



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی برق

گزارش پروژه درس هوش محاسباتی
گرایش کنترل

عنوان
کنترل خودروی هوشمند به روش فازی

نگارش
الناز فیروزمند
ناهید اعتصامی
پریسا داج خوش
احسان عزیزی

استاد راهنما
سرکار خانم دکتر عبداللهی

زمستان ۱۳۹۵

1. مقدمه

برای کنترل اتوماتیک خودرو روش های متفاوتی وجود دارد که یکی از آن ها استفاده از کنترلر های منطق فازی است. یکی از مزیت های این روش، توانایی آن در تلفیق تجربه و دانش انسانی برای ایجاد روابط کنترلر بین پارامتر هاست.

خودرو های بدون سرنشین کاربرد های فراوانی دارند و می توان از آنها برای افزایش امنیت ناوبری، کاهش ترافیک و جایگزینی انسان در در موقعیت های خطرناک، مقاصد نظامی و محیط هایی که حضور انسان ممکن نیست (مثل کره ماه و ...) استفاده کرد.

تا کنون تحقیقات فراوانی در مورد خودرو های بدون سرنشین در دنیا انجام شده است و کنترلر های فازی با موفقیت در این زمینه به کار گرفته شده اند.

روند پیاده سازی پروژه به شرح زیر است.

با توجه به اینکه ایده اصلی فازی بر اساس تجربه و دانش انسان است. بر اساس تجربه راننده، برای هدایت فرمان و سرعت از ماژول های جداگانه ای کمک گرفتیم و برای هر یک اهدافی را تعیین کردیم. برای مثال برای بخش طراحی کنترلر زاویه هدایت فرمان به نکات زیر توجه کردیم:

۱. خودرو برای رسیدن به هدف مسیر بهینه ای را انتخاب کند.

۲. زمانی که به مانع رسید، بتواند آن را با زاویه مناسبی رد کند.

۳. بتواند با زاویه مطلوبی به هدف برسد.

به همین ترتیب برای طراحی بخش کنترلر سرعت نیز اهداف زیر را در نظر گرفتیم:

۱. خودرو زمانی که به مانع می رسد، بتواند سرعت خود را به اهستگی کاهش دهد.

۲. با رسیدن به مقصد توقف کند.

۳. وقتی در یک جاده با پیچ تند در حرکت است، بتواند سرعت مناسب خود را تنظیم کند.

بنابراین می توان با در نظر گرفتن موارد فوق، حرکت ایمن خودرو را بدون برخورد با مانع و هم چنین در کوتاه ترین مسیر تضمین نمود.

لذا برای کنترل زاویه فرمان و سرعت، به ترتیب از دو و سه ماژول جداگانه استفاده کردیم. به این ترتیب که دو ماژول کنترلر زاویه فرمان، وظیفه تنظیم زاویه فرمان برای رسیدن به هدف و عدم برخورد با مانع را بر عهده دارند که مجموع خروجی این دو کنترلر به عنوان خروجی نهایی زاویه فرمان استفاده شد.

در بخش کنترلر سرعت نیز همانطور که پیش تر به آن اشاره شد، از سه ماژول مجزا استفاده کردیم که به ترتیب وظیفه تنظیم زاویه پدال گاز را در سه حالت کلی به عهده داشتند که عبارتند از:

زمانی که به مقصد نزدیک می شویم، یا زمانی که یک مانع اعم از متحرک و یا ثابت، مقابل ما است، و همچنین زمانی که در مسیر یا پیچ تند و یا پیچ های جاده هستیم.

در نهایت مجموع خروجی کنترلر را برای هر یک از حالات زیر به عنوان تنظیم زاویه پدال گاز در نظر گرفته شدند.

در بخش دوم این پروژه کنترلی طراحی شد که میتواند یک شی متحرک را دنبال کند. که نتایج نهایی در محیط واقعیت مجازی پیاده سازی شدند.

2. منطق فازی

برای کنترل خودروی بدون سر نشین ابتدا کنترل خودرو را به دو بخش کنترل فرمان و کنترل پدال گاز (که رابطه ی مستقیم با نیرو و نیرو نیز رابطه ی مستقیم با شتاب خودرو دارد) تقسیم کردیم.

کنترلی که برای کنترل فرمان طراحی میشود میبایست اهداف زیر را محقق کند:

۱. وسیله را به سمت هدف هدایت کند.
۲. وقتی سنسور خودرو متوجه حضور مانعی شد طوری زاویه فرمان را تنظیم کند که خودرو از کنار مانع عبور کرده و با آن برخورد نکند.
۳. مسیر بهینه را طی کند.
۴. در نهایت خودرو را در هدف مشخص شده و با زاویه مشخص شده متوقف کند.

کنترلی که برای کنترل سرعت از طریق کنترل زاویه پدال گاز طراحی میشود میبایست اهداف زیر را محقق کند:

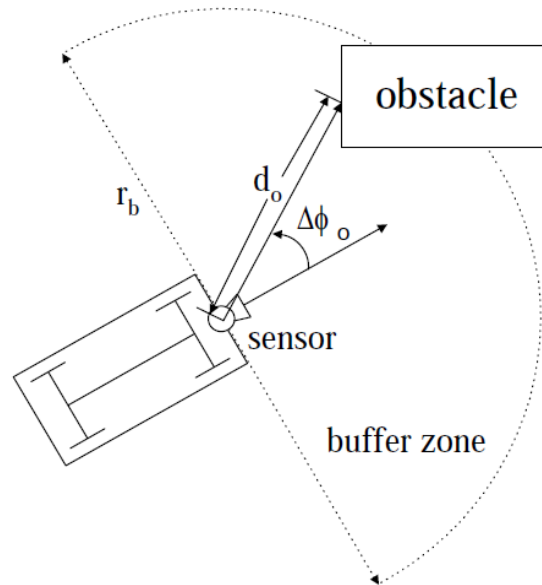
۱. خودرو از توقف کامل به راه بیفتد و بعد از رسیدن به هدف نیز توقف کامل کند.
۲. زمانی که خودرو به یک مانع برسد سرعت را کم کند و بعد از اینکه از مانع عبور کرد سرعت را مجدداً زیاد کند.
۳. وقتی خودرو به پیچ جاده میرسد سرعت را متناسب با شدت پیچ کم کند.

برای هر یک از اهداف ذکر شده یک ماژول کنترل فازی با منطق ممدانی طراحی شد. همچنین توابع عضویت ها و قوانینی که در این کنترلر ها به کار بردیم بر اساس حس کلی از رانندگی؛ و تعاریفی که از متغیر ها داریم، ساخته و تعریف شدند.

به عنوان فرضیات مسئله فرض شده که مختصات هدف تعیین شده و میخواهیم خودرو را به سمت هدف هدایت کنیم.

همچنین فرض شده که خودرو به یک سنسور سنجش نزدیکی اجسام (مثلا سنسور آلتراسونیک یا مجهز شده تا فاصله و جهت مانع را شناسایی کند. سنسور موانع داخل یک محدوده شعاعی مشخص (محدوده حائل) را تشخیص میدهد و فاصله خودرو تا موانع داخل این محدوده را میسنجد. شکل منطقه حائلی که توسط سنسور شناسایی میشود در شکل زیر نمایش داده شده است. سنسور مینیمم فاصله تا مانع را میسنجد اما تضمینی وجود ندارد که خروجی سنسور در اولین اندازه گیری واقعا کمترین فاصله تا مانع باشد به همین

خاطر نمونه برداری فرکانس بالا انجام می شود، به بیان دیگر تعداد نمونه های زیادی برداشته میشود تا اندازه گیری با صحت مناسبی انجام شود.



