AIDS Lab II
EXP 8

AIM: Design of a Fuzzy Control System Using Fuzzy Tool

THEORY:

Fuzzy logic control systems handle imprecise or uncertain information using linguistic variables rather than precise numerical values. Key components:

1. **Fuzzification:** Converting crisp inputs to fuzzy sets using membership functions
2. **Fuzzy Inference:** Applying if-then rules to determine system behavior
3. **Defuzzification:** Converting fuzzy outputs back to crisp values

Applications include:

- Temperature control systems
- Automotive systems (ABS, transmission control)
- Consumer electronics (washing machines, air conditioners)
- Industrial automation

Theoretical foundations:

- Fuzzy Set Theory: Extends classical set theory to handle partial membership
- Linguistic Variables: Terms like "hot," "cold" instead of precise numbers
- Membership Functions: Define how each point is mapped to a membership value (0-1)
- Mamdani Inference: Common fuzzy inference method using min-max operations

Code:

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

# Create fuzzy variables
temperature = ctrl.Antecedent(np.arange(0, 101, 1), 'temperature')
cooling_power = ctrl.Consequent(np.arange(0, 101, 1), 'cooling_power')

# Define membership functions for temperature
temperature['cold'] = fuzz.trimf(temperature.universe, [0, 0, 25])
temperature['cool'] = fuzz.trimf(temperature.universe, [15, 25, 35])
temperature['moderate'] = fuzz.trimf(temperature.universe, [25, 35, 45])
temperature['warm'] = fuzz.trimf(temperature.universe, [35, 45, 55])
temperature['hot'] = fuzz.trimf(temperature.universe, [45, 100, 100])

# Define membership functions for cooling power
cooling_power['low'] = fuzz.trimf(cooling_power.universe, [0, 0, 30])
cooling_power['medium'] = fuzz.trimf(cooling_power.universe, [20, 40, 60])
cooling_power['high'] = fuzz.trimf(cooling_power.universe, [50, 100, 100])
```

```
# Visualize membership functions
temperature.view()
cooling_power.view()
plt.show()

# Define fuzzy rules
rule1 = ctrl.Rule(temperature['cold'], cooling_power['low'])
rule2 = ctrl.Rule(temperature['cool'], cooling_power['low'])
rule3 = ctrl.Rule(temperature['moderate'], cooling_power['medium'])
rule4 = ctrl.Rule(temperature['warm'], cooling_power['medium'])
rule5 = ctrl.Rule(temperature['hot'], cooling_power['high'])

# Create control system
cooling_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5])
cooling_sim = ctrl.ControlSystemSimulation(cooling_ctrl)

# Test with sample temperatures
test_temperatures = [15, 25, 35, 45, 60]
results = []

for temp in test_temperatures:
    cooling_sim.input['temperature'] = temp
    cooling_sim.compute()
    results.append(cooling_sim.output['cooling_power'])
    print(f"Temperature: {temp}°C -> Cooling Power: {cooling_sim.output['cooling_power']:.2f}%")

# Visualize results
plt.figure(figsize=(10, 6))
plt.plot(test_temperatures, results, 'bo-')
plt.xlabel('Temperature (°C)')
plt.ylabel('Cooling Power (%)')
plt.title('Fuzzy Control System Response')
plt.grid(True)
plt.show()
```

Output:

```
Temperature: 15°C -> Cooling Power: 16.67%
Temperature: 25°C -> Cooling Power: 16.67%
Temperature: 35°C -> Cooling Power: 40.00%
Temperature: 45°C -> Cooling Power: 50.00%
Temperature: 60°C -> Cooling Power: 72.22%
```

CONCLUSION:

This experiment demonstrates the design and implementation of a fuzzy logic control system for temperature regulation. Key findings:

1. **Handling Uncertainty**: Fuzzy systems effectively manage imprecise inputs and gradual transitions between states
2. **Natural Modeling:** Linguistic rules mimic human reasoning (e.g., "if hot, then high cooling")
3. **Smooth Control:** Output changes gradually rather than in abrupt steps

Fuzzy control systems are particularly valuable in applications where:

- Precise mathematical models are unavailable
- Human expertise is available in linguistic form
- Smooth control response is preferred over on/off switching

Future work could explore:

- Adaptive fuzzy systems that learn from data
- Hybrid neuro-fuzzy systems combining neural networks with fuzzy logic
- Multi-input systems considering humidity, occupancy, etc.
- Real-time implementation with hardware interfaces

This approach provides a robust foundation for building intelligent control systems that handle real-world complexity and uncertainty effectively.