

به نام ایزد منان

تمرین دوم تئوری درس مبانی هوش محاسباتی، «سیستم های فازی»



استاد درس: دکتر عبادزاده

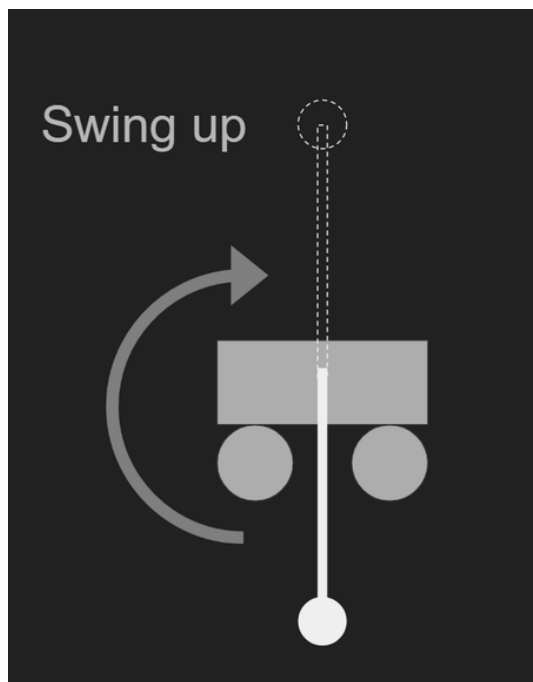
پاییز ۱۴۰۰ – دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر

نکاتی در مورد این تمرین نیاز به توجه و دقت دوستان دارد.

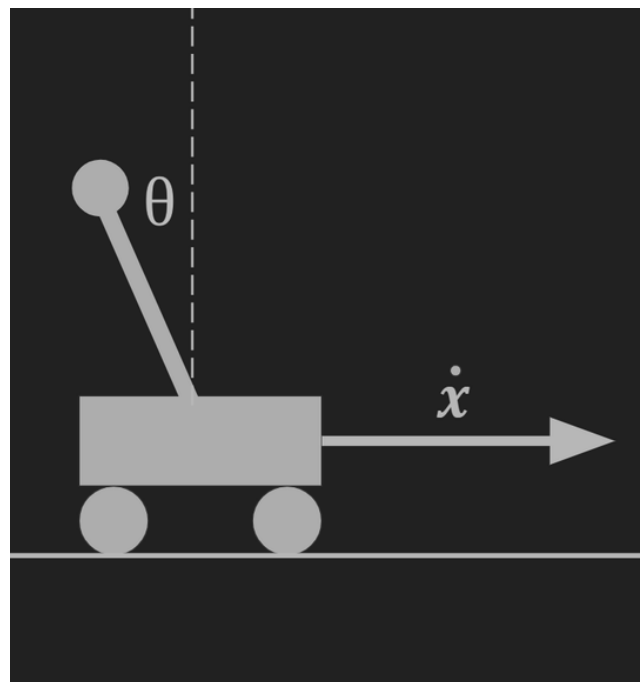
- 1- هرگونه کپی کردن باعث عدم تعلق نمره به تمامی افراد مشارکت کننده در آن می شود.
- 2- آخرین مهلت ارسال تمرین، ساعت ۲۳:۵۵ دقیقه روز **جمعه ۳ دی** می باشد. این زمان با توجه به جمع بندی های صورت گرفته، شرایط و با توجه به سایر تمرین ها در نظر گرفته شده است و قابل تمدید نمی باشد.
- 3- دوستان فایل ارسالی خود را به صورت فشرده و به صورت «شماره دانشجویی_P2» مانند P2_97310000 نام گذاری کنید.
- 4- در صورت هرگونه سوال یا مشکل می توانید با تدریس یاران درس از طریق ایمیل در ارتباط باشید.

ci.1400fall@gmail.com

یکی از کاربرد های مهم منطق فازی در علم مکاترونیک و کنترلر های هوشمند ربات ها می باشد. در این پروژه قصد داریم یکی از مسائل کنترلی بسیار معروف در این حوزه را به کمک منطق فازی حل کنیم. مسئله آونگ معکوس یا همان inverted pendulum یکی از مسائل معروف می باشد که به کمک کنترلر های مختلف مانند PID controller و Fuzzy controller قابل حل می باشد. همچنین به کمک یادگیری تقویتی (reinforcement learning) نیز این مسئله قابل حل می باشد که در صورت علاقه می توانید در مورد آن جستجو کنید. در این مسئله آونگی به یک گاری که تنها در راستای محور x می تواند حرکت کند به صورت معکوس متصل شده است. هدف مسئله ثابت نگه داشتن آونگ در بالای گاری به صورت معکوس می باشد. در شکل های زیر می توان حالت اولیه مسئله و حالت کلی را مشاهده کرد.



