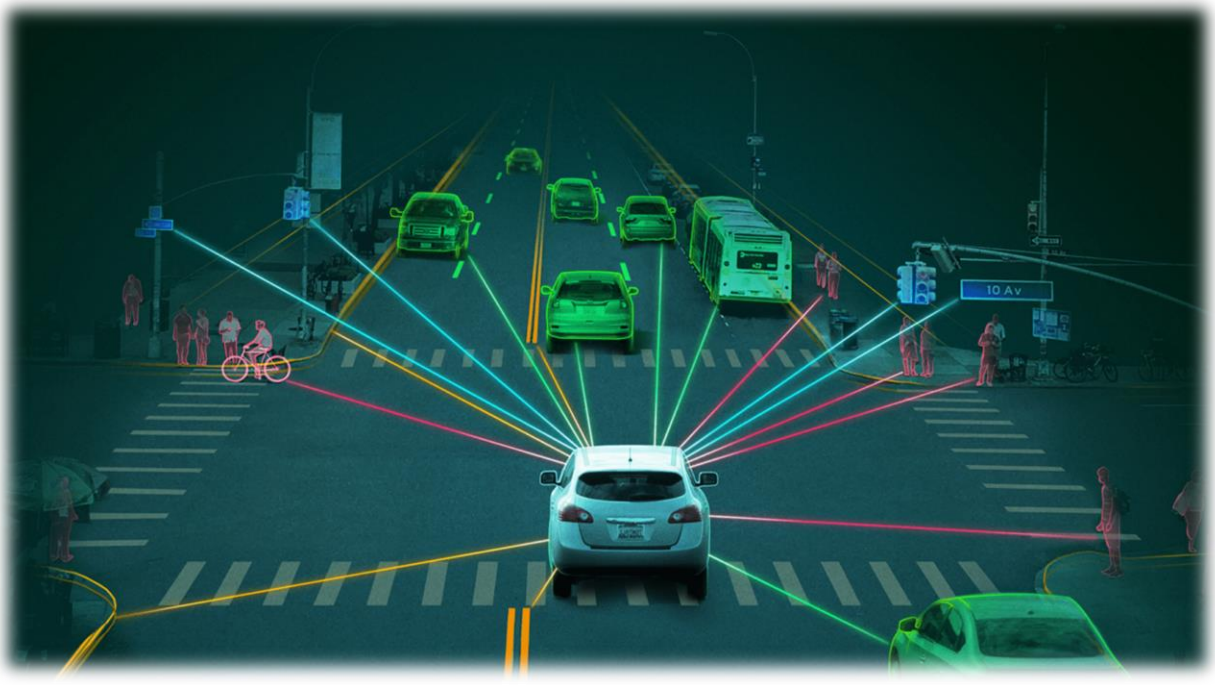
Department of
Computer Engineering

Amirkabir University of Technology
(Tehran Polytechnic)

پروژه فازی

هوش محاسباتی بهار ۱۴۰۲

اتومبیل خودران فازی



مقدمه

در این پروژه می‌خواهیم لامبورگینی خود را به هوش مصنوعی فازی مجهز کنیم تا بتواند مسیر مشخص شده را بدون هیچ گونه برخوردی طی کند و به مقصد خود پیش دوستان شاد درختی اش برسد. برای این منظور فرض کنید که ۳ عدد سنسور اولتراسونیک به خودروی خود متصل کردیم که به ترتیب در ۱- سمت چپ جلوی ماشین و ۲- سمت راست جلوی ماشین و ۳- جلوی خودرو نصب شده است. کار این سنسورها این است که فاصله خودرو را تا گاردریل و مانعی که در امتداد امواج ارسال شده‌شان هست را برمیگرداند. این پروژه ۲ فاز دارد. در فاز اول ما تنها از دو سنسور اول برای جلوگیری از برخورد استفاده می‌کنیم و سنسور سوم در فاز دوم (امتیازی) به کمک ما می‌آید.

برای پیاده سازی لازم است تا به ترتیب ۳ مرحله زیر را که توضیحات مربوط به آن‌ها در ادامه آمده است را انجام دهید:

- **مرحله اول)** ورودی ها را به صورت فازی در آورید.(Fuzzification)
- **مرحله دوم)** خروجی را به صورت فازی به کمک قواعد در آورید.(Inference)
- **مرحله سوم)** خروجی فازی شده را به صورت مقدار مطلق در آورید.(Defuzzification)

قوانینی که در پروژه باید از آن‌ها استفاده کنید در فایل rules.txt آمده است و شما فقط باید مراحل را در fuzzy_controller.py انجام دهید. در تابع decide در هر لحظه فاصله ی سمت چپ و فاصله ی سمت راست به شما داده می شود و شما باید خروجی مرحله ی سوم (میزان چرخش فرمان) را به عنوان return تابع برگردانید.