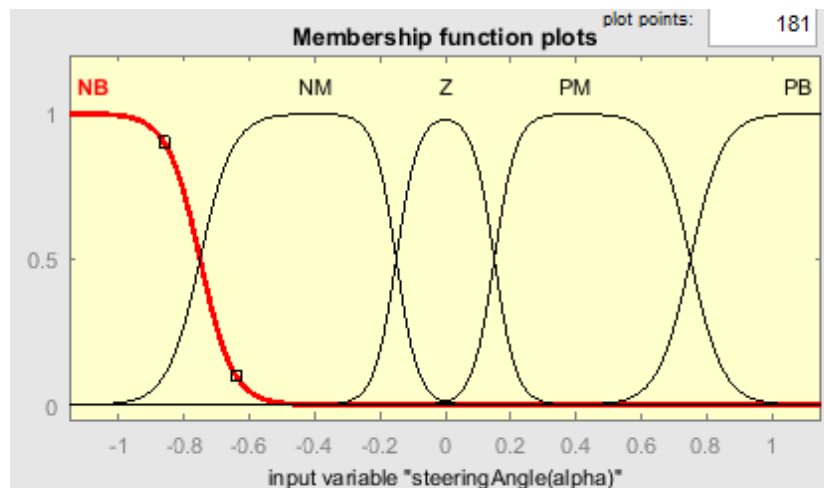
۳. کنترلر های فازی

کنترلر فرمان

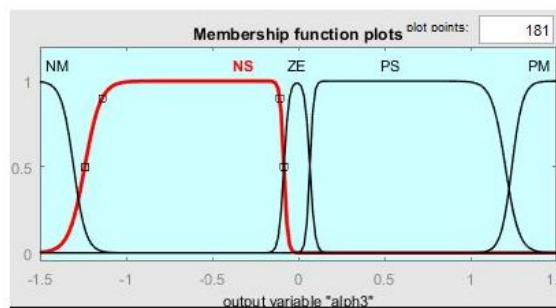
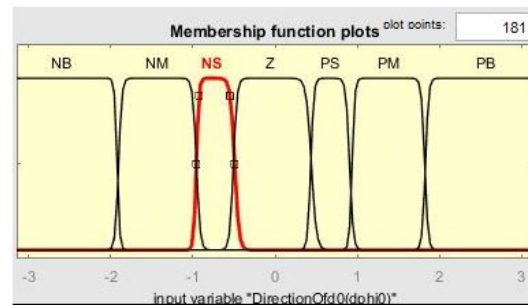
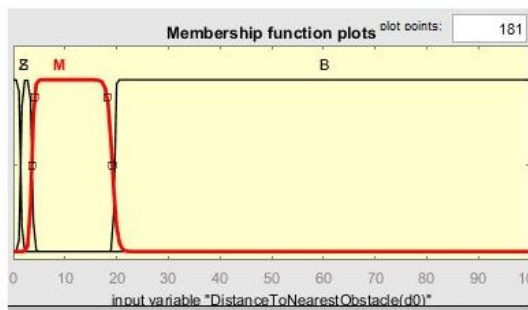
برای طراحی این کنترلر مسئله به چند بخش تقسیم شده و برای هر بخش یک ماژول طراحی شده است. مسئله ی اول هدایت فرمان به سمت هدف و مسئله دوم هدایت فرمان به گونه ایست که در صورت مواجهه با مانع، خودرو با آن برخورد نکند.

برای این ماژول، دو ورودی زاویه ی هدایت فرمان و زاویه به مقصد در نظر گرفته شده که به خروجی آن تغییرات زاویه ی هدایت است. برای هر یک پنج تابع عضویت تعریف شده است: ۱- بزرگ منفی (NB) ۲-

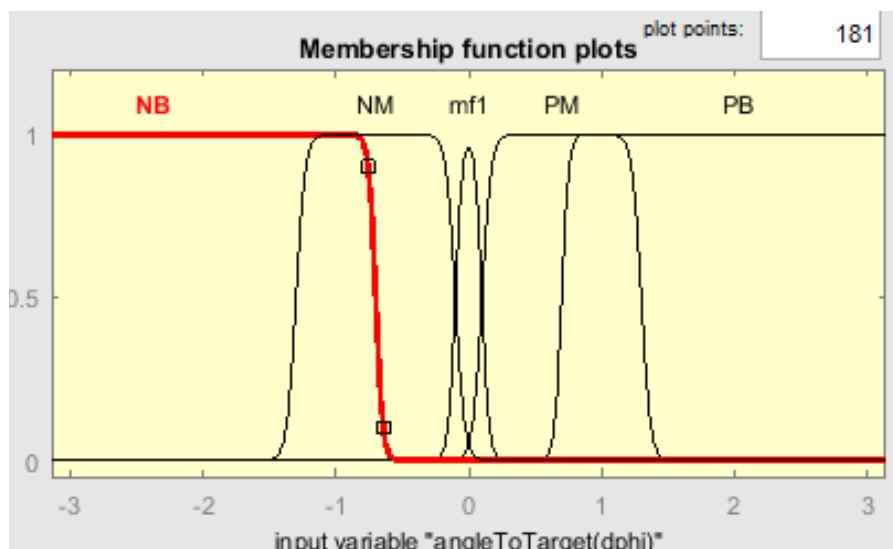
متوسط منفی (NM) ۳- صفر (Z) ۴- متوسط مثبت (PM) ۵- بزرگ مثبت (PB)



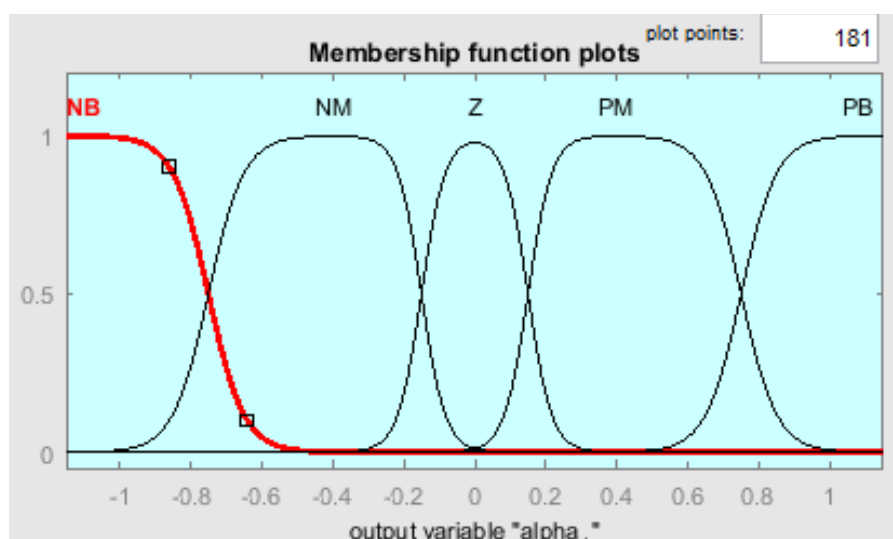
شکل ۱ ورودی زاویه ی هدایت فرمان



Active:



شکل ۲ ورودی زاویه نسبت به مقصد



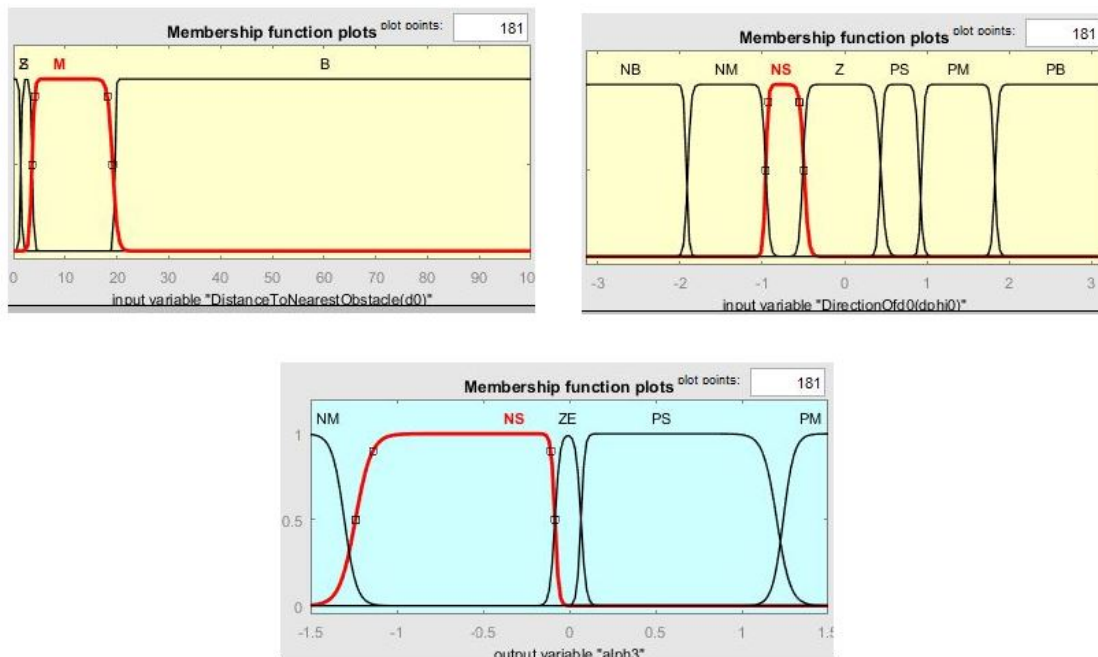
شکل ۳ خروجی تغییرات زاویه ی هدایت

مجموعه ی قوانین برای این ماژول به صورت (شکل ۴) است.

