AIDS LAB EXPERIMENT 08

Name : Shrusti Chintawar
Class : D20B
RollNo : 09

Aim : To design fuzzy control system using Fuzzy tool /library.

Theory :

A **fuzzy control system** is a control method based on **fuzzy logic**, which allows handling uncertain or imprecise information by using degrees of truth rather than binary true/false values. Unlike traditional controllers, fuzzy controllers use **fuzzy sets** and **linguistic rules** to model complex or nonlinear systems where exact mathematical models are difficult to obtain.

The design process includes:

- **Fuzzification:** Converting crisp inputs into fuzzy values.
- **Rule Evaluation:** Applying a set of IF-THEN rules to infer fuzzy outputs.
- **Defuzzification:** Transforming fuzzy outputs into crisp control signals.

Using a **fuzzy tool/library** simplifies this process by providing functions to create membership functions, define rules, and simulate the controller, enabling efficient design and testing of fuzzy control systems.

Designing a Fuzzy Controller
Steps include:

- **Selection of input and output variables.**
- **Defining membership functions** for inputs and outputs.
- **Constructing fuzzy rules** based on expert knowledge or system behavior.
- **Implementation and simulation** using a fuzzy tool/library.
- **Tuning and validation** to ensure desired control performance

Advantages of Fuzzy Control Systems

- Handles nonlinear and complex systems effectively
- Robust to noise and uncertainties.
- Incorporates human expert knowledge through linguistic rules.
- No precise mathematical model required.

Code:

```
[1] !pip install scikit-fuzzy
    !pip install networkx==2.6.3
```

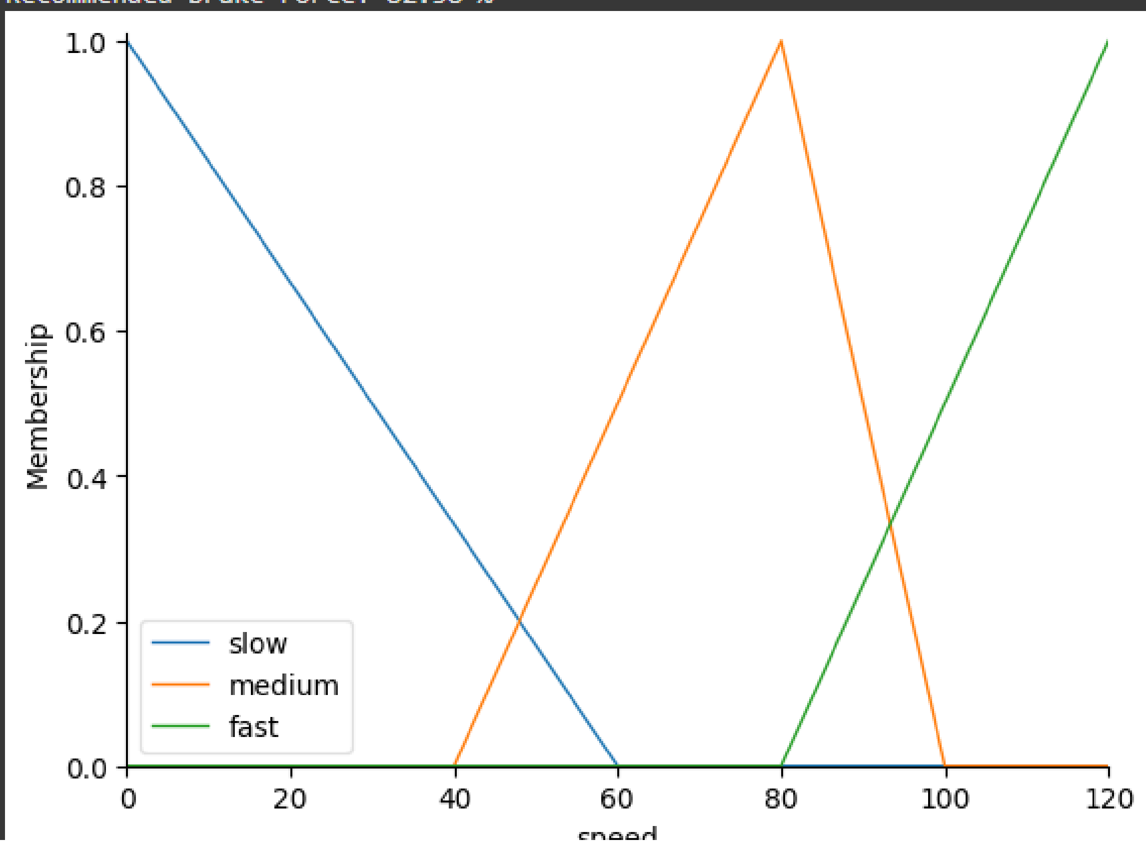
```
Requirement already satisfied: scikit-fuzzy in /usr/local/lib/python3.11/dist-packages (0.5.0)
Requirement already satisfied: networkx==2.6.3 in /usr/local/lib/python3.11/dist-packages (2.6.3)
```

