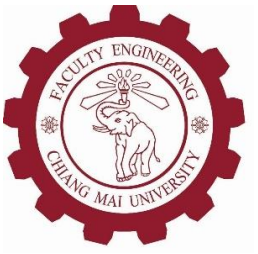


สร้างระบบควบคุมการจราจร
Fuzzy Traffic Control System แบบ Mamdani

โดย

นายศุภกฤต ก๋องคำ

650610858

เสนอ

รศ.ดร.ศันสนีย์ เอื้อพันธ์วิริยะกุล

รายงานนี้เป็นส่วนหนึ่งของรายวิชา 261456

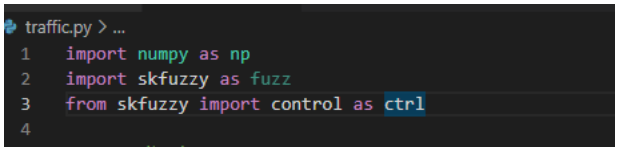
สาขาวิชาวิศวกรรมหุ่นยนต์และปัญญาประดิษฐ์

ภาคเรียนที่1 ปีการศึกษา 2566

มหาวิทยาลัยเชียงใหม่

หัวข้อการเขียน Simulation Program สำหรับ Traffic Control System

สร้างระบบควบคุมการจราจร Fuzzy Traffic Control System แบบ Mamdani โดยใช้ Python และไลบรารี scikit-fuzzy เพื่อจำลองและทดสอบระบบควบคุมการจราจรโดยคำนวณความหนุนของสัญญาณไฟจราจร จากข้อมูลเช่น จำนวนรถแต่ละช่วงเวลาและความเร็วของรถที่ผ่าน



1. ลักษณะการทำงานของระบบ และ Rules ที่ใช้

เราจะสร้างระบบควบคุมการจราจรที่คำนวณความหนุนของสัญญาณไฟจราจรในแต่ละทิศทาง (เช่น ทิศตะวันออก-ตะวันตก และ ทิศเหนือ-ใต้) โดยใช้กฎควบคุม Mamdani เบื้องต้น.

a. กำหนดตัวแปร Input

จำนวนรถในช่วงเวลา (Traffic_Count): เป็นจำนวนรถที่ผ่านในแต่ละช่วงเวลา

ความเร็วของรถในช่วงเวลา (Traffic_Speed): เป็นความเร็วของรถที่ผ่านในแต่ละช่วงเวลา

```
# กำหนดตัวแปร Input
traffic_count = ctrl.Antecedent(np.arange(0, 101, 1), 'Traffic_Count')
traffic_speed = ctrl.Antecedent(np.arange(0, 101, 1), 'Traffic_Speed')
```

b. กำหนดตัวแปร Output

ระดับความหนุนของสัญญาณไฟจราจร (Traffic_Light_Support): เป็นระดับความหนุนที่กำหนดระดับของสัญญาณไฟจราจรในแต่ละทิศทาง

```
# กำหนดตัวแปร Output
traffic_light_support = ctrl.Consequent(np.arange(0, 101, 1), 'Traffic_Light_Support')
```

c. กำหนดฟังก์ชันการเชื่อมั่น:

ในที่นี้ เราสามารถใช้ฟังก์ชันชุดเดียวสำหรับตัวแปร Input และ Output แต่ละตัว

```
# กำหนดฟังก์ชันการเชื่อมั่น
traffic_count.automf(3)
traffic_speed.automf(3)
traffic_light_support.automf(3)
```