$\alpha \Rightarrow \Delta\phi \Downarrow$	NB	NM	Z	PM	PB
NB	Z	Z	NM	NB	NB
NM	Z	Z	NM	NM	NM
Z	PM	Z	Z	Z	NM
PM	PB	PB	PM	Z	Z
PB	PB	PB	PM	Z	Z

شکل ۴ مجموعه قوانین هدایت زاویه به سوی مقصد

ماژول هدایت زاویه برای عدم برخورد با مانع در این ماژول نیز مشابه قسمت قبل دو ورودی و یک خروجی وجود دارد. ورودی ها شامل فاصله از مانع و زاویه نسبت به مانع هستند.

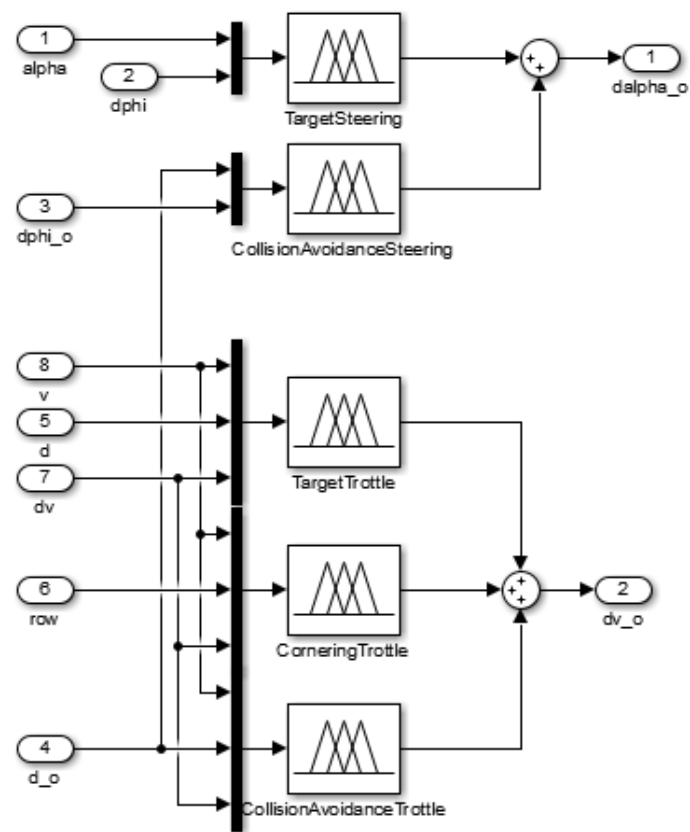


Active:

شکل ۵ دو ورودی و خروجی ماژول هدایت زاویه برای عدم برخورد با مانع

کنترلر سرعت

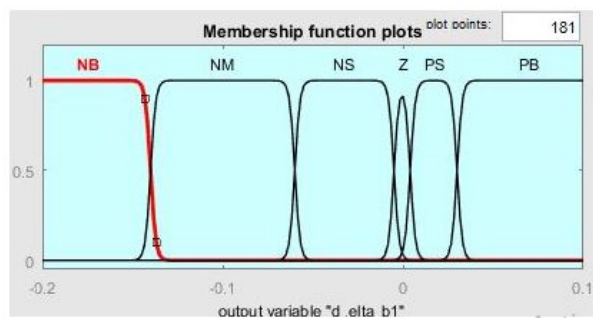
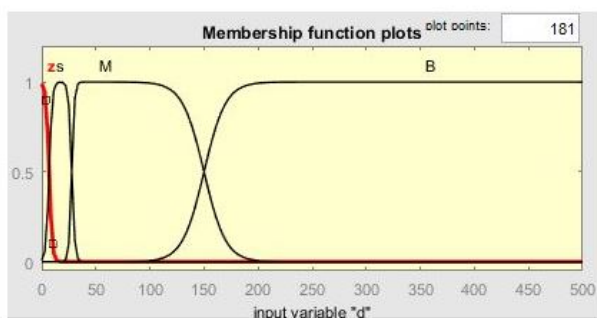
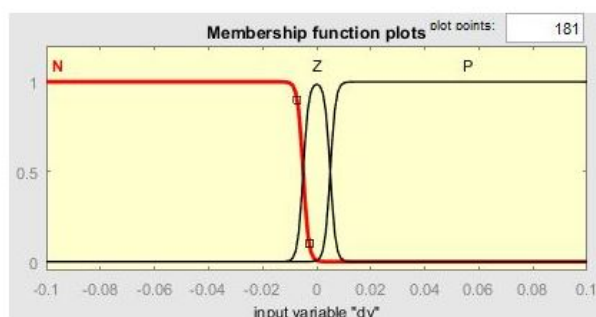
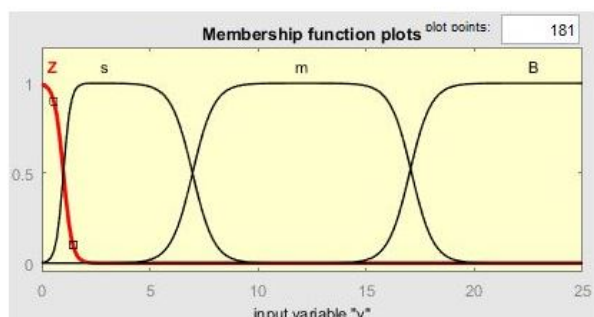
برای استفاده از دانش و تجربه انسانی موثر در کنترل سرعت خودرو، مجدداً مثل بخش کنترل فرمان، مسئله به چند بخش جداگانه تقسیم شده است و ماژول کنترل کننده فازی برای هر بخش طراحی شده است. این بخش ها همانطور که در شکل زیر مشاهده میشود شامل مقدار گاز دادن برای رسیدن به هدف، مقدار گاز دادن برای پیچیدن در پیچ جاده و مقدار گاز دادن برای وضعیت عدم برخورد با مانع میباشد. در این شکل ورودی و خروجی های ماژولی که برای این هدف طراحی شد نیز قابل مشاهده است. زاویه نهایی پدال گاز جمع خروجی این ۳ ماژول است. پس در مجموع از کنترلرهای فازی دو خروجی داریم: تغییر زاویه فرمان و تغییر زاویه پدال گاز (که مقدار مثبت برای آن معادل فشردن پدال گاز و مقدار منفیش یعنی ترمز گرفتن)



جدول قوانین فازی برای ماژول اول (سرعت برای رسیدن به هدف) مطابق جدول زیر است:

$v \downarrow$	$d \Rightarrow \Delta v \downarrow$	Z	S	M	B
Z	N	PS	PS	PB	PB
	Z	PS	PS	PB	PB
	P	NS	Z	PB	PB
S	N	NS	Z	PS	PB
	Z	NS	NS	Z	PB
	P	NS	NM	NS	PB
M	N	NB	NS	PS	PS
	Z	NB	NM	Z	PS
	P	NB	NB	NM	Z
B	N	NS	NS	Z	PS
	Z	NB	NS	NS	Z
	P	NB	NB	NS	NB

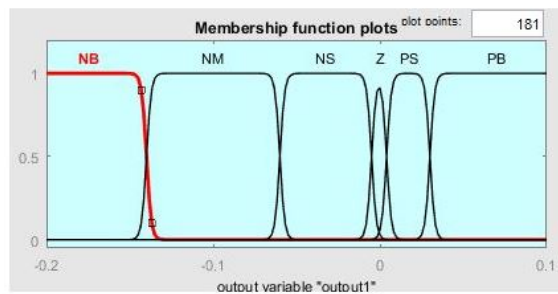
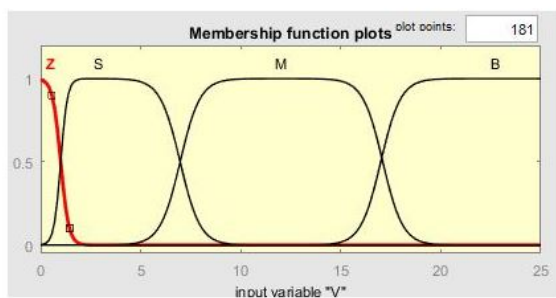
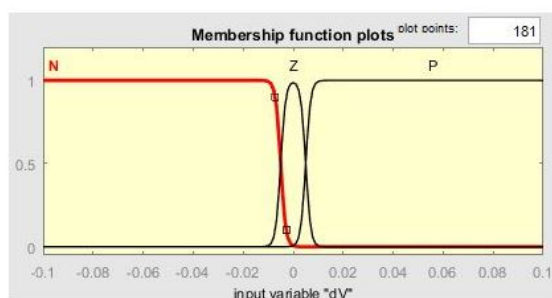
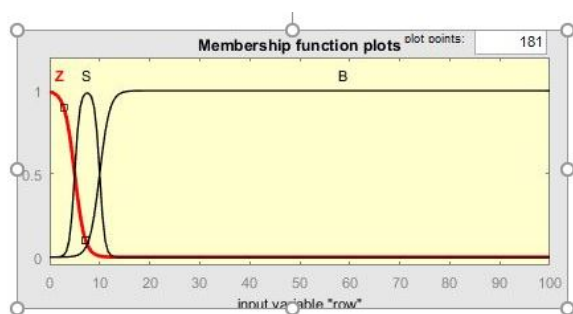
و توابع عضویت ورودی و خروجی:



برای مازول دوم (سرعت در پیچ جاده) جدول قوانین فازی:

$v \downarrow$	$\rho \Rightarrow \Delta v \downarrow$	Z	S	B
Z	N	Z	Z	Z
	Z	Z	Z	Z
	P	Z	Z	Z
S	N	Z	Z	Z
	Z	Z	Z	Z
	P	NS	Z	Z
M	N	Z	Z	Z
	Z	NS	Z	Z
	P	NS	Z	Z
B	N	NS	Z	Z
	Z	NS	Z	Z
	P	NS	NS	Z

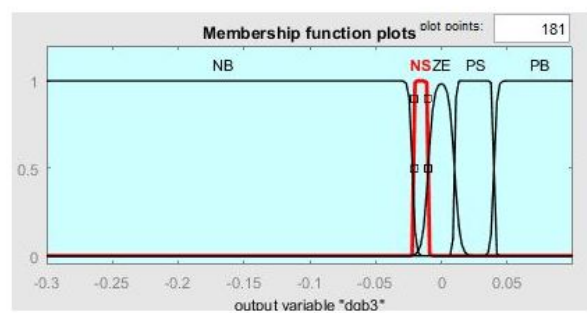
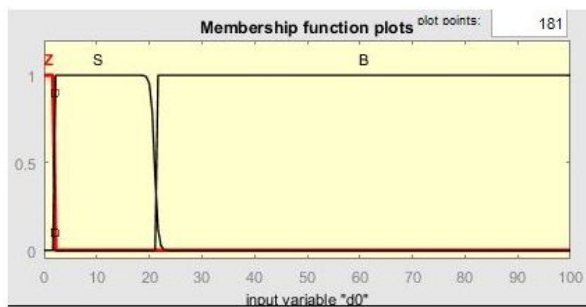
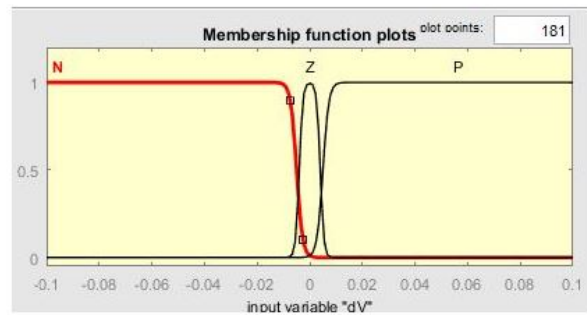
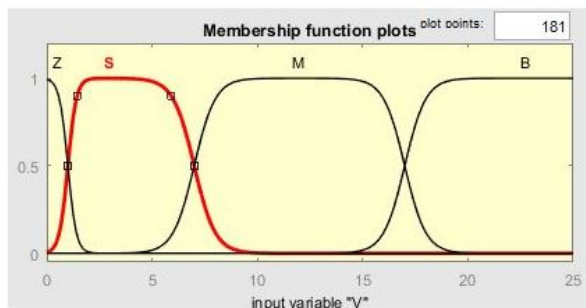
و توابع عضویت:



و برای مازول سوم، کنترل سرعت برای عدم برخورد با مانع جدول قوانین فازی:

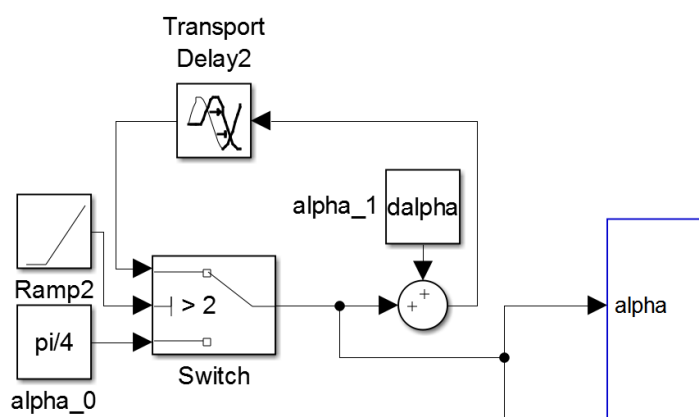
$v \downarrow$	$d_o \Rightarrow \Delta v \downarrow$	Z	S	B
Z	N	PS	PS	Z
	Z	PS	PS	Z
	P	NS	Z	Z
S	N	PS	Z	Z
	Z	PS	Z	Z
	P	NS	NS	Z
M	N	NS	Z	Z
	Z	NB	NS	Z
	P	NB	NB	Z
B	N	NS	NS	Z
	Z	NB	NS	Z
	P	NB	NB	Z

و توابع عضویت:



4. سیمولینک زاویه هدایت فرمان و پدال (مدل سیستم)

برای بلوک های کنترل فازی، نیازمند ورودی هایی هستیم که یا به عنوان مقدار اولیه به سیستم وارد می شوند، و یا با استفاده از خروجی های مراحل قبل تولید می شوند. می دانیم که خروجی های سیستم، تغییرات زاویه ی هدایت فرمان (α_o) و تغییرات زاویه ی هدایت پدال های گاز و ترمز (dv_o) هستند. (مقادیر مثبت مربوط به پدال گاز و مقادیر منفی برای پدال ترمز اند.) ورودی تولیدی اول، زاویه ی هدایت فرمان (α) است. برای تولید این ورودی، کافی ست تا تغییرات زاویه ی فرمان را با آخرین مقدار زاویه ی فرمان جمع کنیم. این مقدار در ابتدا برای مدت ۲ ثانیه توسط بلوکی به صورت اولیه تعریف می شود. (شکل ۶)



شکل ۶ بلوک اعمال مقدار اولیه به زاویه ی هدایت و تولید ورودی

مختصات مانع و مختصات مقصد، به صورت x و y هایی توسط بلوک های $constant$ تولید شده اند. با توجه به اینکه تغییرات زاویه ی پدال، در محدوده ی مثبت، نمایشگر افزایش سرعت بوده و در محدوده ی منفی، نمایشگر کاهش سرعت می باشند، برای تولید مدل سرعت از طریق زاویه ی پدال، کافی است زاویه ی مزبور را در یک بهره ی مناسب ضرب کنیم. تغییرات زاویه در محدوده ی $(0.02, -0.01)$ است که موجب اعمال تغییراتی حدوداً در محدوده ی $(2, -1)$ در سرعت خواهد شد. بنابراین انتخاب عدد ۱۰۰ به عنوان بهره برای این منظور مناسب است. از آنجایی که این خروجی مستقیماً برابر است با تغییرات سرعت، به عنوان ورودی تغییرات سرعت (dv)، همین خروجی را با استفاده از یک تاخیر، اعمال می نماییم. بدین ترتیب ورودی تغییرات سرعت به راحتی قابل تولید است.

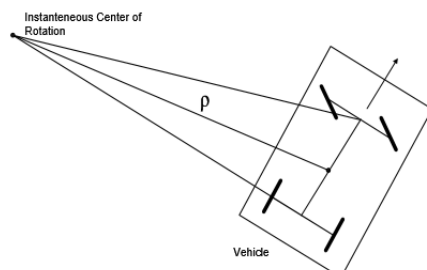
مشابه محاسبه ی زاویه ی هدایت فرمان، با جمع تغییرات سرعت، با آخرین مقدار سرعت، سرعت ورودی جدید محاسبه خواهد شد. این مقدار نیز در ابتدا مشابه (شکل ۶) توسط کاربر به صورت مقدار اولیه تعیین می گردد.

