Name: Selas Moro

Roll No.: 39 **BE CMPN B3** **PID:182074**

Experiment No. 6

Fuzzy Inference System Design

Aim: To design a fuzzy inference system.

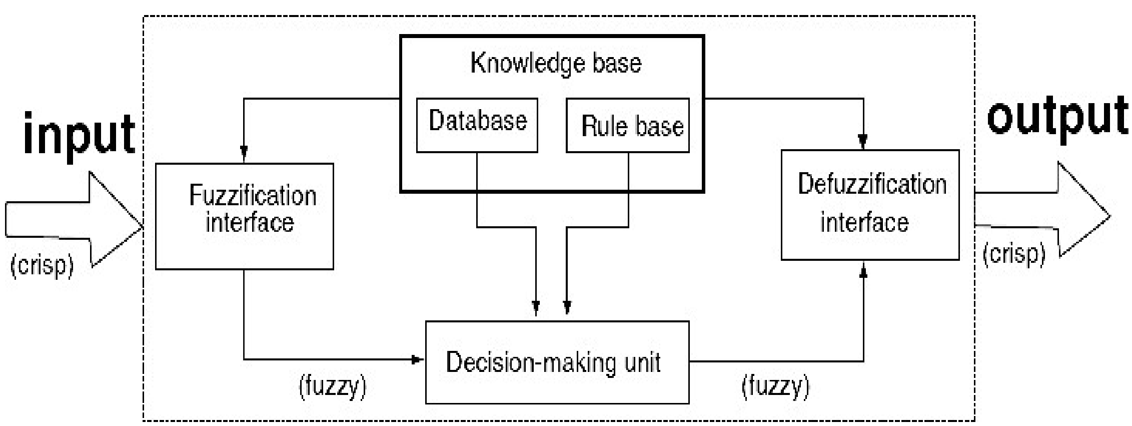
Theory:

Fuzzy Inference System (FIS)

A Fuzzy Inference System (FIS) is a way of mapping an input space to an output space using fuzzy logic.FIS uses a collection of fuzzy membership functions and rules, instead of Boolean logic, to reason about data.

FIS consists of four main components:

1. Fuzzification interface
2. Rule base,
3. Database,
4. Decision-making unit and Defuzzification interface.



Function of each block

1. Rule base : contains a number of fuzzy IF–THEN rules
2. Database : defines the membership functions of the fuzzy sets used in the fuzzy rules
3. Decision-making unit: performs the inference operations on the rules.
4. Fuzzification interface : transforms the crisp inputs into degrees of match with linguistic values; and

Defuzzification interface : transforms the fuzzy results of the inference into a crisp output Firstly, a crisp set of input data are gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is known as fuzzification. Afterwards, an inference is made based on a set of rules. Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions, in the defuzzification step.

Steps in designing Fuzzy Inference Systems

1. Define the linguistic variables and terms (initialization)
2. Construct the membership functions (initialization)
3. Construct the rule base (initialization)
4. Fuzzification
5. Inference (Evaluate the rules in the rule base and combine the results of each rule)
6. Defuzzification

Problem Statement

Implementation of a Fuzzy Controller to determine speed of Fan and Compressor of an air conditioner

Steps in designing Fuzzy Inference Systems for Washing Machine Problem Algorithm

1. **Identify a problem statement, where you can apply fuzzy logic.**
   Implementation of a Fuzzy Controller to determine speed of Fan and Compressor of and air conditioner

2. **Define Input and Output descriptors and linguistic variables for each input**

and output

Consider the washing machine FIS.

Let speed of fan is the linguistic variable which represents the speed of fan. Then, the set of decompositions for the linguistic variable dirt is P(t) = {slow , medium ,fast }.

Let compreesor is the linguistic variable which represents the amount ofspeed of compressor..

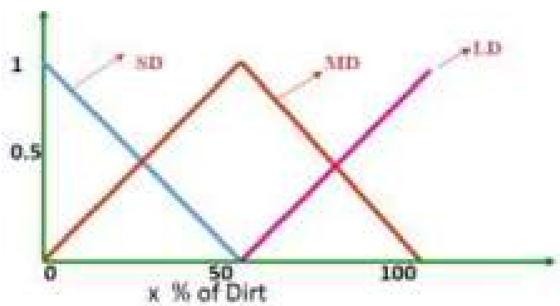
Then, the set of decompositions for the linguistic variable grease is G(t) = { small, medium, fast}.