برای اینکه بتوانید خروجی پیاده‌سازیتان را مشاهده نمایید، در صورتی که کتابخانه pygame را روی سیستمتان نصب ندارید، باید با دستور زیر آن را نصب کنید:

- pip install pygame

پس از پیاده سازی میتوانید به کمک دستور زیر کدتان را اجرا نمایید:

- python simulator.py

چنانچه قبل از پیاده‌سازی دستور فوق را اجرا کنید خواهید دید که خودرو به گاردریل برخورد میکند. (همچنین می توانید برای تست کردن و آشنایی بیشتر با محیط بازی با استفاده از دکمه های چپ و راست کیبورد خودرو را کنترل کنید.)

فاز اول

مرحله اول) Fuzzification

ورودی های این مسئله شامل موارد زیر می باشد:

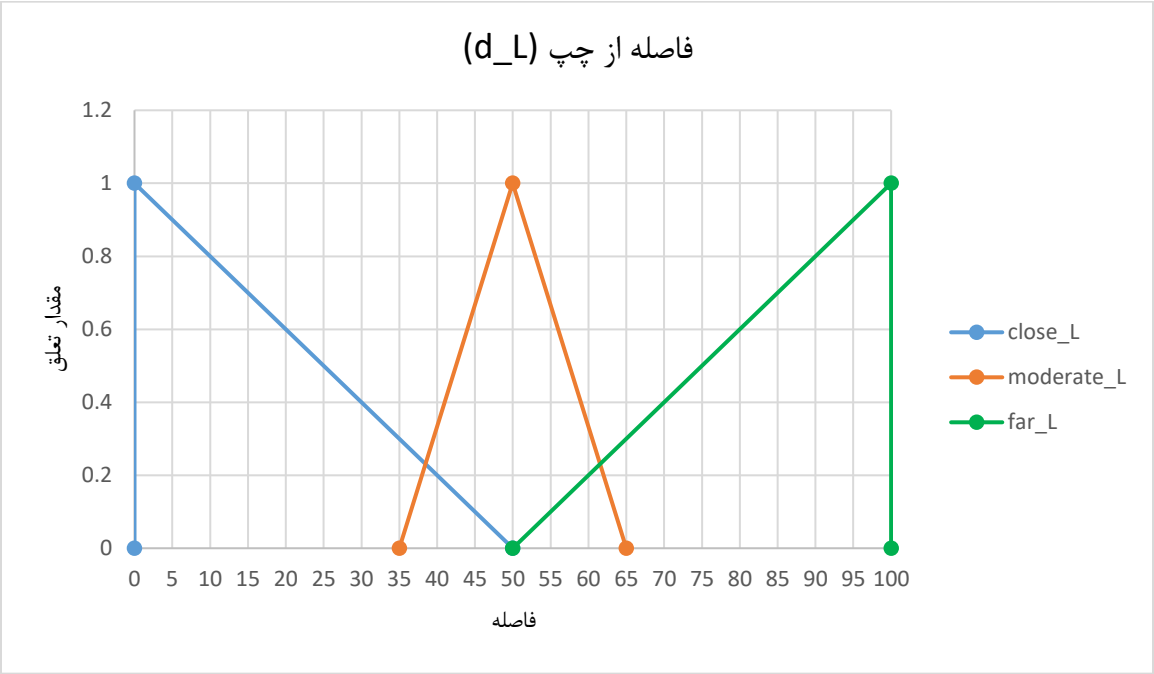
۱- فاصله ی سمت چپ جلوی ماشین تا گاردریل

۲- فاصله ی سمت راست جلوی ماشین تا گاردریل

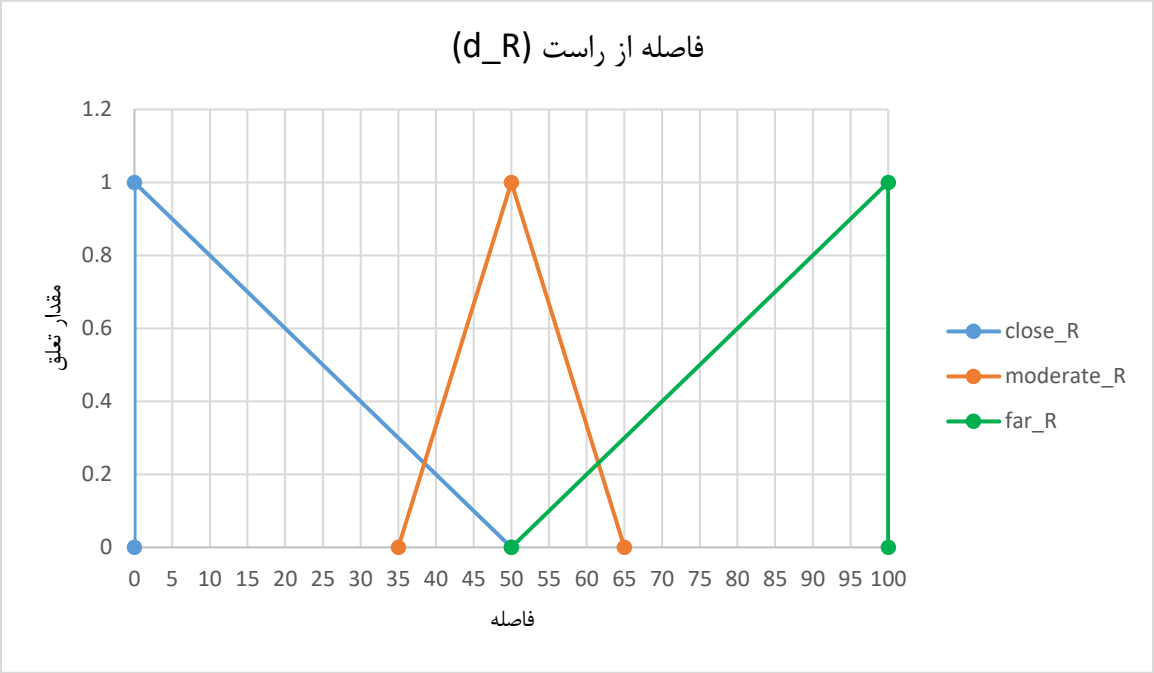
و در نهایت خروجی مسئله مقدار چرخش فرمان (rotation) می باشد .

برای حل مسئله به کمک منطق فازی، لازم است مقادیر ما از حالت مطلق به حالت فازی (نادقیق، نسبی) تبدیل شوند. به این مرحله Fuzzification یا فازی سازی گفته می شود. برای این منظور میبایست مجموعه های فازی تعریف شود و طبق تابع تعلق، میزان تعلق هر مقدار به هر مجموعه محاسبه شود. برای این منظور تابع های تعلق مجموعه های مورد نیاز در شکل های زیر آمده‌اند:

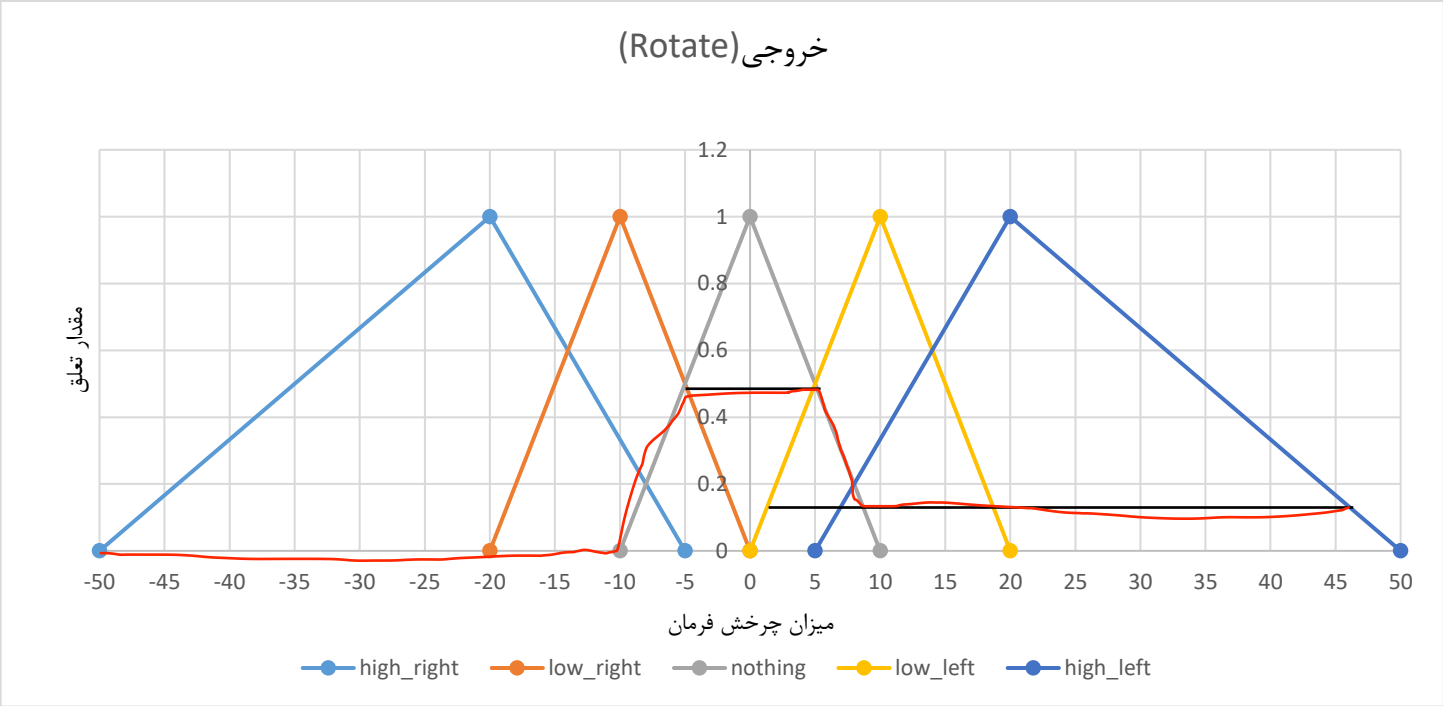
تابع تعلق مربوط به فاصله از چپ را برای ۳ مجموعه close_L و moderate_L و far_L می توانید در نمودار زیر مشاهده کنید:



تابع تعلق مربوط به فاصله از راست را برای ۳ مجموعه close_R و moderate_R و far_R می توانید در نمودار زیر مشاهده کنید:



تابع تعلق مربوط به خروجی میزان چرخش فرمان را برای ۵ مجموعه high_right، low_right، nothing، low_left، high_left میتوانید در نمودار زیر مشاهده کنید:



برای سادگی کار تابع های تعلق به صورت خطی تعریف شده اند و در پیاده سازی می بایست با توجه به شکل‌های بالا معادله خطوط را بدست آورید. بدیهی است که معادله های بالا معادلات خطوط ساده می باشند که با دو نقطه بدست می آیند. به عنوان مثال برای متغیر چرخش فرمان (rotation) ، مجموعه فازی high_left به صورت زیر تعریف می شود :

$$For\ x \in [5,20] \rightarrow membership\_function(x) = \frac{1}{15}x - \frac{1}{3}$$

$$For\ x \in [20,55] \rightarrow membership\_function(x) = -\frac{1}{35}x + \frac{11}{7}$$

$$Otherwise \rightarrow membership\_function(x) = 0$$

مرحله دوم) Inference

در مرحله بعد لازم است مقادیر فازی بدست آمده در قوانین موجود برای حل مسئله بررسی شوند. به این مرحله Inference گفته می شود . به طور مثال قوانین زیر را در نظر بگیرید:

$$IF\ (d\_L\ IS\ close\_L\ )\ \ AND\ \ (d\_R\ IS\ moderate\_R)\ \ THEN\ \ Rotate\ IS\ low\_right$$

* این قانون می گوید اگر فاصله ی سمت چپ جلوی ماشین تا گاردریل نزدیک بود و فاصله ی سمت راست جلوی ماشین تا گاردریل معمولی بود ، مقدار چرخش فرمان باید مقدار کمی به راست باشد.

$$IF\ (d\_L\ IS\ close\_L\ )\ \ AND\ \ (d\_R\ IS\ far\_R)\ \ THEN\ \ Rotate\ IS\ high\_right$$

$$IF\ (d\_L\ IS\ moderate\_L\ )\ \ AND\ \ (d\_R\ IS\ moderate\_R)\ \ THEN\ \ Rotate\ IS\ nothing$$