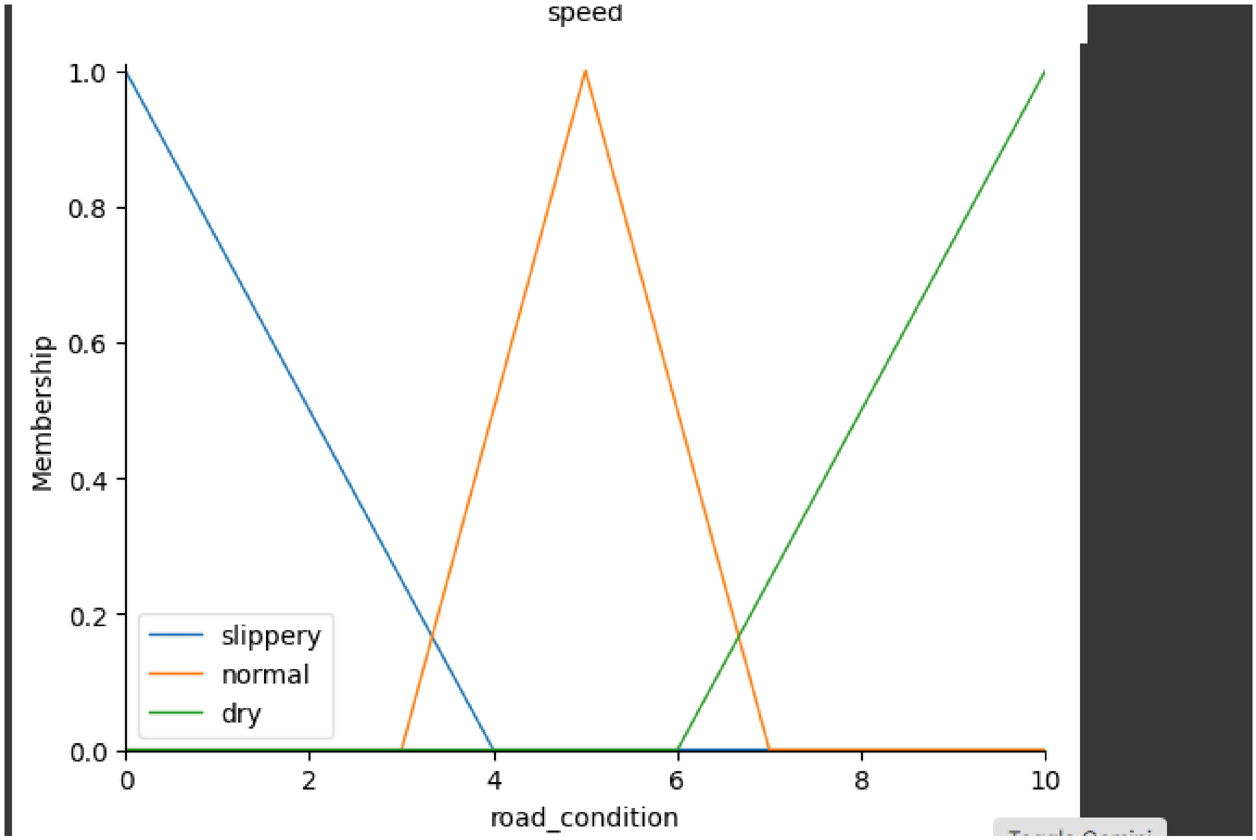
```python
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
```