Let Cooling Time© is the linguistic variable which represents the amount of time required to determine speed of Fan. Then, the set of decompositions for the linguistic is T(t) = {slow, medium, fast }.

3. **Construct membership functions for input and output**

Membership functions are used in the fuzzification and defuzzification steps of a FLS, to map the non-fuzzy input values to fuzzy linguistic terms and vice versa. A membership function is used to quantify a linguistic term.

DIRT



GREASE

$$
\mu_{grease(y)} = \begin{cases} \mu_{SG}(y) = (50 - y) / 50 & 0 \le y \le 50 \\ \mu_{MG}(y) = (y) / 50 & 0 \le y \le 50 \\ \qquad (100 - y) / 50 & 50 \le y \le 100 \\ \mu_{LG}(y) = (y - 50) / 50 & 50 \le y \le 100 \end{cases}
$$

WASHTIME

3. Construct the Fuzzy Rule Base

Grease →

Dirt ↓

	SG	MG	LG
SD	VS	M	L
MD	S	M	L
LD	M	L	VL

4. Fuzzification

Converts the crisp input to a linguistic variable using the membership functions stored in the fuzzy knowledge base.

- $\mu_{MD}(60) = (100 - 60) / 50 = 4/5$
- $\mu_{LD}(60) = (60 - 50) / 50 = 1/5$
- $\mu_{MG}(70) = (100 - 70) / 50 = 3/5$
- $\mu_{LG}(70) = (70 - 50) / 50 = 2/5$

- $\mu_{MD}(60) = (100 - 60) / 50 = 4/5$
- $\mu_{LD}(60) = (60 - 50) / 50 = 1/5$
- $\mu_{MG}(70) = (100 - 70) / 50 = 3/5$
- $\mu_{LG}(70) = (70 - 50) / 50 = 2/5$

5. Inference

The evaluations of the fuzzy rules and the combination of the results of the individual rules are performed using fuzzy set operations. The mostly used operations are OR and AND operators which are max and min, respectively. The results of individual rules can be combined in different ways.

4. **Construct Rule Base.**

Size of plate

grease

	small	medium	large
small	**very small**	**small**	**medium**
medium	**medium**	**medium**	**large**
high	**large**	**medium**	**very large**

5. **Do rule evaluation for 3 cases : low , medium and high**

Case I: When dirt is small and plate size is small

Example: Given Input 12% of Plate size and 10% of Grease

- $\mu_{SMALL}(10) = (50 - 10)/50 = 0.8$
- $\mu_{MEDIUM}(10) = 10/50 = 0.2$
- $\mu_{SMALL}(12) = (50-12)/50 = 0.76$
- $\mu_{MEDIUM}(12) = 12/50 = 0.24$

Applying fuzzified inputs on rulebase

	$\mu_{SMALL}(8)=$ 0. 76	**$\mu_{MEDIUM}(8)=0$ .24**	**0**
$\mu_{SMALL}(10)=$ 0 .8	**$\mu_{VS}(z)$**	**$\mu_{S}(z)$**	**0**
$\mu_{MEDIUM}(10)=$ 0.2	**$\mu_{M}(z)$**	**$\mu_{M}(z)$**	**0**
0	**0**	**0**	**0**

Appling Min Operation

	$\mu_{SG}(8)$=0.84	$\mu_{MG}(8)$=0.16	0
$\mu_{SP}(10)$=0.8	0.76	0.24	0
$\mu_{MP}(10)$=0.2	0.2	0.2	0
0	0	0	0

Applying Max Operation

The fuzzy set = {0.76, 0.24, 0.2, 0.2}. The max of the fuzzy set that is 0.76 which corresponds to $\mu_{VS}(z)$.

