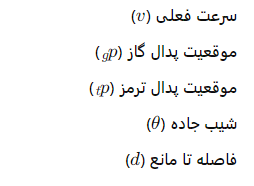
به نام خدا

برای پیاده‌سازی سیستم ترمز و گاز یک ماشین کوچک با استفاده از منطق جدول کارنو و منطق فازی، ابتدا باید مراحل طراحی منطق فازی و جدول کارنو را برای کنترل سیستم‌های ترمز و گاز با ۵ متغیر مشخص کنیم. این کار شامل تعریف متغیرها، تعیین مجموعه‌های فازی، و طراحی منطق تصمیم‌گیری برای هر سیستم است.

1. طراحی سیستم ترمز و گاز با منطق فازی

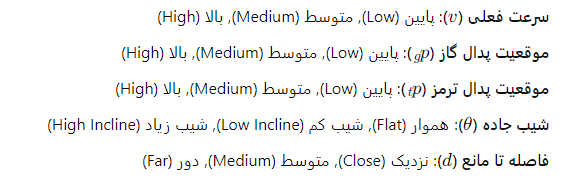
تعریف متغیرهای سیستم ترمز و گاز

برای هر دو سیستم، ۵ متغیر ورودی تعریف می‌شود:



1.1. منطق فازی برای سیستم ترمز

1. \*\*تعریف مجموعه‌های فازی:\*\*



2. \*\*تعریف قوانین فازی برای سیستم ترمز:\*\*

- اگر سرعت بالا و فاصله نزدیک باشد، ترمز زیاد

- اگر سرعت متوسط و فاصله متوسط باشد، ترمز متوسط

- اگر سرعت پایین و فاصله دور باشد، ترمز کم

3. \*\*طراحی تابع عضویت:\*\*

برای هر متغیر، تابع عضویت برای مجموعه‌های فازی تعریف می‌شود. به عنوان مثال، تابع عضویت برای سرعت می‌تواند به صورت خطی و مثلثی باشد.

4. \*\*طراحی جدول تصمیم فازی:\*\*

برای هر ترکیب از مجموعه‌های فازی، مقدار ترمز خروجی تعیین می‌شود. این مقدار می‌تواند توسط توابع عضویت و قوانین فازی ترکیب شود.

1.2. منطق فازی برای سیستم گاز

1. \*\*تعریف مجموعه‌های فازی مشابه سیستم ترمز\*\*

2. \*\*تعریف قوانین فازی برای سیستم گاز:\*\*

- اگر سرعت بالا و شیب جاده کم باشد، گاز زیاد

- اگر سرعت متوسط و موقعیت پدال گاز بالا باشد، گاز متوسط

- اگر سرعت پایین و موقعیت پدال گاز پایین باشد، گاز کم

3. \*\*طراحی تابع عضویت و جدول تصمیم مشابه سیستم ترمز\*\*

2. پیاده‌سازی منطق فازی در Python

2.1. نصب کتابخانه‌های لازم

ابتدا کتابخانه‌های لازم را نصب کنید:

pip install numpy matplotlib scikit-fuzzy

2.2 کد پیاده‌سازی

در زیر یک پیاده‌سازی ساده برای منطق فازی سیستم ترمز و گاز آورده شده است.

----------------------------------------------------------------------------------------

import numpy as np

import matplotlib.pyplot as plt

import skfuzzy as fuzz

from skfuzzy import control as ctrl

# تعریف متغیرهای ورودی برای سیستم ترمز

speed = ctrl.Antecedent(np.arange(0, 101, 1), 'speed')

brake\_pedal = ctrl.Antecedent(np.arange(0, 101, 1), 'brake\_pedal')

gas\_pedal = ctrl.Antecedent(np.arange(0, 101, 1), 'gas\_pedal')

slope = ctrl.Antecedent(np.arange(-10, 11, 1), 'slope')

distance = ctrl.Antecedent(np.arange(0, 101, 1), 'distance')

# تعریف متغیر خروجی

brake = ctrl.Consequent(np.arange(0, 101, 1), 'brake')