حالت اولیه



حالت کلی مسئله

به عبارت دیگر در این مسئله باید سعی کنیم به کمک نیرو x وارده به گاری، زاویه θ را در مقداری نزدیک به ۰ نگه داریم.

ورودی های این مسئله شامل موارد زیر می باشد:

۱- زاویه آونگ با محور عمود بر گاری (Θ) (pa)

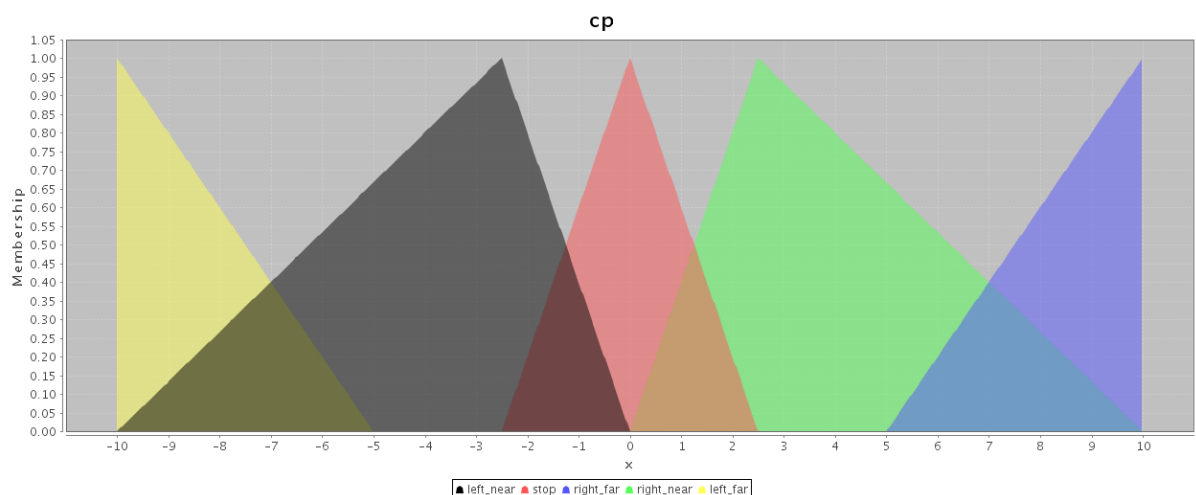
۲- سرعت زاویه ای آونگ (Ω) (pv)