برای محاسبه ی تغییرات موقعیت، کفایت ابتدا تغییرات سرعت را در محور های x و y تجزیه نماییم. سپس با اعمال انتگرال به این داده ها، تغییرات موقعیت را در راستای x و y خواهیم داشت. که مشابه زاویه ی هدایت و سرعت مشابه (شکل ۶) به آن ها مقدار اولیه می دهیم.

$$dvx = dv * \cos(\alpha_o);$$

$$dv_y = dv \sin(\alpha_o);$$

ورودی دیگر، شعاع خمیدگی خودرو است. در هندسه ی دیفرانسیل، شعاع خمیدگی، یک خمیدگی دو طرفه است. برای یک منحنی، برابر با شعاع کمان چرخشی است، که به بهترین شکل در آن نقطه، به منحنی میل می نماید. (شکل ۷)



شکل ۷ شعاع انحنای خودرو

برای محاسبه ی شعاع انحنای (r_{OW}) از رابطه ی زیر استفاده می گردد.

$$R = \left| \frac{ds}{d\varphi} \right| = \left| \frac{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}{\dot{x}\ddot{y} - \dot{y}\ddot{x}} \right|, \quad \text{where} \quad \dot{x} = \frac{dx}{dt}, \quad \ddot{x} = \frac{d^2x}{dt^2}, \quad \dot{y} = \frac{dy}{dt}, \quad \ddot{y} = \frac{d^2y}{dt^2}.$$

بنابراین در متلب خواهیم داشت :

$$dv_x = ddv \cos(\alpha_o);$$

$$dv_y = ddv \sin(\alpha_o);$$

$$r_{ow} = v^3 / \text{abs}((dx * dv_y) - (dy * dv_x));$$

دو ورودی دیگر، زاویه ی خودرو از مقصد (dphi) و زاویه ی خودرو از مانع (dphi_o) هستند. برای محاسبه ی این زوایا، از روابط زیر استفاده می گردد.

```

if ((x_t-x)<0)
    dphi = atan ((y_t-y)/(x_t-x))+pi ;
else
    dphi = atan ((y_t-y)/(x_t-x));
end

if ((x_o-x)<0)
    dphi_o = atan ((y_o-y)/(x_o-x));
else
    dphi_o = atan ((y_o-y)/(x_o-x));
end

```

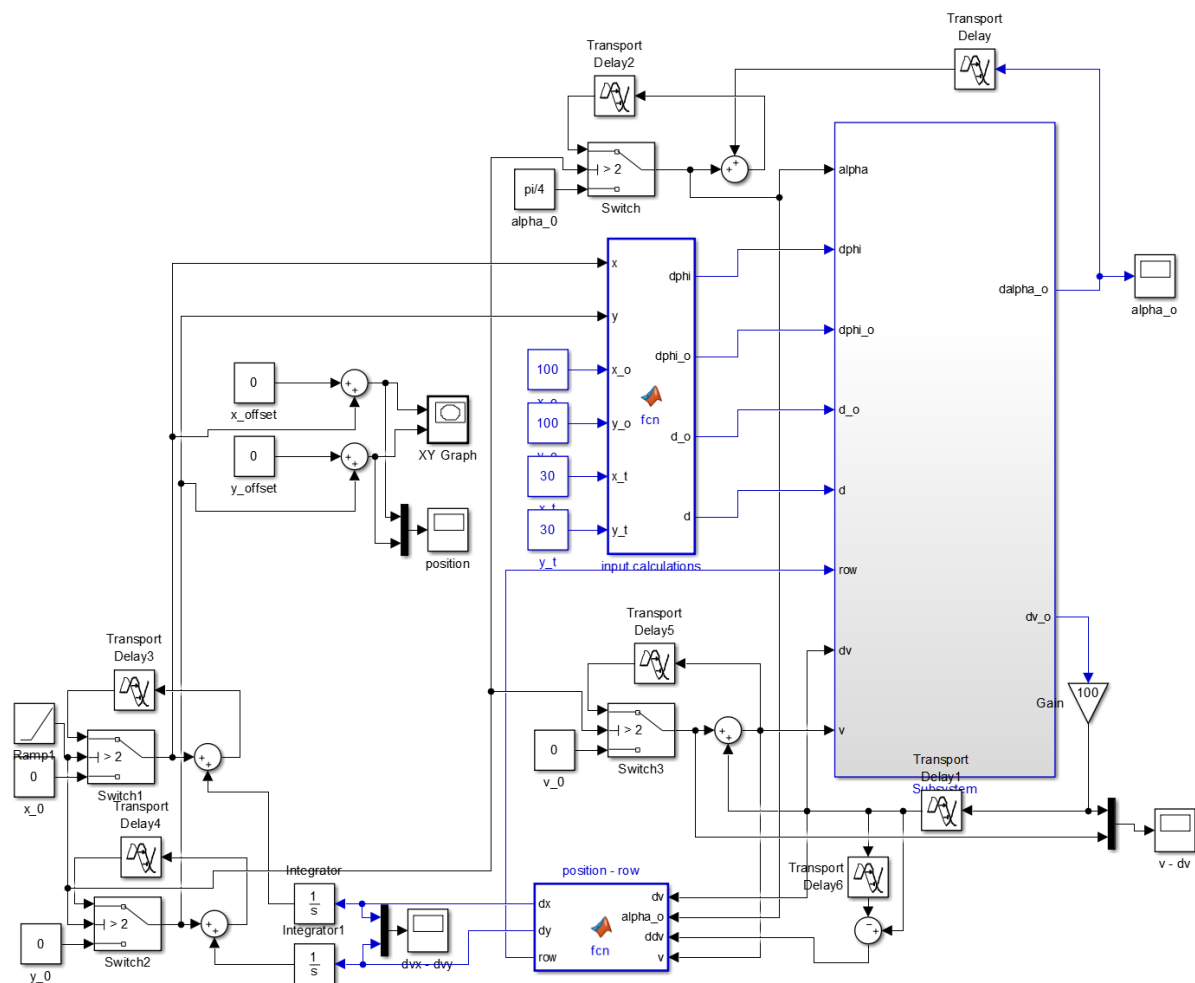
علت استفاده از شرط if این است که رابطه ی \tan^{-1} فقط برای زوایای در محدوده ی $(-90, 90)$ درجه قابل استفاده است. در انتها نوبت به محاسبه ی مقادیر فاصله از مقصد (d) و فاصله از مانع (d_o) می رسد. رابطه ی مربوطه به صورت زیر است.

```

d_o = norm([x_o-x),(y_o-y)];

```

با اعمال مختصات های x و y محاسبه شده، به یک بلوک xy graoh می توان موقعیت نقطه ی حرکتی را در مختصات x و y مشاهده نمود.



شکل ۸-۱ نمایشی از بلوک دیگرام های سیمولینک متلب

با توجه به (شکل ۳)، مطالب توضیح داده شده قابل مشاهده است. دستور های مربوط به دو تابع متلب، پیش تر ذکر شده اند. در بلوک subsystem کنترلر های فازی قرار گرفته اند.

۵. دنبال کردن یک شی متحرک^۱

در این بخش در نظر داریم با استفاده از منطق فازی، خودرو هوشمند ما یک شی متحرک را دنبال کند. که در موارد مختلفی می تواند کاربرد داشته باشد. برای این منظور به ترتیب زیر عمل کردیم:

۱. ابتدا برای نشان دادن شی متحرک از یک مدل استفاده کردیم. در این مدل پروفایل شتاب شی را تولید کردیم. که با انتگرال گیری از آن، سرعت و سپس مکان آن بدست آمد. خروجی ۷۱ در کد زیر بیانگر شتاب شی متحرک است.

¹ car following moving object

```

function y1 = fcn(t)
y1 = 0;

if t>0 && t<10.1
    y1 = 2;
elseif (t>=10.1 && t<15) || (t>=25 && t<40) || (t>=44 && t<49) || (t>=53 && t<58) || (t>=60 && t<65)
    y1 = 0;
elseif t>=15 && t<25
    y1 = -2;
elseif t>=40 && t<44
    y1 = 5;
elseif t>=49 && t<53
    y1 = -5;
elseif t>=58 && t<60
    y1 = 10;
elseif t>=65 && t<67
    y1 = -10;
end
end