d. กำหนดกฎควบคุม

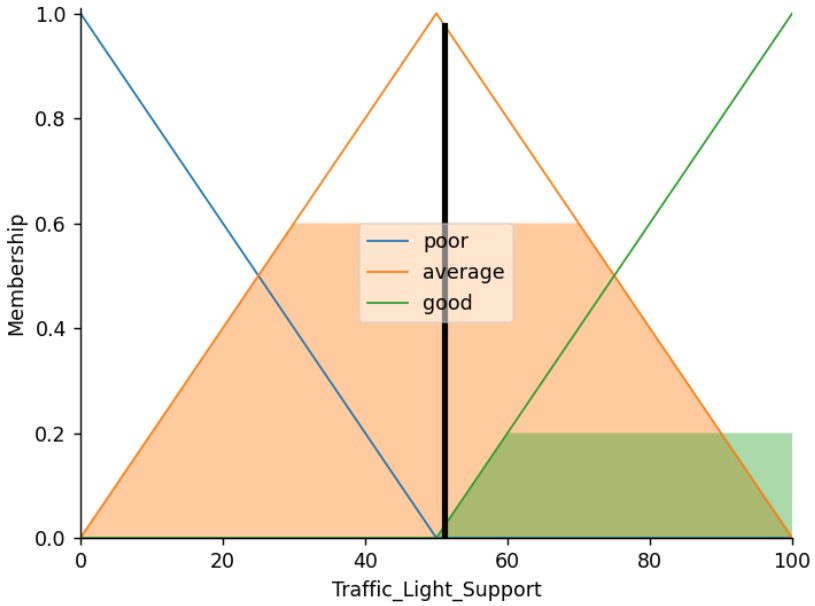
เราสามารถกำหนดกฎควบคุมโดยใช้ความหนุนจากตัวแปร Input เพื่อกำหนดระดับของสัญญาณไฟจราจร โดยใช้ความหนุนนี้เป็นอินพุตในกฎต่าง ๆ ตามทิศทาง

```
# กำหนดกฎควบคุม
rule1 = ctrl.Rule(traffic_count['poor'] & traffic_speed['poor'], traffic_light_support['good'])
rule2 = ctrl.Rule(traffic_count['average'] & traffic_speed['average'], traffic_light_support['average'])
rule3 = ctrl.Rule(traffic_count['good'] & traffic_speed['good'], traffic_light_support['poor'])
```

กฎ	count	speed	light_support
1	poor	poor	good
2	average	average	average
3	good	good	poor

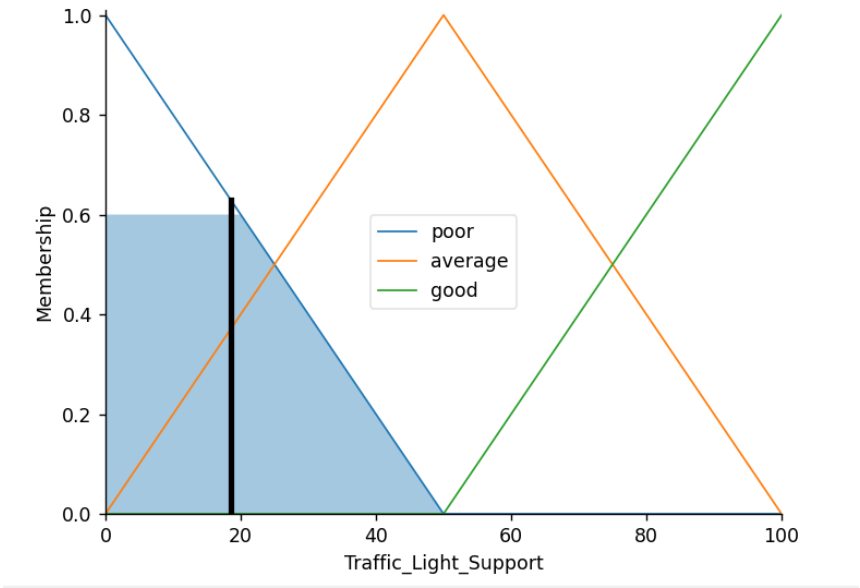
ผลลัพธ์ที่พร้อมแสดงในข้อมูลออกมา ค่า "Traffic Light Support" จะแสดงระดับความหนุนของสัญญาณไฟจราจรตามข้อมูลที่กำหนด โดยผลลัพธ์ของ "Traffic Light Support" จะมีค่าในช่วงที่กำหนดให้ (0-100) และแสดงเป็นกราฟชนิดแสดงค่าของ "Traffic Light Support" ที่คำนวณโดยระบบ Fuzzy Logic