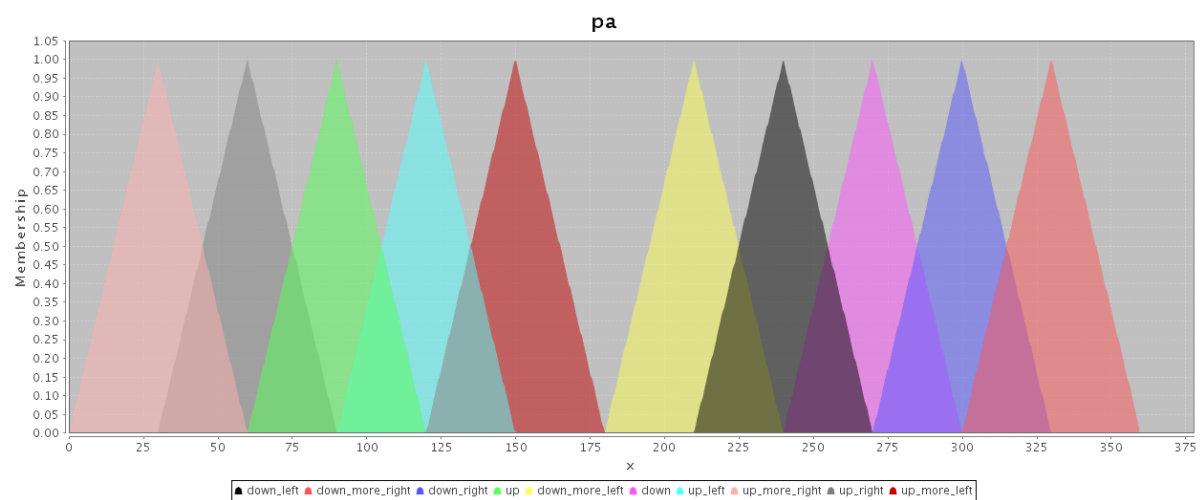
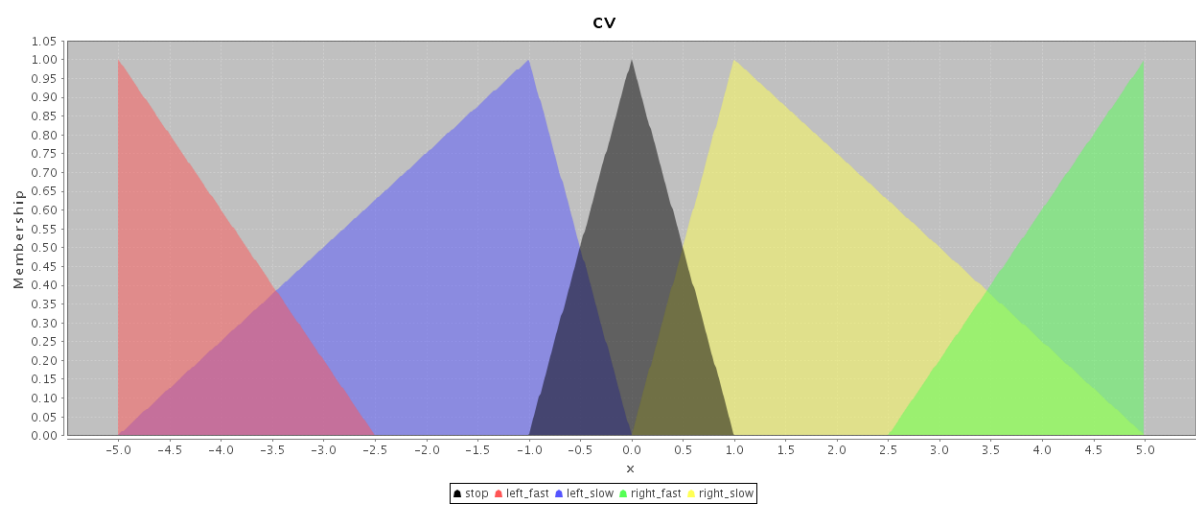
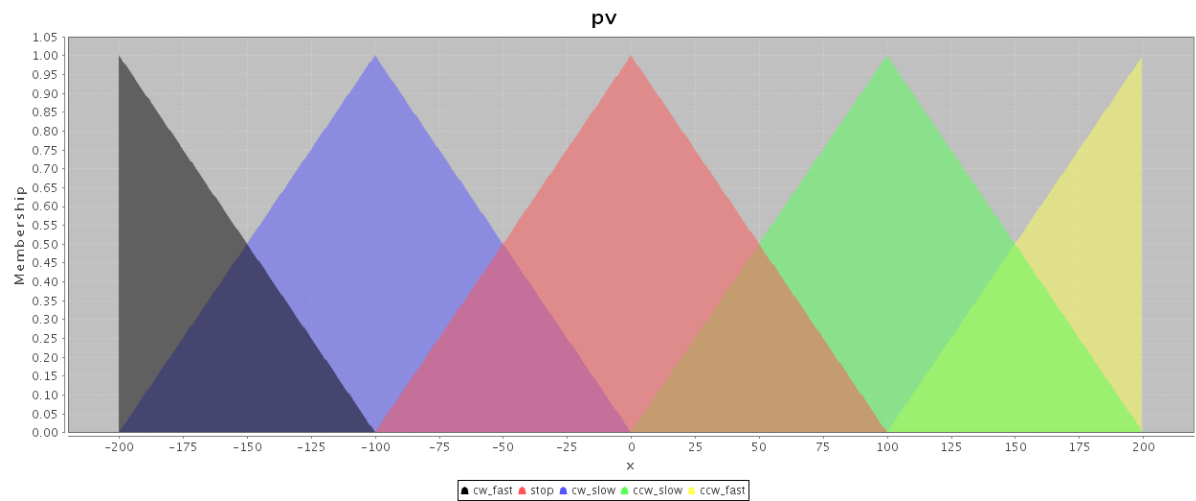
۳- موقعیت گاری (cp)