```

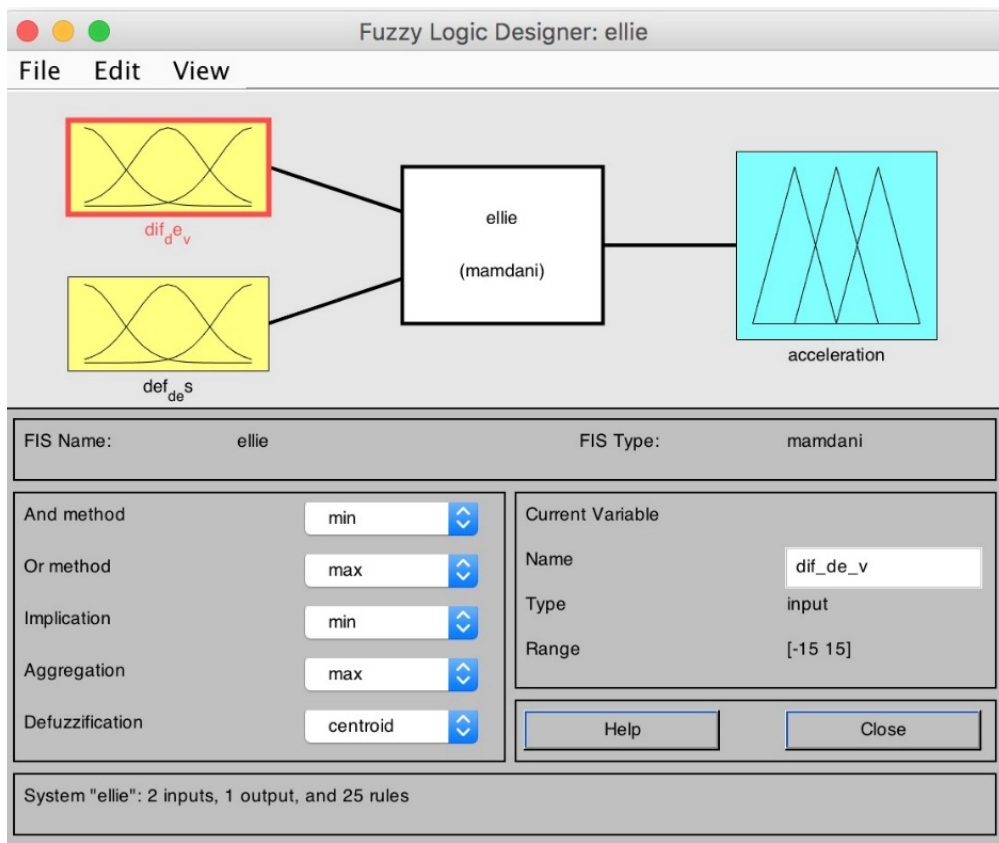
۲. برای خودرو یک مدل در نظر گرفتیم که از روی سرعت شی متحرک، شتاب خود را تولید می کند.

```

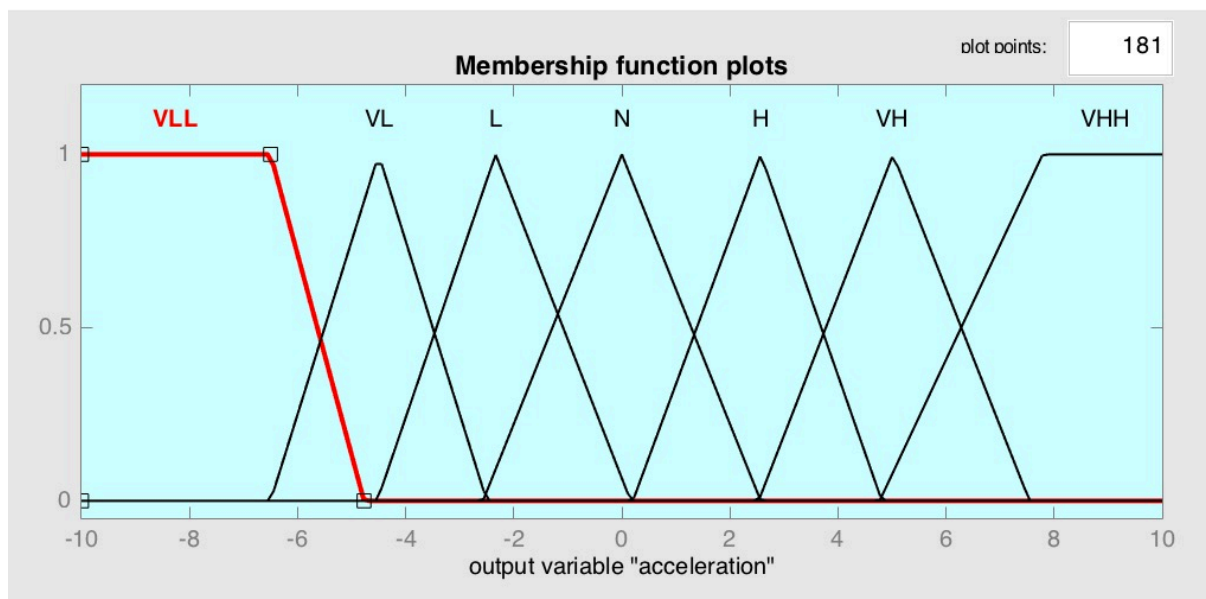
1 function y = fcn(u)
2
3 - velocity = [0:5:80]*10/36;
4 - sepration = [1,8,10,12,14,16,19,23,26,29,33,37,41,46,51,56,62];
5 - y = interp1(velocity, sepration, u, 'Linear');
6
7 end

```

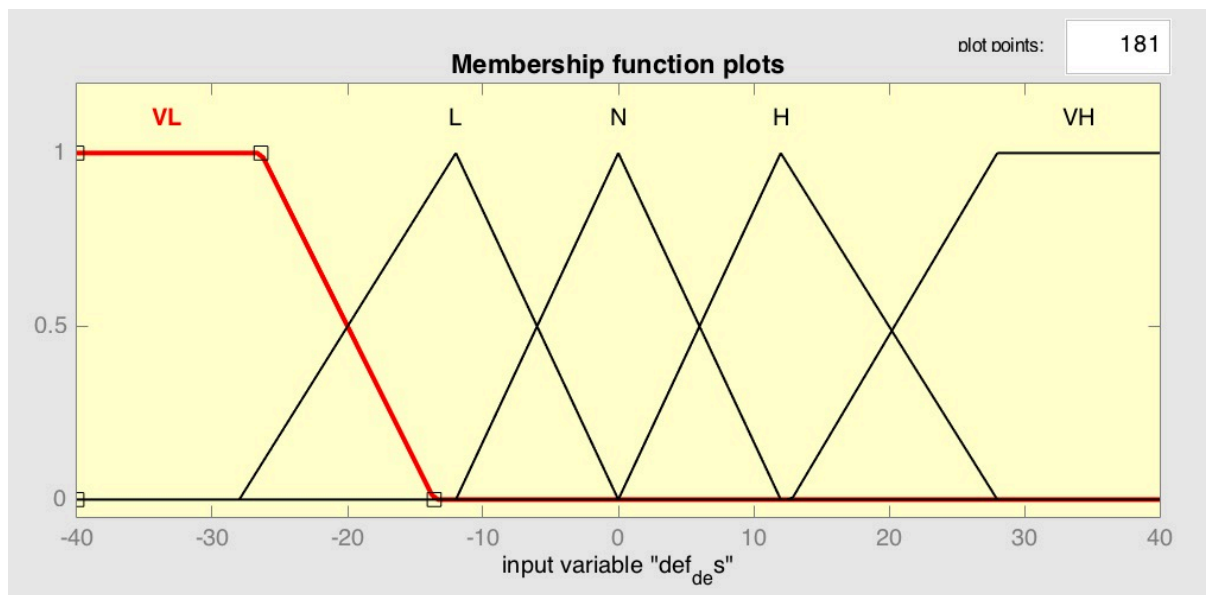
۳. از یک کنترلر فازی برای هدف خود بهره بردیم که ورودی آن خطای سرعت و شتاب بین شی متحرک و خودرو است. و خروجی آن شتاب خودرو است. توابع عضویت و مجموعه قوانین در شکل های ... و آمده است.