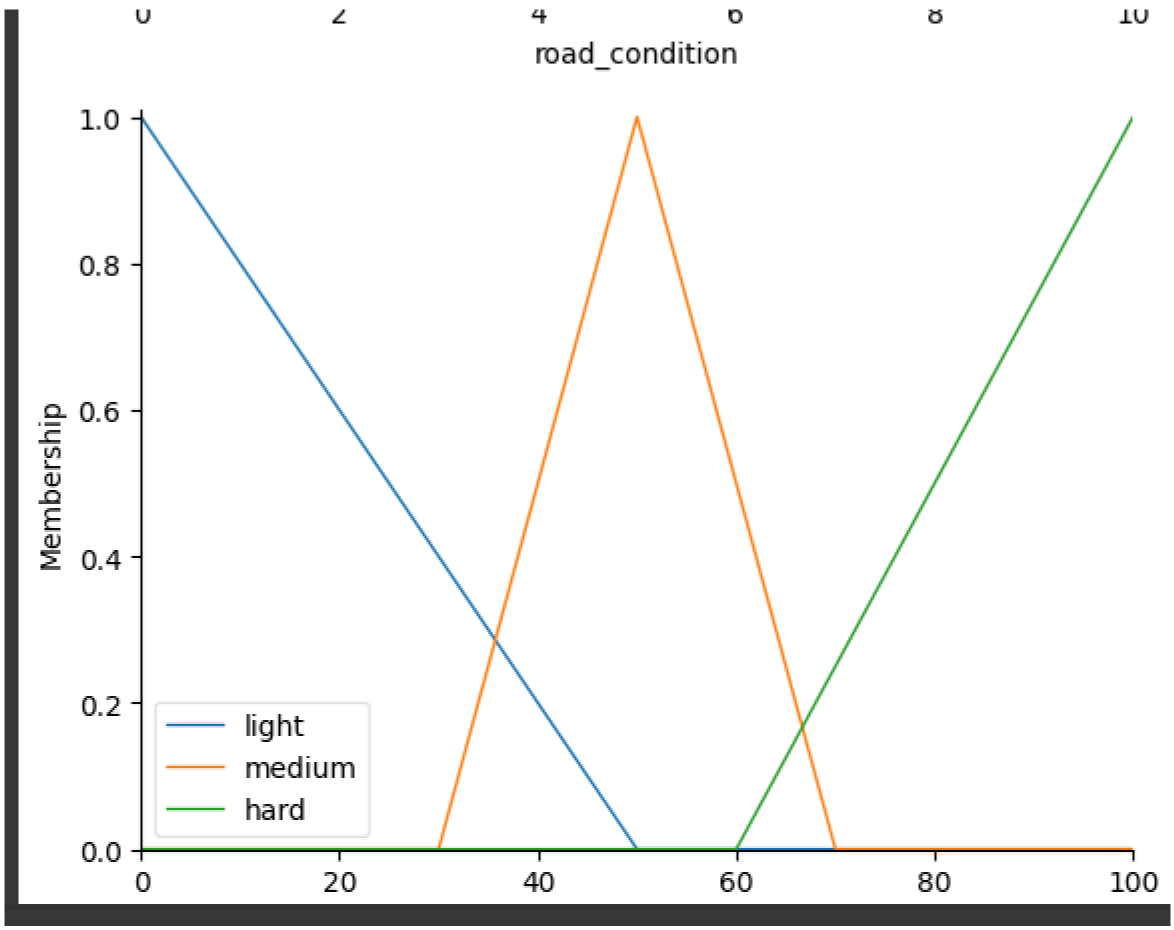
```python
speed = ctrl.Antecedent(np.arange(0, 121, 1), 'speed')           # km/h
road_condition = ctrl.Antecedent(np.arange(0, 11, 1), 'road_condition')  # 0-10
brake_force = ctrl.Consequent(np.arange(0, 101, 1), 'brake_force') # %

# --- Membership Functions ---
# Speed
speed['slow']  = fuzz.trimf(speed.universe, [0, 0, 60])
speed['medium'] = fuzz.trimf(speed.universe, [40, 80, 100])
speed['fast']   = fuzz.trimf(speed.universe, [80, 120, 120])

# Road Condition (higher is better grip)
road_condition['slippery'] = fuzz.trimf(road_condition.universe, [0, 0, 4])
road_condition['normal']   = fuzz.trimf(road_condition.universe, [3, 5, 7])
road_condition['dry']      = fuzz.trimf(road_condition.universe, [6, 10, 10])

# Brake Force
brake_force['light'] = fuzz.trimf(brake_force.universe, [0, 0, 50])
brake_force['medium'] = fuzz.trimf(brake_force.universe, [30, 50, 70])
brake_force['hard'] = fuzz.trimf(brake_force.universe, [60, 100, 100])
```

```python
# --- Define Rules ---
rule1 = ctrl.Rule(speed['fast'] & road_condition['slippery'], brake_force['hard'])
rule2 = ctrl.Rule(speed['fast'] & road_condition['dry'], brake_force['medium'])
rule3 = ctrl.Rule(speed['slow'] & road_condition['dry'], brake_force['light'])
rule4 = ctrl.Rule(speed['medium'] & road_condition['normal'], brake_force['medium'])
rule5 = ctrl.Rule(speed['medium'] & road_condition['slippery'], brake_force['hard'])
rule6 = ctrl.Rule(speed['slow'] & road_condition['slippery'], brake_force['medium'])

# --- Control System ---
brake_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6])
brake_sim = ctrl.ControlSystemSimulation(brake_ctrl)

# --- Example Input ---
brake_sim.input['speed'] = 90           # km/h
brake_sim.input['road_condition'] = 3   # slippery

# --- Compute Output ---
brake_sim.compute()
print("Recommended Brake Force:", round(brake_sim.output['brake_force'], 2), "%")
```

