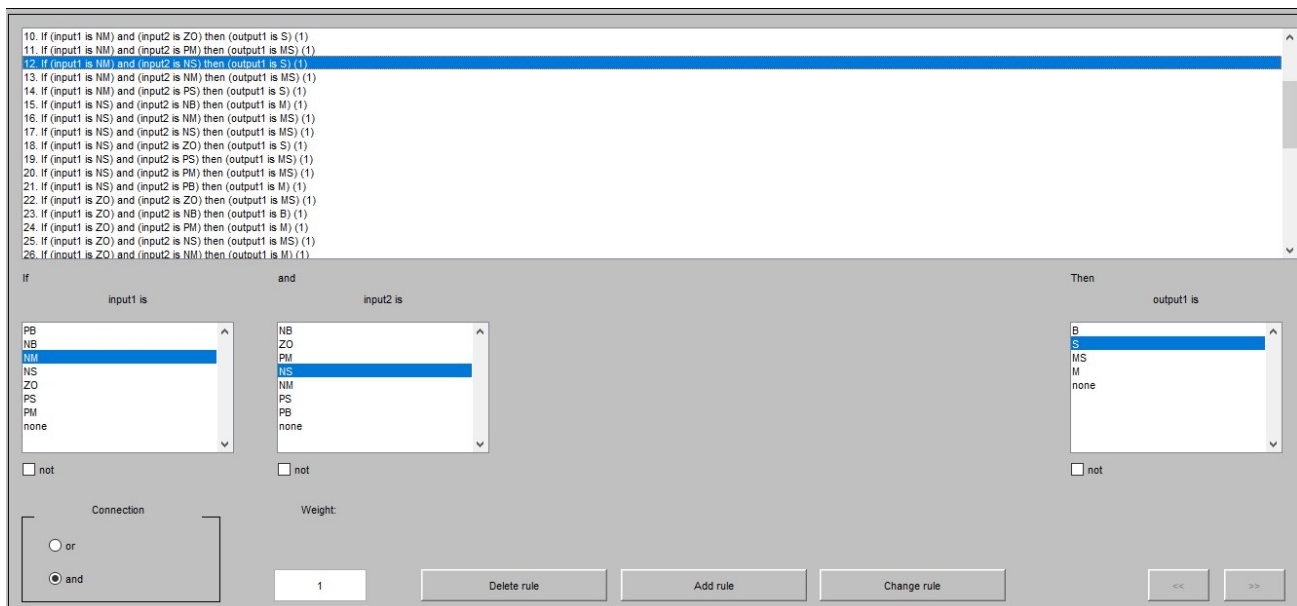
		$\dot{e}(t)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(t)$	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

شکل ۳- قوانین فازی α

شکل ۲- قوانین فازی k_d'

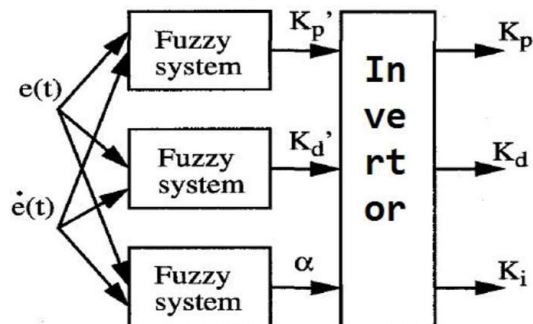
شکل ۱- قوانین فازی k_p'

برای نمونه قوانین پیاده سازی شده در فازی متلب برای پارامتر آلفا به صورت شکل زیر است. بقیه قوانین هم دقیقاً همانند جداول بالا پیاده سازی شدند.



شکل ۴- قوانین فازی در متلب برای پارامتر آلفا

در نتیجه برای تبدیل سه پارامتر k_p' و k_d' و α به k_p و k_d و k_i از یک Invertor که درون خود روابط زیر را دارد، استفاده می‌کنیم.