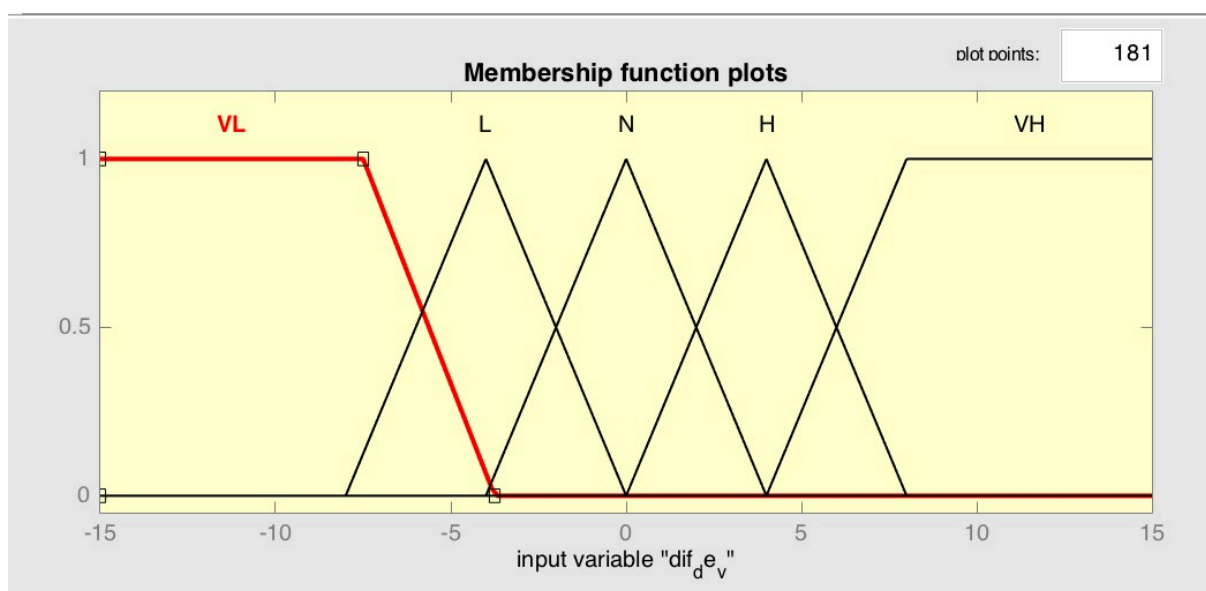
شکل 9 منطق فازی



شکل 10 تابع عضویت خروجی

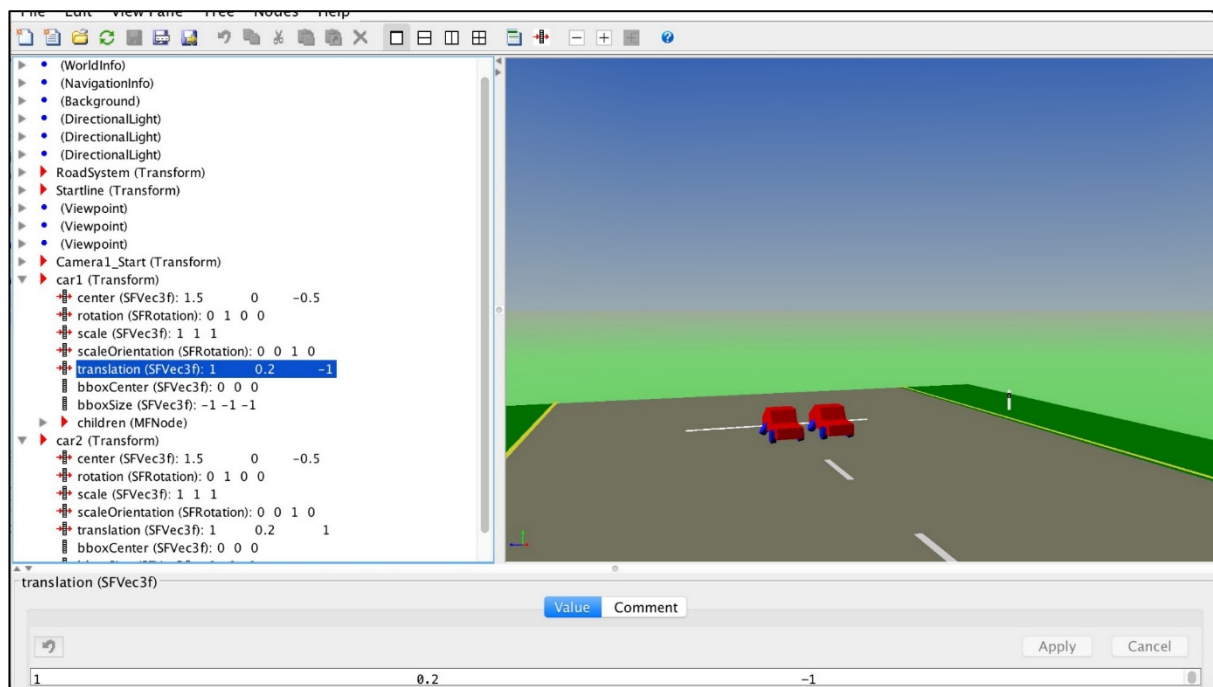


شکل 11 توابع عضویت ورودی اول



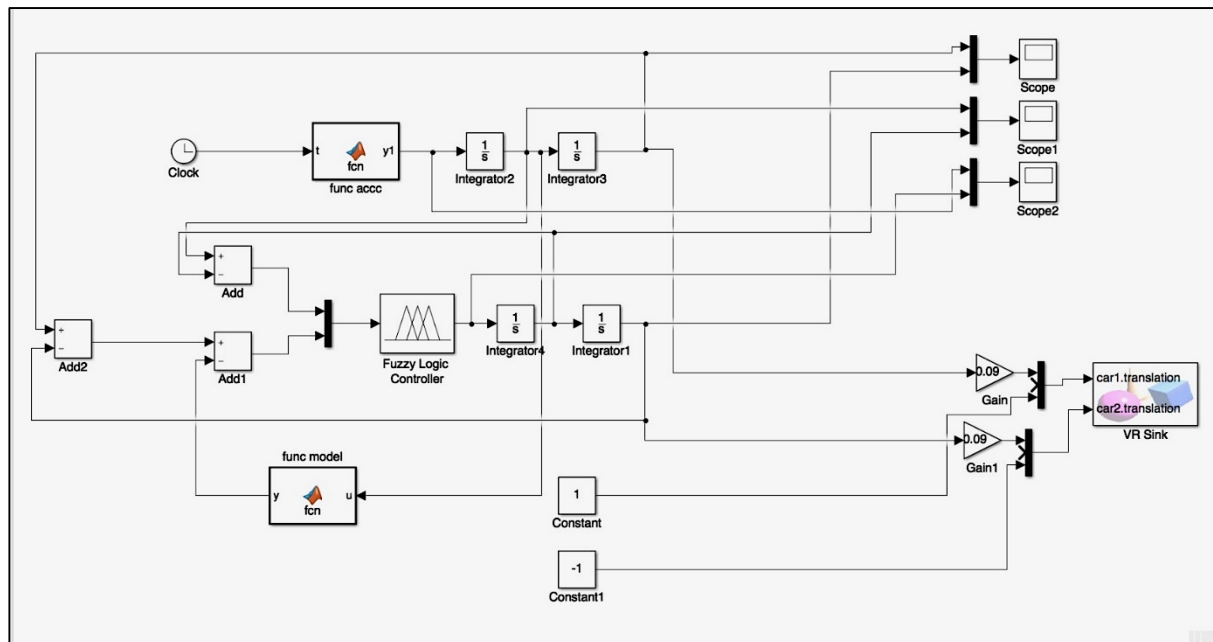
شکل 12 توابع عضویت ورودی دوم

۴. در نهایت برای نمایش دنبال کردن شی توسط خودرو، از تولباکس واقعیت مجازی متلب استفاده کردیم. به این صورت که با استفاده از کتابخانه این تولباکس، محیط جاده و دو ماشین به عنوان ماشین اصلی و شی متحرک اضافه شدند؛ و با تعریف پورت ورودی برای کنترل مختصات آن را با استفاده از داده های بدست آمده از مکان خودرو و شی و متحرک کنترل کردیم.



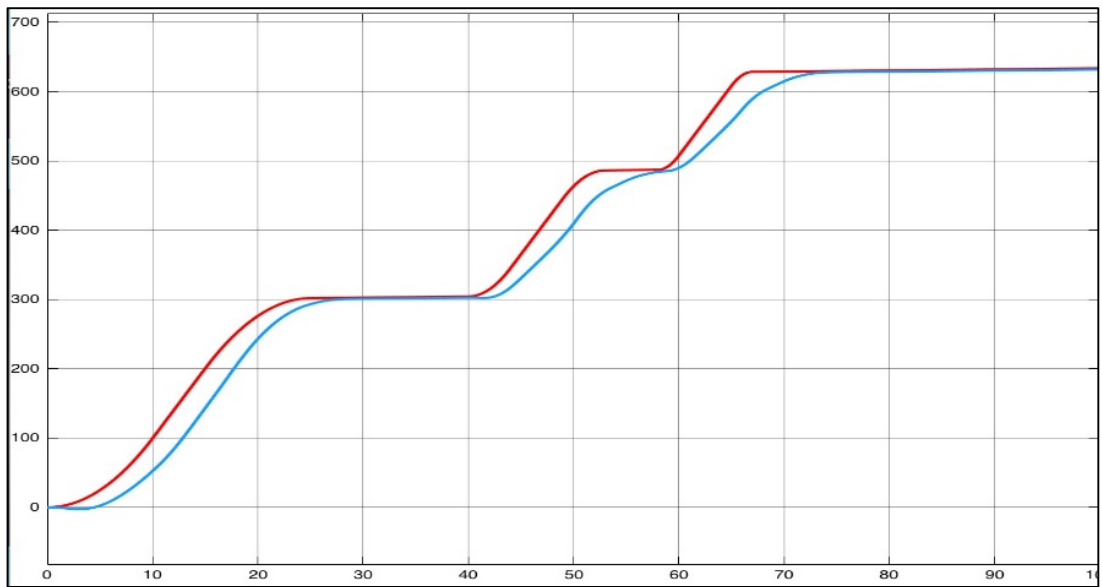
شکل 13 محیط واقعیت مجازی سیمولینک

در نهایت کد سیمولینک اصلی به صورت زیر خواهد بود.

