Fuzzy Inference System for Traffic Light Control

Introduction

Traffic congestion is a common issue in urban areas, leading to inefficiency and frustration among commuters. Traditional traffic light systems operate based on fixed timing, which may not be adaptable to varying traffic conditions. A fuzzy inference system (FIS) offers a more flexible approach to traffic light control by considering multiple factors simultaneously and making decisions based on fuzzy logic.

In this project, we explore a simple FIS designed for traffic light control using Python and the scikit-fuzzy library. The system considers two input variables, namely density and speed, and generates an appropriate output, green_time, to optimize traffic flow.

Fuzzy Inference System Components

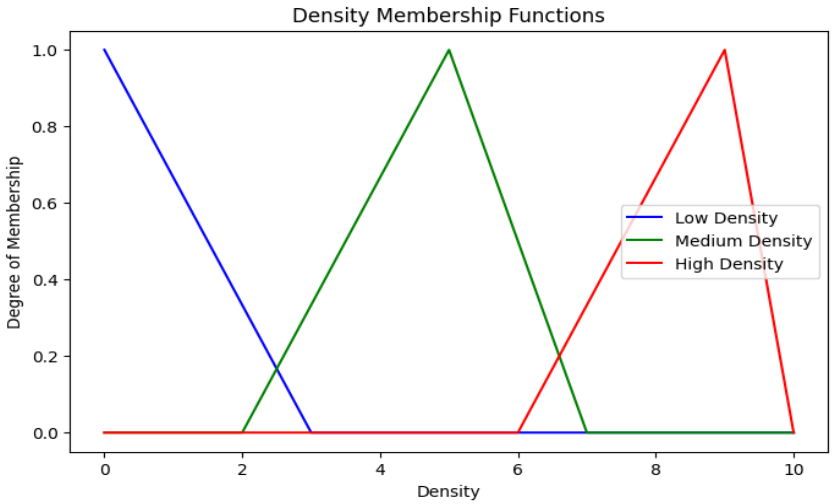
1. **Input Variables**
    a. Density: Represents the density of vehicles on the road, categorized into low, medium, and high.
    b. Speed: Indicates the speed of vehicles, classified as slow, moderate, and fast.

2. **Output Variable**
    a. Green Time: Determines the duration of the green signal for the traffic light, divided into short, medium, and long intervals.

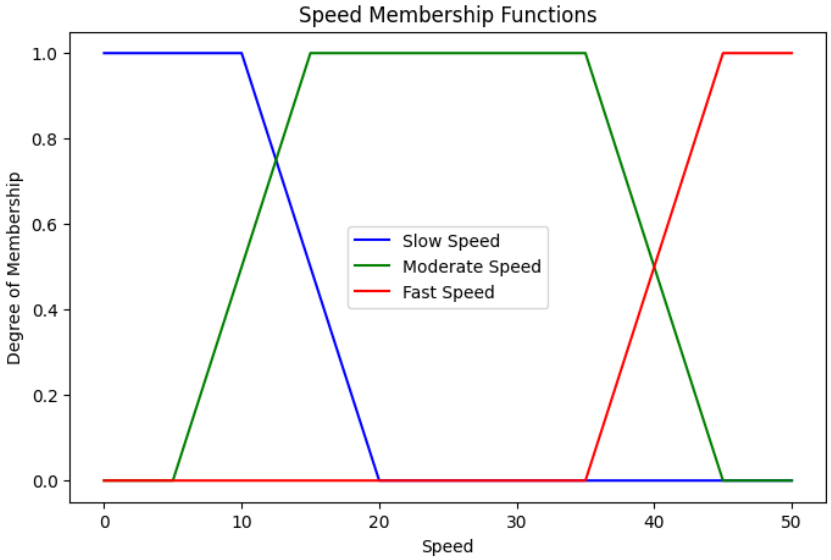
3. **Membership Functions**

    Membership functions define the degree of membership for each input and output variable across their respective universes of discourse. They determine how input values are mapped to fuzzy sets.