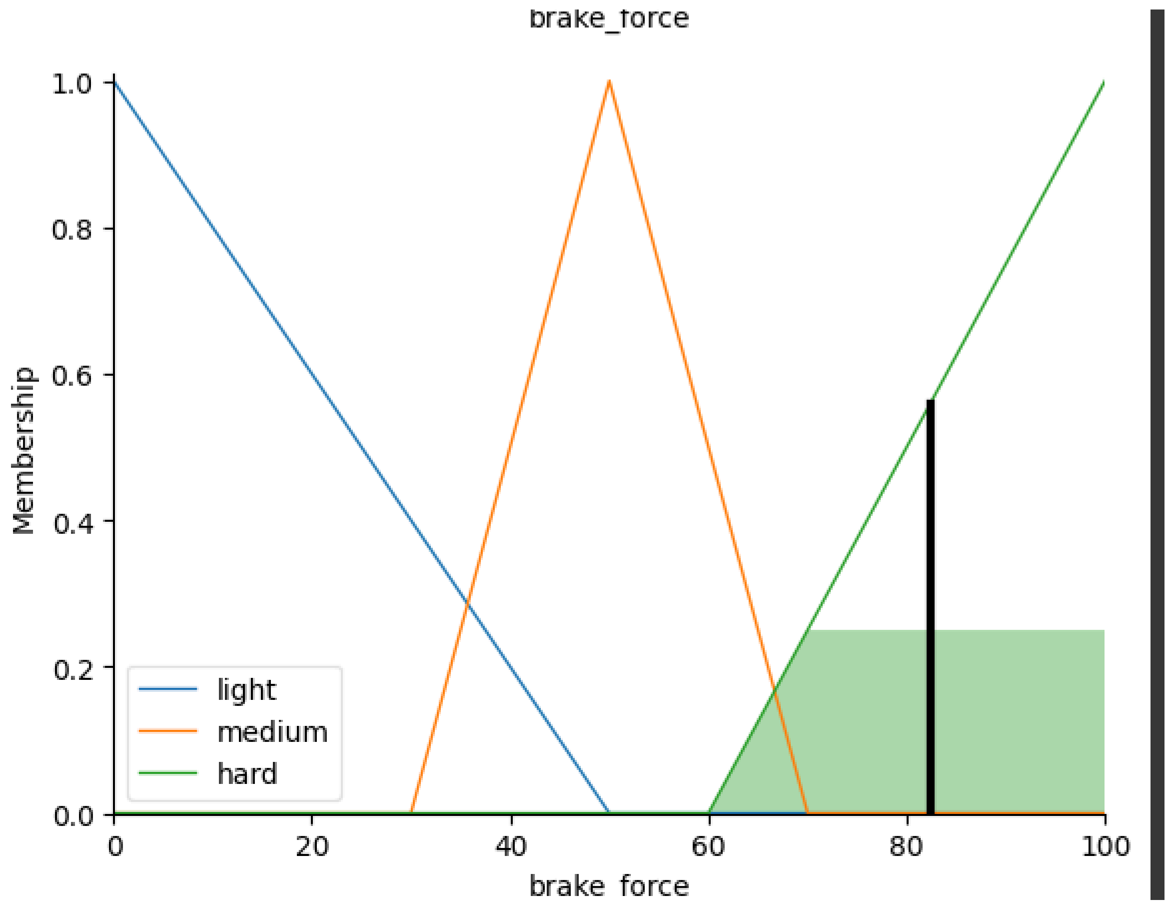
```
Recommended Brake Force: 82.38 %
```




speed
Membership
1.0
0.8
0.6
0.4
0.2
0.0
slippery
normal
dry
0
2
4
6
8
10
road_condition



0
2
4
6
8
10
road_condition
Membership
1.0
0.8
0.6
0.4
0.2
0.0
light
medium
hard
0
20
40
60
80
100



Conclusion:
The experiment shows that fuzzy logic can effectively model real-world decisions, providing adaptive and smooth control for complex systems.