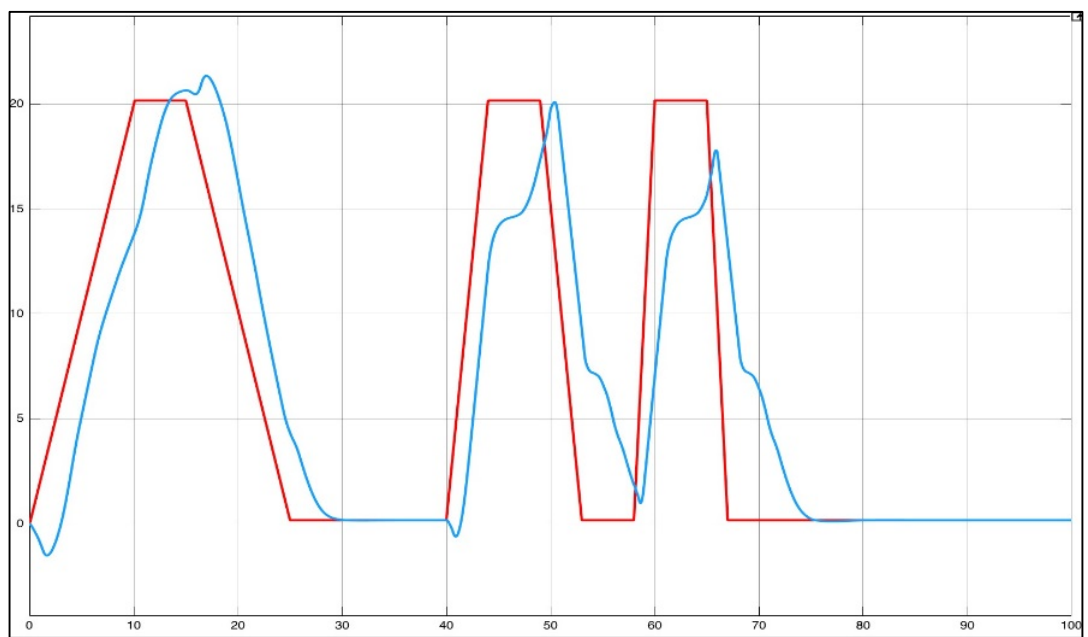


شکل 14 کد سیمولینک مربوط به دنبال کنندگی شی متحرک

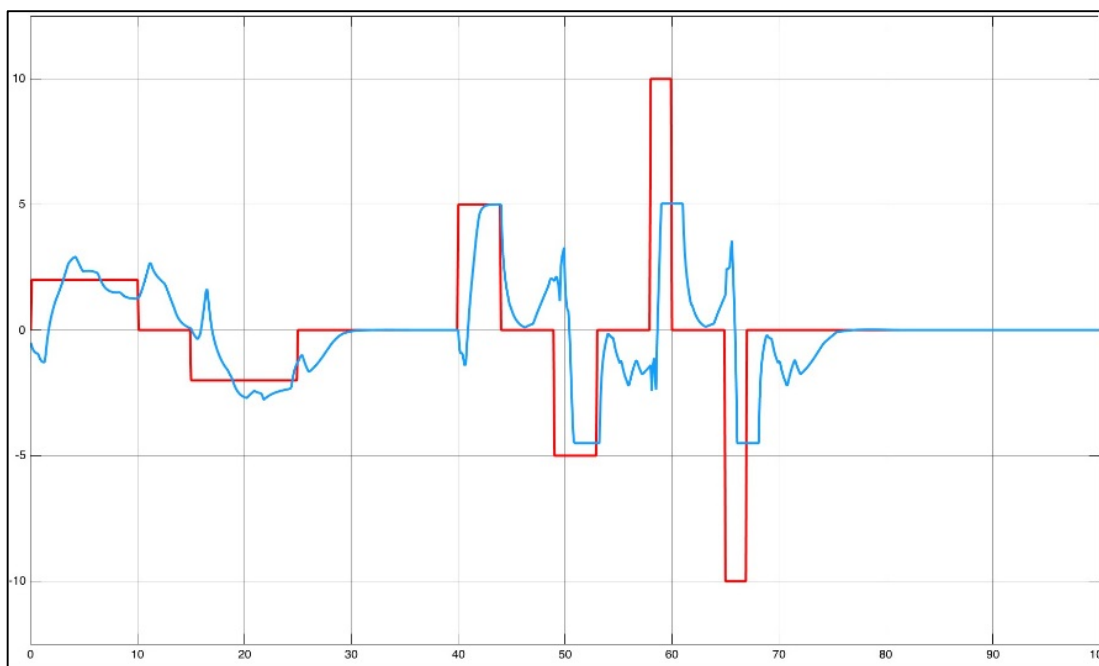
نتایج سیمولیشن به ترتیب زیر خواهد بود:



شکل 15 خروجی مکان شی متحرک و خودرو هوشمند



شکل 16 خروجی سرعت شی متحرک و خودرو هوشمند



شکل 17 خروجی شتاب شی متحرک و خودرو

همانطور که مشاهده می کنید، نمودار رنگ قرمز مربوط شی متحرک است و رنگ آبی مربوط به خودرو هوشمند کنترل شده به روش فازی است. با توجه به نمودار، پروفایل شتاب تولید شده در شکل به خوبی پروفایل شتاب خودرو و هم چنین سرعت و مکان را، با دقت مناسبی دنبال می کند. در نهایت مکان خودرو در فاصله های زمانی تعیین شده بر مکان شی منطبق می شود. نمایش انیمیشن این بخش در پیوست تقدیم شده است.

6. جمع بندی

در این پروژه هدف کنترل فازی خودرو بدون سرنشین به روش فازی بود. برای کنترل فازی این سیستم با توجه به تجربه و دانش انسان در حین رانندگی، برای هدایت و کنترل سرعت دو کنترلر جداگانه طراحی شد. کنترلر هدایت دارای دو ماژول جداگانه برای کنترل هدایت برای حالت رسیدن به هدف و عدم برخورد به مانع است. در واقعیت نیز به همین گونه است که راننده با توجه به هدف و موانع پیش روی خود زاویه فرمان را تنظیم می کند. که خروجی در دو حالت با هم جمع می شوند و به عنوان زاویه هدایت فرمان استفاده می شود. به تبع وقتی به مانع نزدیک هستیم، وزن خروجی مربوط به حالت وجود مانع بیشتر خواهد بود. کنترلر سرعت نیز دارای سه ماژول جداگانه برای کنترل سرعت است، که با توجه به مکان هدف و مانع پیش رو و زاویه پیچ جاده زاویه پدال گاز مناسب را برای کنترل سرعت تولید می کند. برای اینکه خودرو توانایی دنبال کردن شی و خودرو دیگری هم داشته باشد، کنترلر فازی جداگانه ای طراحی شد.

گفتنی است برای طراحی سیستم های فازی که نیاز به تجربیات یک راننده نیاز بود، از اطلاعات قوانین مقاله های اشاره شده در بخش مراجع استفاده کردیم. اما تمامی مراحل مدل سازی دینامیک خودرو و کد زنی سیمولینک متلب و ساخت محیط سه بعدی برای ارایه، توسط اعضای گروه صورت گرفت. در بخش پارکینگ خودرو هوشمند به روش دابل و پارکینگ در گاراژ نیز تلاش هایی شد و قوانین مربوطه پیاده سازی شدند. اما به دلیل ضیق وقت به عنوان فعالیت های آینده خود در این زمینه به آن خواهیم پرداخت.

- [1] B. Boada, M. Boada, V. Diaz, Fuzzy-logic applied to yaw moment control for vehicle stability, *Vehicle System Dynamics* 43(10) (2005) 753-770.
- [2] M. Trabia, L.Z. Shi, N.E. Hodge, A fuzzy logic controller for autonomous wheeled vehicles, (2006).
- [3] H.-m. Li, X.-b. Wang, S.-b. Song, H. Li, Vehicle Control Strategies Analysis Based on PID and Fuzzy Logic Control, *Procedia Engineering* 137 (2016) 234-243.
- [4] K. Kozłowski, D. Pazderski, Modeling and control of a 4-wheel skid-steering mobile robot, *Int. J. Appl. Math. Comput. Sci* 14(4) (2004) 477-496.
- [5] P. Benavidez, J. Lambert, A. Jaimes, M. Jamshidi, Landing of an ardrone 2.0 quadcopter on a mobile base using fuzzy logic, 2014 World Automation Congress (WAC), IEEE, 2014, pp. 803-812.