Defuzzification

$\mu_{VS}(z) = (20-z) / 20$

Substitute 0.76

in $\mu_{VS}(z)$

z = 4.8 minutes

Thus 4.8 minutes is the wash time required for 10% of grease and 12% of size of plate

Case II: When plate size is large and grease is large

Example: Given Input 80% of Dirt and 80% of Grease

- $\mu_{MEDIUM}(80) = (100-80) /50 = 0.4$
- $\mu_{LARGE}(80) = (80-50)/50 = 0.6$
- $\mu_{MEDIUM}(80)= (100-80) /50 = 0.4$
- $\mu_{LARGE}(80)= (80-50) /50 = 0.6$

Applying fuzzified inputs on rulebase

	0	**$\mu_{MEDIUM}(80)$= 0.4**	**$\mu_{LARGE}(80)$ = 0.6**
0	**0**	**0**	**0**
$\mu_{MEDIUM}(80)$= 0.4	**0**	**$\mu_M(z)$**	**$\mu_L(z)$**
$\mu_{LARGE}(80)$= 0 .6	**0**	**$\mu_M(z)$**	**$\mu_{VL}(z)$**

Appling Min Operation

	0	**$\mu_{MG}(80)$=0.4**	**$\mu_{LG}(80)$=0.6**
0	**0**	**0**	**0**
$\mu_{MP}(80)$=0.4	**0**	**0.4**	**0.4**
$\mu_{LP}(80)$=0.6	**0**	**0.4**	**0.6**

Applying Max Operation

The fuzzy set = {0.4, 0.4, 0.4, 0.6}. The max of the fuzzy set that is 0.6 which corresponds to $\mu_{VL}(z)$.

Defuzzification

$\mu_{VS}(z) = (z-60) / 20$

Substitute 0.6 in

$\mu_{VL}(z)$. z = 72

minutes

Thus 72 minutes is the wash time required for 80% of size of plate and 80% of grease

Case III: When plate size is medium and grease is medium

Example: Given Input 50% of size of plate and 40% of Grease

- $\mu_{SMALL}(50) = (50 - 50) / 50 = 0$
- $\mu_{MEDIUM}(50) = 50 / 50 = 1$
- $\mu_{SMALL}(40) = (50-40) / 50 = 0.2$
- $\mu_{MEDIUM}(40) = 40 / 50 = 0.8$

Applying fuzzified inputs on rulebase

	$\mu_{SG}(8)$=0.2	**$\mu_{MG}(8)$=0.8**	**0**
$\mu_{SP}(10)$=0.0	**$\mu_{VS}(z)$**	**$\mu_{M}(z)$**	**0**

$\mu_{MP}(10)$=1.0	**$\mu_{S}(z)$**	**$\mu_{M}(z)$**	**0**
0	**0**	**0**	**0**

Appling Min Operation

	$\mu_{SMALL}(8)$= 0	**$\mu_{MEDIUM}(8)$= 1**	**0**
$\mu_{SMALL}(10)$= 0 .2	**0.0**	**0.0**	**0**
$\mu_{MEDIUM}(10)$= 0.8	**0.2**	**0.8**	**0**
0	**0**	**0**	**0**

Applying Max Operation

The fuzzy set = {0.0, 0.0, 0.2, 0.8}. The max of the fuzzy set that is 0.8 which is of $\mu_M(z)$.

Defuzzification

$\mu_M(z) = (z-20) / 20 \quad 20 \leq z \leq 40$

$(60-z)/20$

$40 \leq z \leq 60$

Substitute 0.8 in $\mu_M(z)$.

z1 = 36 and z2 = 44, Average = (36 + 44) /2 = 40.

Thus 40 minutes is the wash time required for 50% of size of plate and 40% of grease

Implement the Fuzzy Inference System in Python:

Code:

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

room_temperature = ctrl.Antecedent(np.arange(16, 36, 1), 'room_temperature')
temperature = ctrl.Antecedent(np.arange(16, 36, 1), 'temperature')
humidity = ctrl.Antecedent(np.arange(10, 70, 1), 'humidity')
oxygen_level = ctrl.Antecedent(np.arange(15, 50, 1), 'oxygen_level')
fan_speed = ctrl.Consequent(np.arange(0, 1800, 1), 'fan_speed')
compressor_speed = ctrl.Consequent(np.arange(0, 1800, 1), 'compressor_speed')

