Ex No.6: *Implementation of uncertain methods for an Application (Fuzzy Logic)*

Aim:

To implement Fuzzy logic in python and find the graph of temperature, humidity and speed in different conditions.

Algorithm:

1. Locate the input, output, and state variables of the plane under consideration. I 2. Split the complete universe of discourse spanned by each variable into a number of fuzzy subsets, assigning each with a linguistic label. The subsets include all the elements in the universe.
3. Obtain the membership function for each fuzzy subset.
4. Assign the fuzzy relationships between the inputs or states of fuzzy subsets on one side and the output of fuzzy subsets on the other side, thereby forming the rule base. 5. Choose appropriate scaling factors for the input and output variables for normalizing the variables between [0, 1] and [-1, I] interval.
6. Carry out the fuzzification process.
7. Identify the output contributed from each rule using fuzzy approximate reasoning. 8. Combine the fuzzy outputs obtained from each rule.
9. Finally, apply defuzzification to form a crisp output.

Optimization Technique:

1. Decomposing the large-scale system into a collection of various subsystems.
2. Varying the plant dynamics slowly and linearizing the nonlinear plane dynamics about a set of operating points.

3. Organizing a set of state variables, control variables, or output features for the system under consideration.
4. Designing simple P, PD, PID controllers for the subsystems. Optimal controllers can also be designed.

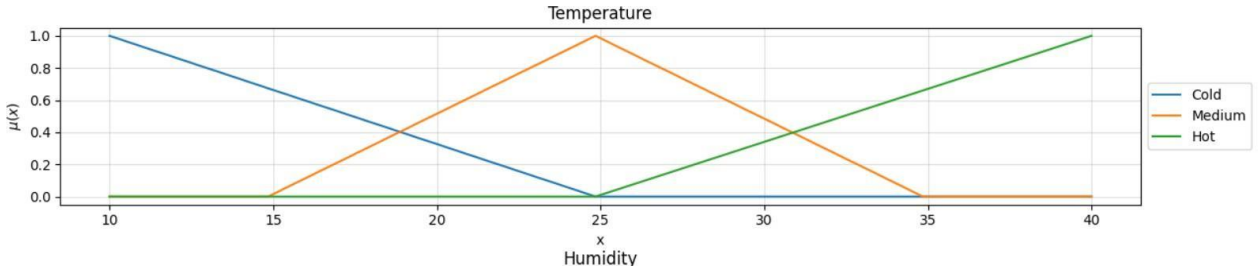
Code:

```
from fuzzy_system.fuzzy_variable_output
import FuzzyOutputVariable from
fuzzy_system.fuzzy_variable_input import
FuzzyInputVariable
# from
fuzzy_system.fuzzy_variable
import FuzzyVariable from
fuzzy_system.fuzzy_system
import FuzzySystem temp =
FuzzyInputVariable('Temperature',
10, 40, 100)
temp.add_triangular('Cold', 10, 10, 25)
temp.add_triangular('Medium', 15, 25, 35)
temp.add_triangular('Hot', 25, 40, 40)
humidity = FuzzyInputVariable('Humidity', 20, 100, 100)
humidity.add_triangular('Wet', 20, 20, 60)
humidity.add_trapezoidal('Normal', 30, 50, 70, 90)
humidity.add_triangular('Dry', 60, 100, 100)
motor_speed = FuzzyOutputVariable('Speed', 0, 100, 100)
motor_speed.add_triangular('Slow', 0, 0, 50)
motor_speed.add_triangular('Moderate', 10, 50, 90)
```

```
motor_speed.add_triangular('Fast', 50, 100, 100)

system = FuzzySystem()
system.add_input_variable(temp)
system.add_input_variable(humidity)
system.add_output_variable(motor_speed)

system.add_rule(
                { 'Temperature':'Cold',
                        'Humidity':'Wet' },
                { 'Speed':'Slow'})

system.add_rule(
                { 'Temperature':'Cold',
                Humidity':'Normal' },
                { 'Speed':'Slow'})

system.add_rule(
                { 'Temperature':'Medium', 'Humidity':'Wet' },
                { 'Speed':'Slow'})

system.add_rule(
                { 'Temperature':'Medium', 'Humidity':'Normal' },
                { 'Speed':'Moderate'})

system.add_rule(
                { 'Temperature':'Cold',
                        'Humidity':'Dry' },
                { 'Speed':'Moderate'})
```

```
system.add_rule(
                { 'Temperature':'Hot',
                        'Humidity':'Wet' },
                { 'Speed':'Moderate'})

system.add_rule(
                { 'Temperature':'Hot',
                        'Humidity':'Normal' },
                { 'Speed':'Fast'})

system.add_rule(
                { 'Temperature':'Hot',
                        'Humidity':'Dry' },
                { 'Speed':'Fast'})
system.add_rule(
                { 'Temperature':'Medium', 'Humidity':'Dry' },
                { 'Speed':'Fast'})

output = system.evaluate_output({
                                'Temperature':18, 'Humidity':60
                })

print(output) system.plot_system()
```

Output:



Result: We have successfully implemented fuzzy uncertainty problem using matplotlib and output is received.