حال فرض کنید مقدار تعلق به مجموعه ی (close_L) مربوط به فاصله از چپ که در مرحله قبل انجام شده است برابر با ۰.۶ باشد همچنین مقدار تعلق به مجموعه ی (moderate_R) مربوط به فاصله از راست برابر با ۰.۷ باشد و مقدار تعلق به مجموعه ی (far_R) مربوط به فاصله از راست برابر با ۰.۴ باشد و مقدار تعلق بقیه مجموعه ها صفر باشد. در این حالت مجموعه قوانین بالا به شکل زیر در می آیند:

- if close_L=0.6 AND moderate_R=0.7 then rotation=low_right
- if close_L=0.6 AND far_R=0.4 then rotation=high_right
- if moderate_L=0 AND moderate_R=0.7 then rotation=nothing

همانطور که می دانید در منطق فازی روش های مختلفی برای محاسبه عملگر های اجتماع و اشتراک وجود دارد. در اینجا از روش ماکزیمم و مینیمم استفاده می کنیم. در نتیجه AND=min و OR=max محاسبه می شوند. به کمک گفته های بالا عبارات زیر حاصل می شوند:

- Membership(low_right)=min(0.6,0.7)=0.6
- Membership(high_right)=min(0.6,0.4)=0.4
- Membership(nothing)=min(0,0.7)=0

به مقادیر بدست آمده بالا، قدرت هر قاعده گفته می شود. در این حالت قاعده ۱ و ۲ تنها قاعده های فعال شده می باشند. بنابراین در این مرحله، خروجی که همان چرخش فرمان (rotation) می باشد را به صورت مجموعه هایی با مقادیر تعلق مختلف بدست می آورید .

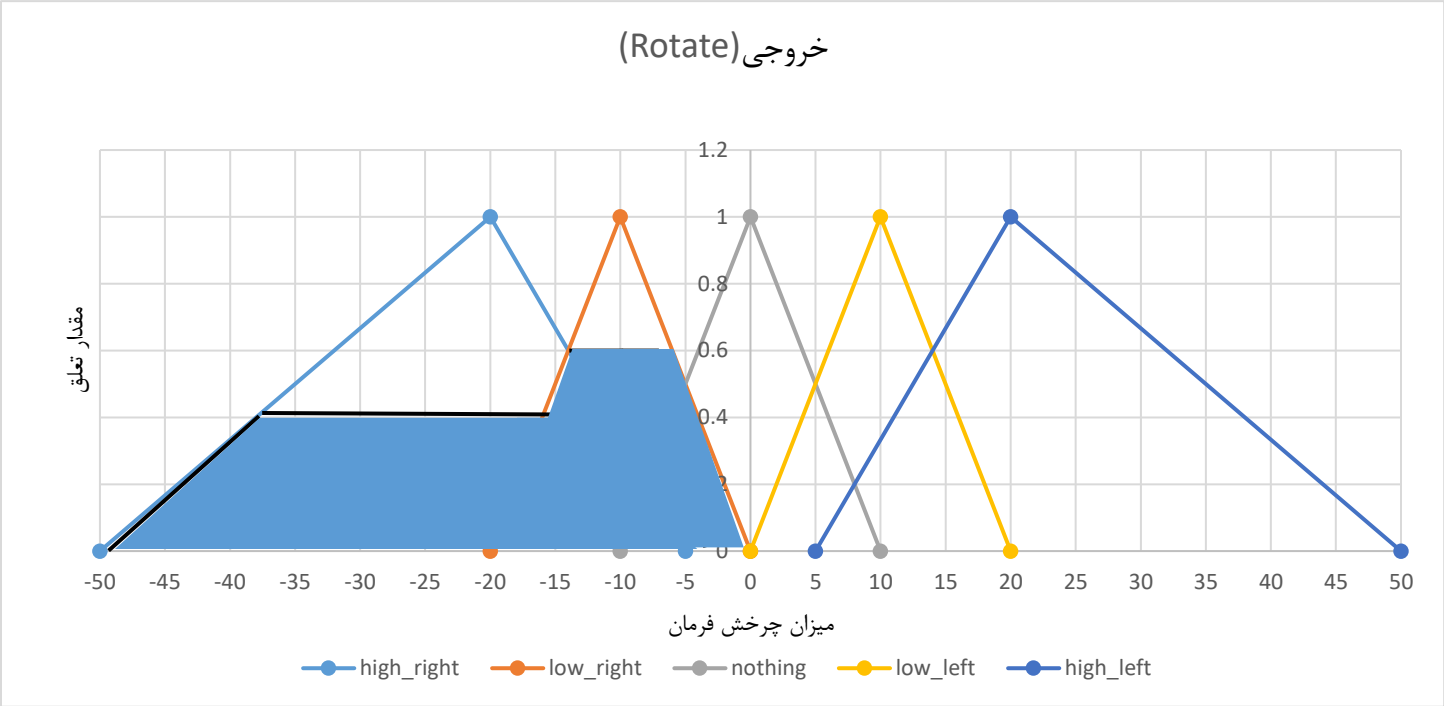
امتیازی : اگر در یک مسئله چندین قانون با مجموعه ی نهایی یکسان فعال شوند برای محاسبه ی مقدار تعلق نهایی این مجموعه چه باید کرد؟ (در این پروژه چنین حالتی پیش نمی آید)