# تعریف مجموعه‌های فازی برای ورودی‌ها

speed['low'] = fuzz.trimf(speed.universe, [0, 0, 50])

speed['medium'] = fuzz.trimf(speed.universe, [0, 50, 100])

speed['high'] = fuzz.trimf(speed.universe, [50, 100, 100])

brake\_pedal['low'] = fuzz.trimf(brake\_pedal.universe, [0, 0, 50])

brake\_pedal['medium'] = fuzz.trimf(brake\_pedal.universe, [0, 50, 100])

brake\_pedal['high'] = fuzz.trimf(brake\_pedal.universe, [50, 100, 100])

gas\_pedal['low'] = fuzz.trimf(gas\_pedal.universe, [0, 0, 50])

gas\_pedal['medium'] = fuzz.trimf(gas\_pedal.universe, [0, 50, 100])

gas\_pedal['high'] = fuzz.trimf(gas\_pedal.universe, [50, 100, 100])

slope['flat'] = fuzz.trimf(slope.universe, [-10, 0, 10])

slope['low\_incline'] = fuzz.trimf(slope.universe, [-10, 0, 10])

slope['high\_incline'] = fuzz.trimf(slope.universe, [0, 10, 10])

distance['close'] = fuzz.trimf(distance.universe, [0, 0, 50])

distance['medium'] = fuzz.trimf(distance.universe, [0, 50, 100])

distance['far'] = fuzz.trimf(distance.universe, [50, 100, 100])

brake['low'] = fuzz.trimf(brake.universe, [0, 0, 50])

brake['medium'] = fuzz.trimf(brake.universe, [0, 50, 100])

brake['high'] = fuzz.trimf(brake.universe, [50, 100, 100])

# تعریف قوانین فازی برای سیستم ترمز

rule1 = ctrl.Rule(speed['high'] & distance['close'], brake['high'])

rule2 = ctrl.Rule(speed['medium'] & distance['medium'], brake['medium'])

rule3 = ctrl.Rule(speed['low'] & distance['far'], brake['low'])

# سیستم فازی برای سیستم ترمز

brake\_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])

brake\_sim = ctrl.ControlSystemSimulation(brake\_ctrl)

# ورودی‌های سیستم

brake\_sim.input['speed'] = 70

brake\_sim.input['distance'] = 30

brake\_sim.compute()

print(f"Brake level: {brake\_sim.output['brake']}")

brake.view(sim=brake\_sim)

plt.show()

----------------------------------------------------------------------------------------

2.3. توضیحات

- \*\*تعریف متغیرها:\*\* برای هر متغیر ورودی و خروجی، مجموعه‌های فازی تعریف شده است.

- \*\*تعریف قوانین فازی:\*\* قوانین برای کنترل سیستم ترمز بر اساس وضعیت متغیرهای ورودی تنظیم شده‌اند.

- \*\*پیاده‌سازی و شبیه‌سازی:\*\* سیستم فازی پیاده‌سازی و شبیه‌سازی شده است تا مقدار ترمز مورد نیاز بر اساس ورودی‌های مشخص محاسبه شود.

3. طراحی جدول کارنو برای منطق دیجیتال

برای طراحی جدول کارنو، ابتدا باید سیستم را به صورت منطقی مدل کرده و سپس جدول کارنو را برای آن ترسیم کنیم. این روش معمولاً برای سیستم‌های با تعداد محدودی متغیر و حالت استفاده می‌شود.

3.1. طراحی جدول کارنو

به طور کلی، برای سیستم ترمز و گاز با ۵ متغیر، طراحی جدول کارنو می‌تواند پیچیده باشد، اما برای مثال ساده‌تر، می‌توان از جدول‌های کارنو برای سیستم‌های با تعداد متغیر کمتر استفاده کرد و سپس این رویکرد را برای سیستم‌های پیچیده‌تر گسترش داد.

نتیجه‌گیری

این پیاده‌سازی به شما یک نمای کلی از نحوه طراحی و پیاده‌سازی سیستم‌های ترمز و گاز با منطق فازی و منطق دیجیتال (جدول کارنو) می‌دهد. برای پیاده‌سازی کامل‌تر و دقیق‌تر، می‌توانید به روش‌های پیچیده‌تر و بهینه‌سازی‌های اضافی توجه کنید.