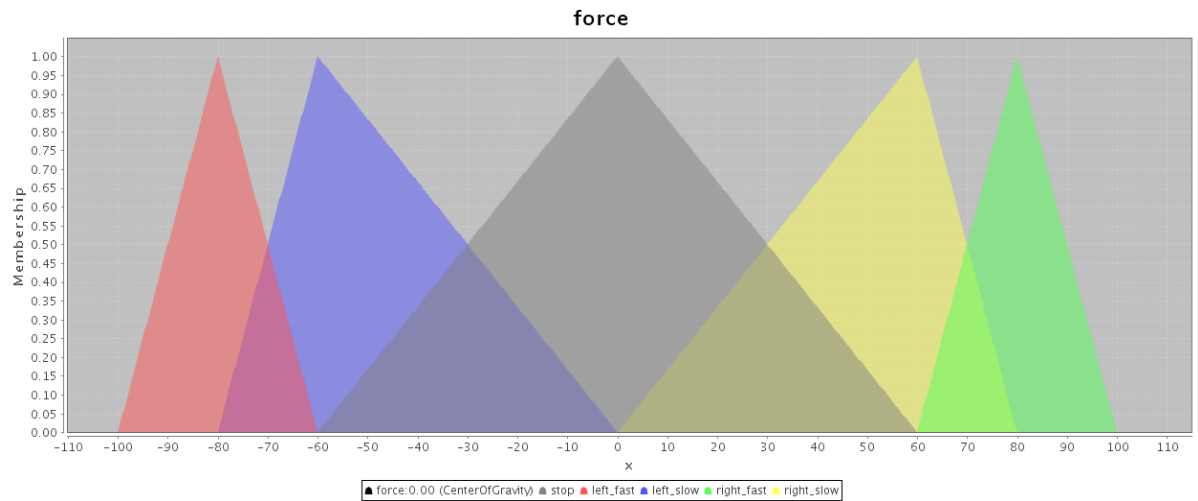
۴- سرعت گاری در راستای محور X (cv)

و در نهایت خروجی مسئله نیروی F می باشد که نیروی وارده به گاری در راستای محور X می باشد. بدیهی است مقادیر گفته شده می تواند مثبت یا منفی باشد که این علامت ها نشان دهنده جهت نیرو ها و یا زاویه ها 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 آمده است:







برای سادگی کار تابع های تعلق به صورت خطی تعریف شده اند و در پیاده سازی می بایست با توجه به شکل های بالا معادله خطوط را بدست آورد. بدیهی است که معادله های بالا معادلات خطوط ساده می باشند که با دو نقطه بدست می آیند. به عنوان مثال برای متغیر نیرو، مجموعه فازی `left_fast` به صورت زیر تعریف می شود:

For $x \in [-100, -80] \rightarrow \text{membership_function}(x) = 0.05x + 5$

For $x \in [-80, -60] \rightarrow \text{membership_function}(x) = -0.05x - 3$

Otherwise $\rightarrow \text{membership_function}(x) = 0$

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 (pa IS up) AND (pv IS ccw_slow) THEN force IS left_slow

* این قانون می گوید اگر زاویه آونگ (ورودی) در مجموعه `up` باشد و سرعت زاویه ای آونگ (ورودی) در مجموعه `ccw_slow` باشد نیرو (خروجی) می بایست در مجموعه `left_slow` باشد.