مرحله سوم) Defuzzification

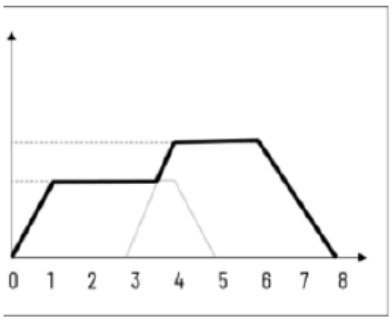
مرحله آخر Defuzzification نام دارد. در این مرحله به کمک استنتاج های انجام شده، مجدد به دنیای مقادیر مطلق بر می گردیم تا نیرو و جواب را به صورت مقدار مطلق بدست آوریم . برای غیرفازی سازی نیز روش های مختلفی وجود دارد که از مهم ترین و پرکاربردترین آن ها روش مرکز جرم می باشد. در ادامه مثال گفته شده مقادیر تعلق برای مجموعه های متغیر چرخش فرمان (rotation) به شکل زیر می باشد:

- low_right=0.6
- high_right=0.4
- nothing=0
- low_left=0
- high_left=0

در مثال بالا خروجی به شکل زیر خواهد شد :



* توجه فرمایید در حالاتی ممکن است شکل ها تو در تو باشند و همانطور که گفته شد max دو شکل (دو تابع) باید در نظر گرفته شود. مانند شکل زیر:



Aggregated fuzzy output

پس از آن که جواب تمام قاعده ها را با هم ترکیب کردیم مرکز جرم شکل حاصل شده را بدست می آوریم. در مثال بالا مرکز جرم برابر است با 21.73-

در واقع با محاسبه مرکز جرم از فضای فازی به فضای مطلق می رویم و جواب نهایی برابر است با 21.73- در نتیجه در سناریو بالا سیستم ما فرمان را به اندازه ی 21.73 به سمت راست می چرخاند تا ماشین فاصله ی خود را از سمت چپ حفظ کند و تصادف نکند.

نکات

- ✓ در محاسبه مرکز جرم استفاده از هر کتابخانه ای و یا هر روشی مانند استفاده از فرمول شکل های معروف یا... مجاز می باشد. پیشنهاد ما استفاده از فرمول اصلی مرکز جرم و محاسبه آن به صورت انتگرالی می باشد زیرا که ممکن است شکل های تودرتو ایجاد شود و در چنین مواردی می بایست تابع خروجی را به صورت قطعه قطعه در آورد و از هر قطعه انتگرال و معادله رو به رو را محاسبه کرد.
- ✓ از آنجایی که توابع تعلق همگی به صورت خطی هستند انتگرال فوق به کمک فرمول های انتگرال قابل محاسبه خواهد بود. گرچه استفاده از روش عددی و یا استفاده از کتابخانه های موجود بلامانع می باشد.

$$x^* = \frac{\int \mu_{\bar{C}}(x).x\,dx}{\int \mu_{\bar{C}}(x)\,dx}$$

- ✓ برای مشاهده نمونه انتگرال گیری برای محاسبه مرکز جرم می توانید از لینک زیر استفاده کنید :

https://codecrucks.com/center-of-gravity-method-for-defuzzification/

شکل زیر نیز یک شبه کد برای روش عددی می باشد :

```
soorat=0.0
makhraj=0.0
X=self.linspace(-50,50,1000)
delta=X[1]-X[0]
for i in X:
    U=self.max_rotate(i)
    soorat+=U*i*delta
    makhraj+=U*delta
center=0.0
if makhraj!=0:
    center=1.0*float(soorat)/float(makhraj)
```

(تابع linspace یک آرایه ۱۰۰۰ تایی از اعداد بین 50- تا 50 که هرکدام $\frac{50-(-50)}{1000} = 0.1$ با یکدیگر فاصله دارند باید تشکیل دهد.)

