Assignment 1 - Group 3 Hints: Fuzzy Control System for Smart Home Climate Control

Task Description

Develop a fuzzy control system for an automated climate control system in a smart home, adjusting temperature and humidity based on external weather conditions and occupant preferences.

Hints for Implementation

Import skfuzzy Library

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

Define Ranges for Inputs and Output

Hint: Define numerical ranges for weather conditions, occupant comfort levels, and HVAC actions.

```
weather_conditions = np.arange(0, 101, 1)
comfort_levels_temp = np.arange(0, 101, 1)
comfort_levels_humidity = np.arange(0, 101, 1)
hvac_actions = np.arange(0, 101, 1)
```

Create Fuzzy Sets and Membership Functions

Hint: Define fuzzy sets for each input and output using appropriate membership functions.

```
# Example for weather conditions
weather_sunny = fuzz.trimf(weather_conditions, [0, 0, 50])
# Continue for other weather conditions, comfort levels, and HVAC actions
```

Formulate Fuzzy Rules

Hint: Create rules that combine weather conditions and comfort levels to determine HVAC actions.

```
# Example Rule
rule1 = ctrl.Rule(weather_conditions['sunny'] & comfort_levels_temp['warm'], hvac_actions['cool'])
# Develop more rules for various scenarios
```

System Simulation

Hint: Create a control system and simulate it with various input scenarios.

```
climate_ctrl = ctrl.ControlSystem([rule1, rule2, ...])  # Include all rules
climate_simulation = ctrl.ControlSystemSimulation(climate_ctrl)
```

Test the System

Hint: Experiment with different combinations of weather conditions and comfort preferences.

```
climate_simulation.input['weather_conditions'] = 80  # Example value
climate_simulation.input['comfort_levels_temp'] = 40  # Example value
climate_simulation.compute()
print(climate_simulation.output['hvac_actions'])
```

Visualization

Hint: Use matplotlib.pyplot to visualize the membership functions and the system's response.

```
import matplotlib.pyplot as plt
weather_conditions.view()  # Visualize weather condition membership functions
# Add similar lines for comfort levels and HVAC actions
```

Submission Guidelines

Submit a Python script that efficiently manages the climate in the smart home, considering both weather conditions and occupant preferences.