IF (pa IS up) AND (pv IS ccw_fast) THEN force IS left_fast

IF (pa IS up) AND (pv IS cw_slow) THEN force IS right_slow

حال فرض کنید به ازای زاویه خاصی مقدار تعلق به مجموعه up که در مرحله قبل انجام شده است برابر با ۰.۶ باشد. همچنین به ازای سرعت زاویه خاصی مقدار تعلق به مجموعه ccw_slow برابر ۰.۷ باشد و مقدار تعلق به مجموعه های cw_slow برابر ۰.۱ و مقدار تعلق بقیه مجموعه ها ۰ باشد.

در این حالت مجموعه قوانین بالا به شکل زیر در می آیند:

If 0.6 AND 0.7 then force is left_slow

If 0.6 AND 0 then force is left_fast

If 0.6 AND 0.1 then force is right_slow

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:

$Membership(left_slow) = \min(0.6, 0.7) = 0.6$

$Membership(left_fast) = \min(0.6, 0) = 0$

$Membership(right_slow) = \min(0.6, 0.1) = 0.1$

به مقادیر بدست آمده بالا قدرت هر قاعده گفته می شود. در این حالت قاعده ۱ و ۳ تنها قاعده های فعال شده می باشند. این مرحله خروجی که همان نیرو می باشد را به صورت مجموعه هایی با مقادیر تعلق مختلف بدست می آورد.

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 متغییر نیرو به شکل زیر می باشد:

Left fast = 0

Left slow = 0.6

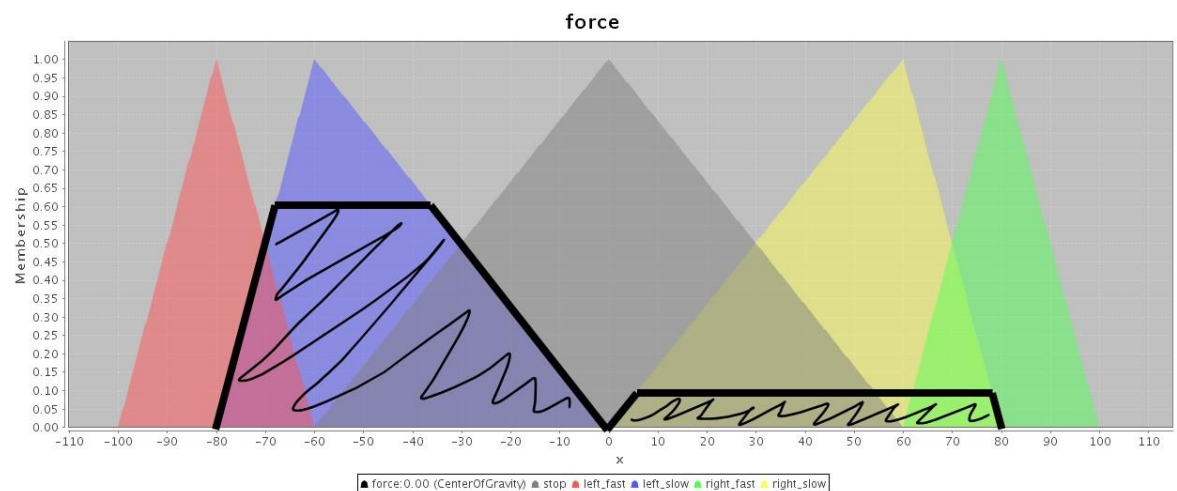
Stop = 0

Right slow = 0.1

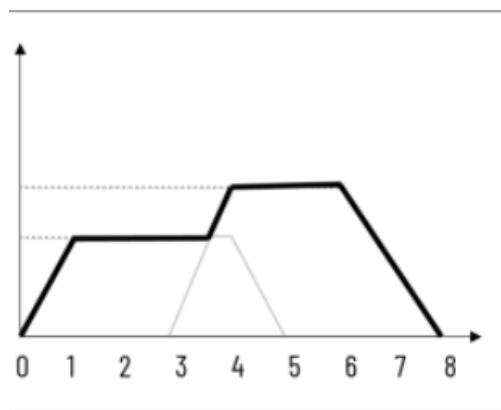
Right fast = 0

* توجه شود در مواردی ممکن است بیش از ۲ قانون فعال شوند و ممکن است چندین مجموعه مقدار تعلق داشته باشند.