- ✓ همچنین کاملاً پیشنهاد می شود که ویدیو های زیر را برای انجام پروژه مشاهده فرمایید. این ویدیو ها همین مسائل را به صورت شهودی و با مثال عددی حل کرده اند که می تواند بسیار به درک شما از مسئله کمک کند .

https://www.youtube.com/watch?v=TReelsVxWxg

- مدت : ۱ ساعت و ۳۰ دقیقه
- موضوع : تدریس منطق فازی از ابتدا

https://www.youtube.com/watch?v=CBTEVFphv-E

- مدت : ۱۶ دقیقه
- موضوع : مثال عددی از ۳ فاز پروژه و مصور سازی آن به همراه توضیح هر ۳ مرحله

- ✓ استفاده از کتابخانه های مربوط به منطق فازی در هیچ یک از فاز های پروژه به جز موارد گفته شده (مانند انتگرال یا محاسبه مرکز جرم) مجاز نمی باشد.
- ✓ اکیدا توجه شود که پروژه حتی در صورت مهارت بالای شما در برنامه نویسی، یک شبه پیاده سازی نمی شود. بنابراین در مدیریت وقت خود دقت فرمایید .
- ✓ در نهایت، شما می بایست فایل های پروژه را به همراه یک گزارش کوتاه راجع به پروژه و عملکرد سیستم در قالب یک فایل زیپ با فرمت CI_PRJ2_studentnumber.zip در کورسز آپلود کنید. (پروژه تحویل دارد). ددلاین این پروژه ۵ خرداد ساعت ۲۳:۵۹ می باشد.

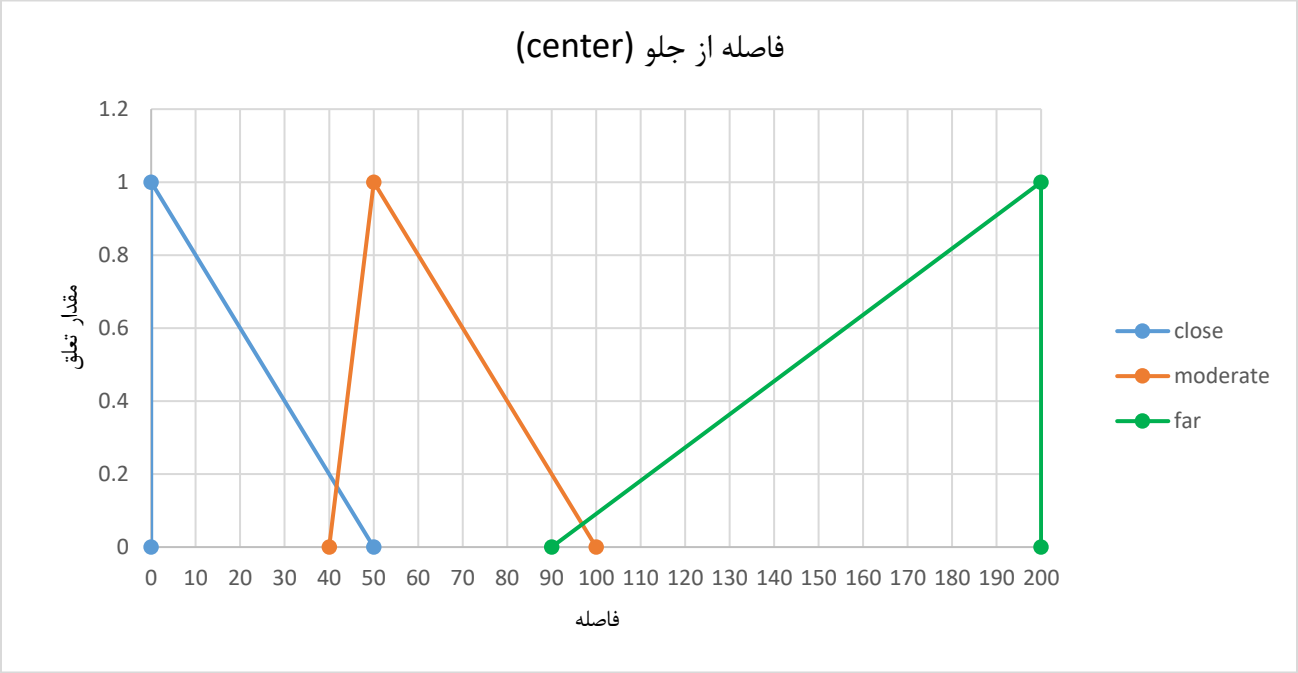
فاز دوم (امتیازی)

پس از انجام کامل فاز اول، در این فاز، می خواهیم سرعت ماشین نیز به جای اینکه ثابت باشد، توسط سیستم فازی کنترل شود. یعنی با استفاده از سنسور سوم (جلوی خودرو) فاصله ی ماشین را ارزیابی می کنیم و در نهایت تصمیم می گیریم که چقدر سرعت داشته باشیم.