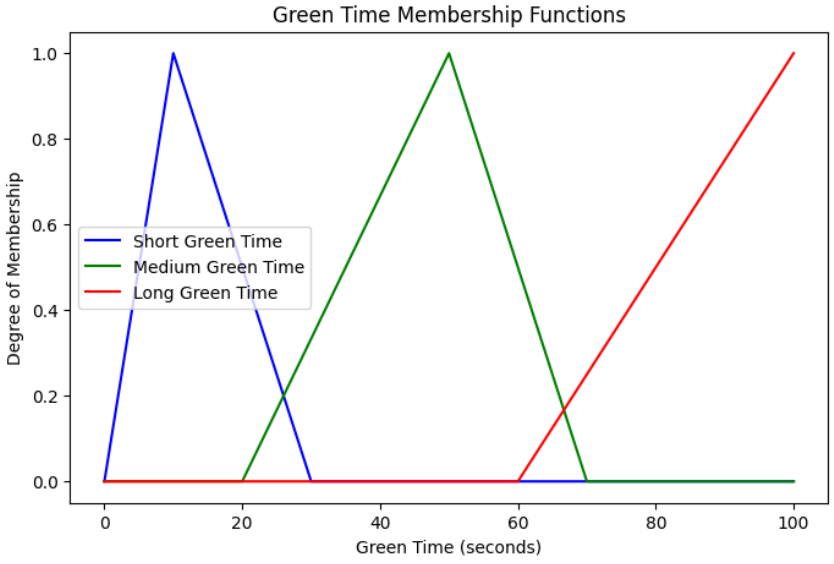
a. Density



b. Speed



c. Green Light



4. **Fuzzy Rules**

   Fuzzy rules govern the decision-making process of the FIS. Each rule specifies how combinations of input variables lead to particular output values.

5. **Fuzzy Inference System**

   The FIS integrates input variables, membership functions, fuzzy rules, and inference mechanisms to compute appropriate output values based on the current traffic conditions.

Python Implementation

Source Code:

```
import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl


# Input variables

density = ctrl.Antecedent(np.arange(0, 11, 1), 'density')

speed = ctrl.Antecedent(np.arange(0, 51, 1), 'speed')


# Membership functions for input variables

density['low'] = fuzz.trimf(density.universe, [0, 0, 3])

density['medium'] = fuzz.trimf(density.universe, [2, 5, 7])

density['high'] = fuzz.trimf(density.universe, [6, 9, 10])


speed['slow'] = fuzz.trapmf(speed.universe, [0, 0, 10, 20])

speed['moderate'] = fuzz.trapmf(speed.universe, [5, 15, 35, 45])

speed['fast'] = fuzz.trapmf(speed.universe, [35, 45, 55, 65])
```

```
# Output variable
green_time = ctrl.Consequent(np.arange(0, 101, 1), 'green_time')

# Membership functions for output variable
green_time['short'] = fuzz.trimf(green_time.universe, [0, 10, 30])
green_time['medium'] = fuzz.trimf(green_time.universe, [20, 50, 70])
green_time['long'] = fuzz.trimf(green_time.universe, [60, 100, 100])

# Fuzzy rules
rule1 = ctrl.Rule(density['low'] & speed['slow'], green_time['long'])
rule2 = ctrl.Rule(density['low'] & speed['moderate'], green_time['medium'])
rule3 = ctrl.Rule(density['low'] & speed['fast'], green_time['short'])
rule4 = ctrl.Rule(density['medium'] & speed['slow'], green_time['medium'])
rule5 = ctrl.Rule(density['medium'] & speed['moderate'], green_time['medium'])
rule6 = ctrl.Rule(density['medium'] & speed['fast'], green_time['short'])
rule7 = ctrl.Rule(density['high'] & speed['slow'], green_time['short'])
rule8 = ctrl.Rule(density['high'] & speed['moderate'], green_time['short'])
rule9 = ctrl.Rule(density['high'] & speed['fast'], green_time['short'])

# Fuzzy inference system
traffic_light_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9])
traffic_light = ctrl.ControlSystemSimulation(traffic_light_ctrl)

# Test the system
traffic_light.input['density'] = 3
traffic_light.input['speed'] = 70
```

```
traffic_light.compute()

green_time_group = 'short' if traffic_light.output['green_time'] <= 30 else 'medium' if traffic_light.output['green_time'] <= 70 else 'long'

print(f"Green time duration: {round(traffic_light.output['green_time'],2)} seconds, Group: {green_time_group}")
```

Output:

```
Green time duration: 14.22 seconds, Group: short
```

Conclusion

The provided Python code demonstrates implementation of a fuzzy inference system for traffic light control. By considering both traffic density and vehicle speed, the system dynamically adjusts the duration of green signals to improve traffic flow and reduce congestion. This approach highlights the effectiveness of fuzzy logic in addressing complex real-world problems.