ابتدا باید جواب های بدست آمده را با هم ترکیب کنیم. برای اینکار تمام پاسخ ها را با هم OR می کنیم یا به عبارتی max خروجی تمام قواعد را بدست می آوریم. در مثال بالا خروجی به شکل زیر خواهد شد:



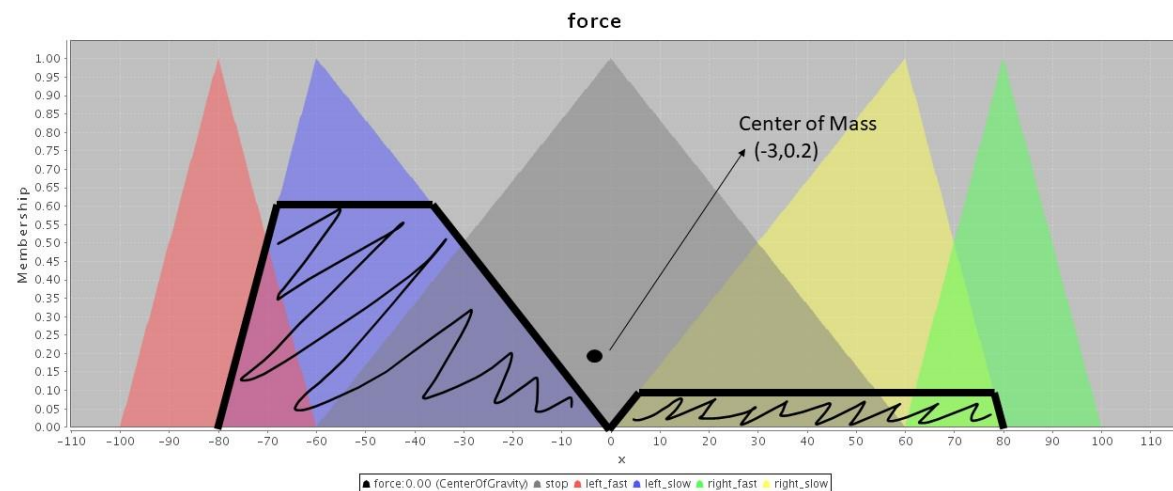
* توجه فرمایید در حالاتی ممکن است شکل ها تو در تو باشند و همانطور که گفته شد max دو شکل (دو



Aggregated fuzzy output

تابع) باید در نظر گرفته شود. مانند شکل زیر:

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 :



با محاسبه مرکز جرم از فضای فازی به فضای مطلق می رویم و جواب نهایی برابر است با ۳- . در نتیجه در سناریو بالا سیستم ما نیرو ۳- نیوتون را به گاری وارد می کند تا آونگ ما به صورت معکوس با زاویه ۰ درجه در بالای گاری ثابت بماند.

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. $\int y(x) \cdot x \, dx / \int y(x) \, dx$ از آن جایی 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.

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

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

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 و حل مسئله آونگ معکوس به صورت تئوری

<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

همچنین در صورت هرگونه ابهام یا پرسش با ایمیل مربوط به تدریساران و یا تلگرام @shayanthrn در تماس باشید.

صورت پروژه:

برای انجام این پروژه، کد ای حاوی شبیه ساز آونگ معکوس به صورت پروژه ضمیمه شده است. برای اجرای این کد لازم است پایتون ۲.۷ بر روی سیستم شما نصب شده باشد. برای دانلود پایتون ۲.۷ به سایت python.org مراجعه بفرمایید.

پس از نصب پایتون نسخه ۲، کاملاً پیشنهاد می شود روند زیر را انجام دهید: (دستورات در محیط cmd نوشته شوند و cmd به صورت run as admin باز شود. اگر از لینوکس استفاده می کنید از terminal استفاده شود)

ابتدا به کمک دستور زیر virtualenv را نصب کنید.

```
Pip install virtualenv
```

سپس به پوشه پروژه رفته و محیط مجازی پایتون ۲ را به کمک دستور زیر ایجاد کنید:

```
Virtualenv -p python2.7 venv
```

حال با توجه به سیستم عامل خودتان دستور مناسب برای فعال سازی محیط مجازی را وارد کنید:

```
Linux -> source venv/bin/activate
```

```
Windows -> .\venv\Scripts\activate
```

با انجام مراحل بالا محیطی مجازی به صورت ایزوله که ورژن پایتون ۲ بر روی آن نصب می باشد ایجاد کرده اید. این کار باعث می شود در صورت وجود نسخه ۳ پایتون بر روی سیستم شما، عملکرد برنامه ها دچار اختلال نشود. برای غیرفعال کردن این محیط نیز کافی است دستور deactivate را وارد کنید.

