

## به نام خدا

گزارش پروژه آونگ معکوس با منطق فازی

محمد متقی ۹۷۳۱۰۵۷

۳ تابع کلی برای ۳ مرحله اصلی داریم:

۱- تابع fuzzification

در این تابع ورودی دریافت میشود و با استفاده از شکل ها داده شده و محاسبه معادله خط هر کدام تابع تعلق آن ها را به مجموعه فازی حساب می کنیم.

pa : REAL; (\* description='pendulum angle', min=0, max=360,  
unit='degrees' \*)

pv : REAL; (\* description='pendulum angular velocity', min=-  
200, max=200, unit='degrees per second' \*)

خروجی این تابع ۱۵ عدد تابع تعلق pa و pv است.

[up\_more\_right,up\_right,up,up\_left,up\_more\_left,down\_more\_left,do  
wn\_left,down,down\_right,down\_more\_right]

[cw\_fast\_pv,cw\_slow\_pv,stop\_pv,ccw\_slow\_pv,ccw\_fast\_pv]

۲- تابع inference

در این تابع ۱۵ عدد خروجی تابع قبل به عنوان ورودی داده می شود.

با استفاده از ۴۳ قوانین گفته شده ۵ لیست از تابع تعلق خروجی این ۴۳ قانون را به عنوان power مجموعه تعلق فازی force در می آوریم و ماکزیمم آن ۵ لیست تابع تعلق را به عنوان خروجی باز می گردانیم.

### ۳- تابع defuzzify

با استفاده از ۵ ورودی تابع inference مرکز جرم شکل حاصل از برخورد ماکزیمم این ورودی ها و مجموعه فازی force را با روش انتگرال گیری به دست می آوریم و آن را به شکل نیروی خروجی باز می گردانیم.

مثلا برای بررسی یک membership به مجموعه فازی force داریم:

```
points=np.linspace(-100,100,5000)
# print(points)
# exit()
dx=points[1]-points[0]
sum = 0
sum2= 0

for point in points:
    # print(point)

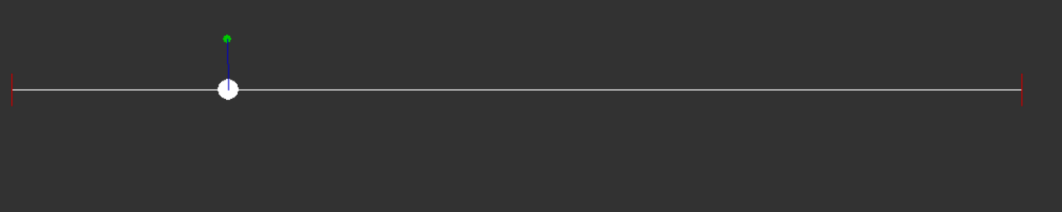
    left_fast = self.left_fast_membership(point)
    # print(left_fast)
    if left_fast>max_left_fast:
        left_fast=max_left_fast
    # print(left_fast)
    # exit()
```

```
def left_fast_membership(self,x):
    if x>=-100 and x<=-80:
        y =0.05*x+5
    elif x>-80 and x<=-60:
        y =-0.05*x-3
    else:
        y = 0
    return abs(y) if y>0 else 0
```

در نهایت خروجی ما به شکل زیر است:

FileEditSelectionViewGoRunTerminalHelpcontroller.zy - Untitled (Workspace) - Visual Studio Code

Fuzzy Inverted Pendulum



```
393     def left_fast_membership(self,x):
394         if x>=-100 and x<=-80:
395             y =0.05*x+5
396         elif x>-80 and x<=-60:
397             y =-0.05*x-3
398         else:
399             y = 0
400         return abs(y) if y>0 else 0
401
402     def left_slow_membership(self,x):
403         if x>=-80 and x<=-60:
404             y =0.05*x+4
405         elif x>-60 and x<=-0:
406             y =-0.016*x
407         else:
408             y = 0
409         # print(y)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

force: -1.3206160178191375  
(-2.096188887781918, -0.5986351951468993)  
force: -0.5986351951468993  
(-1.554600150913702, -0.03964353117972688)  
force: -0.03964353117972688  
(-1.2056298635137406, 0.32489697744740187)  
force: 0.32489697744740187  
(-1.0755782115949029, 0.4678210724889948)  
force: 0.4678210724889948  
[]

master\*

Python 2.7.18 64-bit (venv: venv)

0 0 0

Ln 400, Col 36 (219 selected) Spaces: 4 UTF-8 CRLF Python

1:53 PM  
12/30/2021