فاز اول (Fuzzification)

در این قدم مقادیر مطلق که ورودی ما هستند (pa, pv, cp, cv) داده میشوند و باید توسط تابع های تعلق آنهارا به مقادیر فازی تبدیل کنیم.

که برای تمامی ورودی ها و همچنین force تابع های تعلق در فایل controller.py تعریف شده اند.

نمونه ای از تابع تعلق به صورت زیر است:

```
def force_left_fast(self, x):
    if -100 <= x <= -80:
        return (x / 20) + 5
    elif -80 <= x <= -60:
        return (-x / 20) - 3
    else:
        return 0
```

فاز دوم (Inference)

در این فاز قوانین را بررسی میکنیم و اگر بین rule ها max باشد مینیمم و اگر or باشد ماکسیمم را حساب میکنیم که این موارد نیز در تابع cal_force در فایل controller.py انجام شده اند و قوانین از فایل rules.txt خوانده میشوند.

فاز سوم (Defuzzification)

در این مرحله به کمک استنتاج های انجام شده از rule ها باید مقادیر فازی را به یک مقدار مطلق تبدیل کنیم که همان نیرویی است که باید وارد شود.

برای این مرحله دو تابع integral و second_integral برای محاسبه انتگرال در مرکز جرم در فایل controller.py نوشته شده اند که به صورت زیر میباشند:

```
def integral(self, a, b, start, end):
    return pow(end, 3) * (float(a) / 3) + pow(end, 2) * (float(b) / 2) - (
                pow(start, 3) * (float(a) / 3) + pow(start, 2) * (float(b) / 2))

def second_integral(self, a, b, start, end):
    return pow(end, 2) * (float(a) / 2) + pow(end, 1) * (b) - (pow(start, 2) *
(float(a) / 2) + pow(start, 1) * (b))
```

همچنین محاسبه مرکز جرم نیز در متد cal_force در فایل controller.py انجام شده است.

قسمت امتیازی

در این قسمت قانون های جدید در فایل rules2.txt نوشته شده اند و تابع های تعلق cv و cp نیز در فایل controller.py نوشته شده اند و قوانین استنتاج از rule های جدید نوشته شده نیز در متد cal_force در فایل controller.py قرار داده شده است.