پس از آماده سازی محیط، می بایست کتاب خانه های مورد نیاز را نصب کنید. کتاب خانه در پوشه

libraries موجود است و تنها کافی است که دستورات زیر را برای نصب وارد کنید:

```
Pip install "library name"
```

با اتمام مراحل بالا، شبیه ساز آماده اجرا می باشد. با اجرای فایل `main.py` شبیه ساز اجرا می شود.

Python main.py

این شبیه ساز به کمک کتاب خانه `pygame` پیاده سازی شده است و جزئیات این پیاده سازی مربوط به این پروژه نمی باشد.

در این شبیه ساز کنترلر فازی به کمک کتابخانه `pyfuzzy` پیاده سازی شده است و در صورت اجرا شبیه ساز، به کمک این کنترلر آونگ به صورت عمودی در می آید. وظیفه شما این می باشد که عملکردی مشابه به این کنترلر را بدون استفاده از کتابخانه و به کمک توضیحات اول پروژه پیاده سازی کنید. برای این منظور در فایل `controller.py` تابع `decide` موجود را پاک کنید. تابعی به همین اسم و با ورودی های مشابه تعریف کنید. خروجی این تابع مقدار نیرو (خروجی مسئله) می باشد که عددی بین ۱۰۰- و ۱۰۰ می باشد. برای بدست آوردن مقدار مطلق ورودی ها از تابع `self._make_input(world)` استفاده کنید. با اجرا این متد دیکشنری ای با ورودی های مذکور به شما داده می شود.

در این دیکشنری مقادیر به شکل زیر می باشند:

`Cp=cart position, Cv=cart velocity, Pa=Pendulum angle, Pv=Pendulum velocity`

در ادامه به کمک توضیحات داده شده در ابتدا، این مقادیر ورودی را فازی کنید.

در مرحله بعد به کمک قواعد نوشته شده در فایل `controller/simple.fcl` استنتاج مناسب را انجام دهید. برای سادگی کار قواعد تنها شامل دو ترم `Pv` و `Pa` می باشند و تنها ورودی های مسئله زاویه آونگ و سرعت زاویه ای می باشد. خروجی مسئله نیز همانطور که گفته شد `force` می باشد.

**** مجموعه قواعد بالا تنها به کمک دو پارامتر می توانند مسئله را تا حد خوبی حل کنند. به عنوان بخش امتیازی شما می توانید از مجموعه قواعد موجود در فایل `controller/complex.fcl` استفاده کنید که در این مجموعه قواعد از پارامتر های سرعت و مکان گاری نیز استفاده شده است. بدیهیست دقت و عملکرد مدل دوم بسیار بهتر می باشد.**

پس از محاسبه خروجی به صورت فازی، مقدار نیرو را به کمک توضیحات داده شده به صورت مطلق بدست آورید. این مقدار به عنوان خروجی تابع decide در نظر گرفته شود. انتظار می رود تا حد خوبی سیستم پیاده سازی شده شما بتواند آونگ را ثابت نگه دارد. گرچه وجود مقداری خطا موردی ندارد و اگر سیستم شما بتواند حتی برای چند لحظه آونگ را ثابت نگه دارد نمره کامل به شما تعلق می گیرد.

به صورت خلاصه:

در فاز اول ورودی ها را به صورت فازی در آورید. (fuzzification)

در فاز دوم خروجی را به صورت فازی به کمک قواعد در آورید. (inference)

در فاز سوم خروجی فازی شده را به صورت مقدار مطلق در آورید. (defuzzification)

استفاده از کتاب خانه 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**. پیاده سازی موجود به کمک کتاب خانه تنها برای فهم بهتر مسئله و مشاهده جواب انجام شده است.

همچنین شما می بایست گزارش کوتاه و یک صفحه ای راجع به پروژه و عملکرد سیستم به همراه عکس خروجی تحویل دهید.

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.