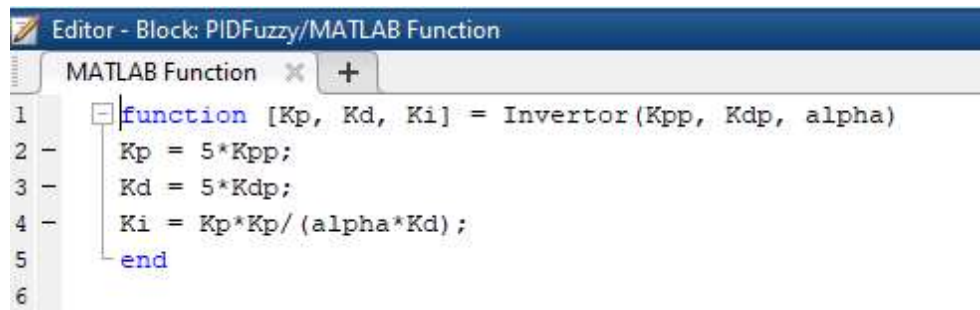


شکل ۵- بلوک دیاگرام PID فازی

$$\begin{aligned} \triangleright K_{p'} &= \frac{K_p - K_{pmin}}{K_{pmax} - K_{pmin}} \\ \triangleright K_{d'} &= \frac{K_d - K_{dmin}}{K_{dmax} - K_{dmin}} \end{aligned}$$

$$\text{Assume } T_i = \alpha T_d \rightsquigarrow K_i = \frac{K_p}{\alpha T_d} = \frac{K_p^2}{\alpha K_d}$$

بلوک Invertor در شکل ۴ را به صورت یک Matlab Function داریم.



```

Editor - Block: PIDFuzzy/MATLAB Function
MATLAB Function  X  +
1  function [Kp, Kd, Ki] = Invertor(Kpp, Kdp, alpha)
2  -      Kp = 5*Kpp;
3  -      Kd = 5*Kdp;
4  -      Ki = Kp*Kp/(alpha*Kd);
5  -      end
6

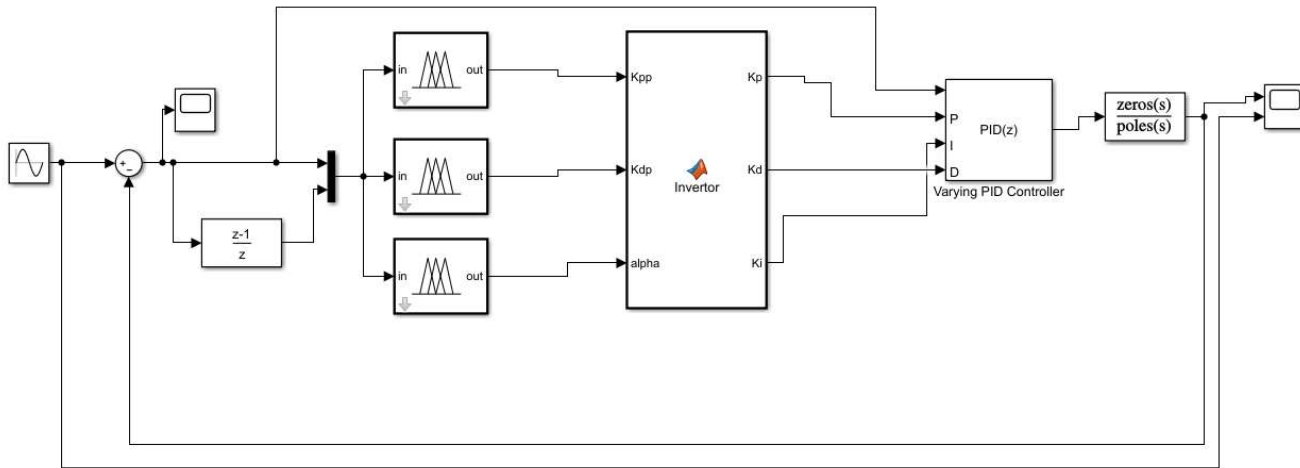
```

شکل ۶- کد مربوط به Invertor

حال ضرایب بدست آمده از سیستم فازی و Invertor را به یک بلوک PID در متلب می‌دهیم و بعد از آن هم Plant را قرار می‌دهیم که در این آزمایش به شکل زیر است:

$$\frac{27}{s^3 + 7s^2 + 15s + 9}$$

بلوک دیاگرام نهایی در سیمولینک به صورت زیر است:



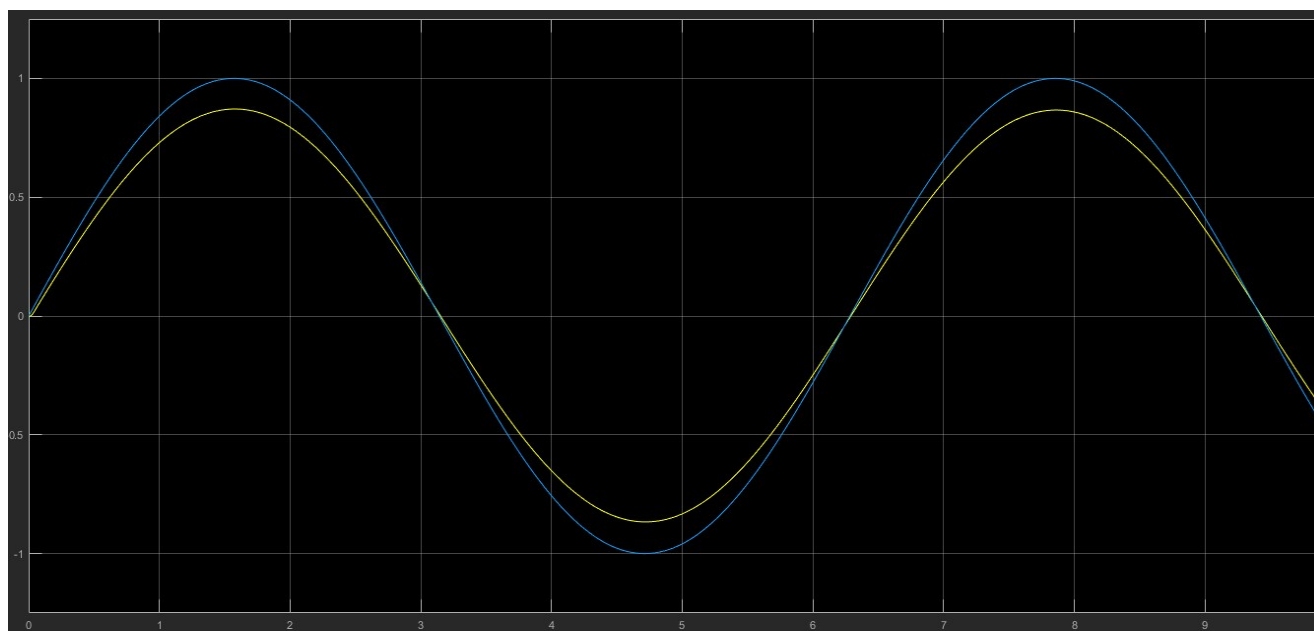
شکل ۷- بلوک دیاگرام کلی سیستم در سیمولینک

Membership function های هر پارامتر نیز دقیقاً همانند اطلاعات موجود در دستور کار پیاده سازی شد و برای نمونه، تابع عضویت خطا و مشتق آن در زیر آمده است:



شکل ۸- تابع عضویت خطای ورودی و مشتق آن

در نهایت خروجی کنترلر PID که به صورت فازی پیاده سازی شده است، با ورودی مقایسه می‌شود. به عنوان یک ورودی سینوسی، نمودار آبی در شکل زیر ورودی است و نمودار زرد هم خروجی سیستم است. همانطور که مشاهده می‌شود خروجی، ورودی را با خطای نسبتاً قابل قبولی دنبال می‌کند. هرچند که واضح است که می‌توان با دقیقتر کردن نواحی توابع عضویت و حتی زیاد کردن آن دقت سیستم را بالا برد. شکل زیر شبیه‌سازی سیستم برای مدت ۱۰ ثانیه است.



شکل ۹- مقایسه ورودی و خروجی بلوک دیاگرام شکل ۷ با ورودی سینوسی با دامنه واحد