เมื่อทดลองป้อน Traffic_count = 40 และ Traffic_speed =30 จะได้ผลลัพธ์เป็นดังนี้



ผลลัพธ์ที่คำนวณอาจเป็นค่าตัวเลขที่มีความหนุนต่ำกล่าวคือสญาณไฟจราจรแสดง "average" เนื่องจากข้อมูลที่คป้อนอยู่ในสภาวะ "average" สำหรับทั้ง "Traffic Count" และ "Traffic Speed" โดยที่กฎกำหนดว่าความหนุนควรเป็น "average" ในกรณีนี้

เมื่อทดลองป้อน Traffic_count = 100 และ Traffic_speed =80 จะได้ผลลัพธ์เป็นดังนี้



ผลลัพธ์ที่คำนวณเป็นค่าตัวเลขที่มีความหนุนต่ำกล่าวคือสญาณไฟจราจรแสดง "poor" เนื่องจากข้อมูลที่ป้อนอยู่ในสภาวะ "poor" สำหรับทั้ง "Traffic Count" และ "Traffic Speed" โดยที่กฎกำหนดว่าความหนุนควรเป็น "poor"

จากการวิเคราะห์จะได้ว่าการคำนวณนี้อาจให้ผลลัพธ์ที่ต่างกับกันขึ้นอยู่กับกฎควบคุมและฟังก์ชันการเชื่อมั่นที่กำหนด ในกรณีนี้เรากำหนดกฎควบคุมเพียง 3 กฎเพื่อตัดสินใจระดับของ "Traffic Light Support" โดยสัญญาณไฟจราจร โดยที่ "Traffic Count" และ "Traffic Speed" อยู่ในสภาวะต่างๆ จะได้ผลลัพธ์ "Traffic Light Support" อยู่ในสภาวะ ถูกคำนวณในโปรแกรมจำลองผ่านกฎที่กำหนด

สรุปผลจากการสร้างโปรแกรมจำลองการควบคุมจราจรพบว่าเบื้องต้นเราสามารถตรวจสอบระดับความหนุนของจราจรในพื้นที่ตัวอย่างจากนั้นเราสามารถรู้จุดที่เราควรมุ่งเน้นเพื่อแก้ปัญหาการจราจรคือจุดที่มีความหนุนต่ำนั่นเอง

ภาคผนวก

Link : https://drive.google.com/drive/folders/1hIpQJls3o-j7G0pdFOIaxua3HheGAVZG?usp=sharing

traffic.py

```python
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

# กำหนดตัวแปร Input
traffic_count = ctrl.Antecedent(np.arange(0, 101, 1), 'Traffic_Count')
traffic_speed = ctrl.Antecedent(np.arange(0, 101, 1), 'Traffic_Speed')

# กำหนดตัวแปร Output
traffic_light_support = ctrl.Consequent(np.arange(0, 101, 1), 'Traffic_Light_Support')

# กำหนดฟังก์ชันการเชื่อมั่น
traffic_count.automf(3)
traffic_speed.automf(3)
traffic_light_support.automf(3)

# กำหนดกฎควบคุม
rule1 = ctrl.Rule(traffic_count['poor'] & traffic_speed['poor'], traffic_light_support['good'])
rule2 = ctrl.Rule(traffic_count['average'] & traffic_speed['average'], traffic_light_support['average'])
rule3 = ctrl.Rule(traffic_count['good'] & traffic_speed['good'], traffic_light_support['poor'])

# สร้างระบบควบคุม
traffic_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
traffic_system = ctrl.ControlSystemSimulation(traffic_ctrl)



traffic_system.input['Traffic_Count'] = 20
traffic_system.input['Traffic_Speed'] = 30
```

```python
traffic_system.compute()
traffic_sup = round(traffic_system.output['Traffic_Light_Support'],5)

print("Traffic Light Support:", traffic_sup)

traffic_light_support.view(sim=traffic_system, fuzzy_number=101, view='simulation')
plt.show()
```