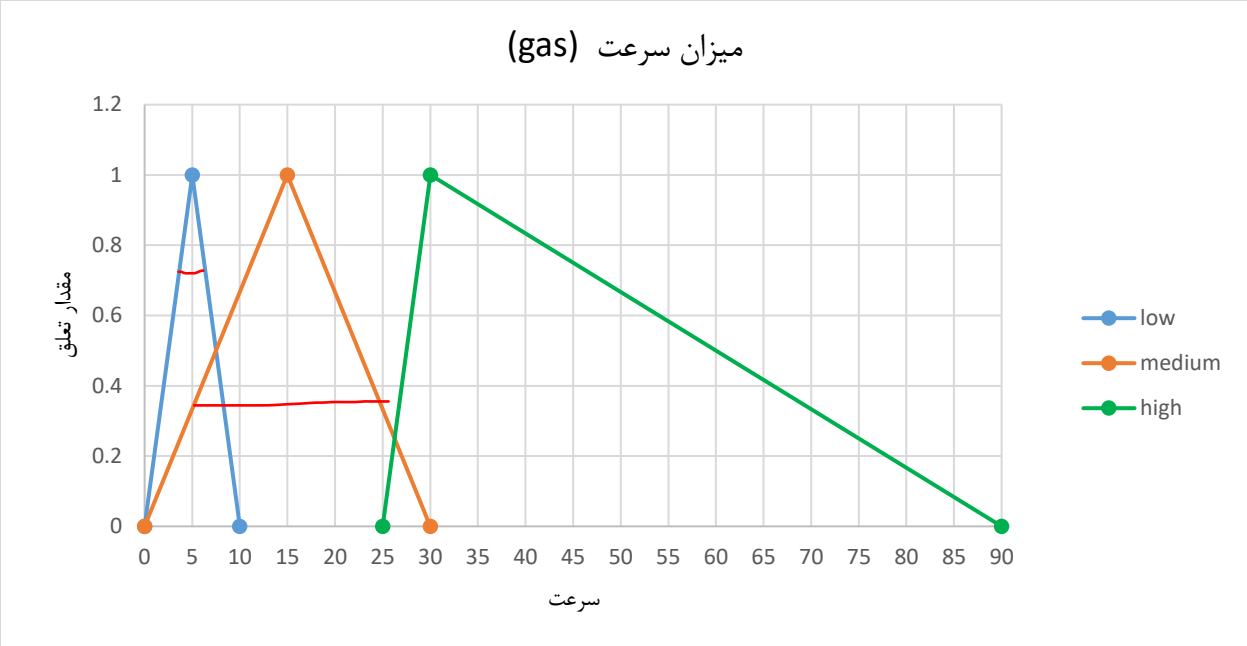
در این فاز باید فایل additional_controller.py کامل شود و قوانین فازی این بخش در فایل additional_rule.txt آمده است. همانند فاز اول، باید در تابع decide فایل additional_controller.py به جای سرعت ثابت ۳۰، مقدار خروجی مرحله ی سوم با توجه به ورودی فاصله از جلو به عنوان return تابع بازگردانده شود. (این سیستم مستقل از سیستم فاز اول است اما برای کارکرد صحیح باید حتماً فاز اول را به درستی کامل کرده باشید تا چرخش فرمان توسط سیستم فاز اول مدیریت شود.)

در صورت پیاده سازی صحیح، ماشین کمی سریعتر به مقصد خواهد رسید.

نمودار ورودی سنسور جلو :



نمودار خروجی میزان سرعت :



چنانچه در رابطه با پروژه سوالی داشتید میتوانید سوالات خود را از طریق آیدی تلگرام تدریسیاران یا ایمیل تدریسیاری مطرح نمایید:

@msajadcr7

@Mah_rahmani

@mrsl2000

موفق باشید - تیم تدریسیاری