room_temperature['very-cold'] = fuzz.trimf(temperature.universe, [0, 0, 20])
room_temperature['cold'] = fuzz.trimf(temperature.universe, [16, 20, 24])
room_temperature['warm'] = fuzz.trimf(temperature.universe, [20, 24, 28])
room_temperature['hot'] = fuzz.trimf(temperature.universe, [24, 28, 32])
room_temperature['very-hot'] = fuzz.trimf(temperature.universe, [28, 32, 36])
# print("room_temperature", room_temperature.universe)

temperature['very-cold'] = fuzz.trimf(temperature.universe, [0, 0, 20])
temperature['cold'] = fuzz.trimf(temperature.universe, [16, 20, 24])
temperature['warm'] = fuzz.trimf(temperature.universe, [20, 24, 28])
temperature['hot'] = fuzz.trimf(temperature.universe, [24, 28, 32])
temperature['very-hot'] = fuzz.trimf(temperature.universe, [28, 32, 36])
# print("temperature", temperature.universe)

humidity['dry'] = fuzz.trimf(humidity.universe, [0, 0, 40])
humidity['refreshing'] = fuzz.trimf(humidity.universe, [30, 40, 50])
humidity['comfortable'] = fuzz.trimf(humidity.universe, [40, 50, 60])
humidity['humid'] = fuzz.trimf(humidity.universe, [50, 60, 70])
# print("humidity", humidity.universe)

oxygen_level['low'] = fuzz.trimf(oxygen_level.universe, [0, 0, 20])
oxygen_level['medium'] = fuzz.trimf(oxygen_level.universe, [10, 20, 30])
```

```
oxygen_level['high'] = fuzz.trimf(oxygen_level.universe, [30, 40, 50])
# print("oxygen_level", oxygen_level.universe)

fan_speed['minimum'] = fuzz.trimf(fan_speed.universe, [0, 0, 600])
fan_speed['slow'] = fuzz.trimf(fan_speed.universe, [300, 600, 900])
fan_speed['medium'] = fuzz.trimf(fan_speed.universe, [600, 900, 1200])
fan_speed['fast'] = fuzz.trimf(fan_speed.universe, [900, 1200, 1500])
fan_speed['maximum'] = fuzz.trimf(fan_speed.universe, [1200, 1500, 1800])
# print("fan_speed", fan_speed.universe)

compressor_speed['minimum'] = fuzz.trimf(fan_speed.universe, [0, 0, 600])
compressor_speed['slow'] = fuzz.trimf(fan_speed.universe, [300, 600, 900])
compressor_speed['medium'] = fuzz.trimf(fan_speed.universe, [600, 900, 1200])
compressor_speed['fast'] = fuzz.trimf(fan_speed.universe, [900, 1200, 1500])
compressor_speed['maximum'] = fuzz.trimf(fan_speed.universe, [1200, 1500, 1800])


# print("compressor_speed", compressor_speed.universe)

def get_fan_speed_control_rules():
    rule1a = ctrl.Rule(
        room_temperature['very-hot'] & temperature['very-cold'] & humidity['humid'],
        fan_speed['maximum']
    )
    rule1b = ctrl.Rule(
        room_temperature['very-hot'] & temperature['very-cold'] & humidity['comfortable'],
        fan_speed['fast']
    )
    rule1c = ctrl.Rule(
        room_temperature['very-hot'] & temperature['very-cold'] & humidity['refreshing'],
        fan_speed['medium']
    )
    rule1d = ctrl.Rule(
        room_temperature['very-hot'] | temperature['very-cold'] | humidity['dry'],
        fan_speed['slow']
    )

    rule2a = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['humid'],
        fan_speed['maximum']
    )
    rule2b = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['comfortable'],
        fan_speed['fast']
    )
    rule2c = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['refreshing'],
        fan_speed['medium']
    )
    rule2d = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['dry'],
        fan_speed['slow']
    )
```

```
rule3a = ctrl.Rule(
    room_temperature['hot'] | temperature['cold'] | humidity['humid'],
    fan_speed['maximum']
)
rule3b = ctrl.Rule(
    room_temperature['hot'] | temperature['cold'] | humidity['comfortable'],
    fan_speed['fast']
)
rule3c = ctrl.Rule(
    room_temperature['hot'] | temperature['cold'] | humidity['refreshing'],
    fan_speed['medium']
)
rule3d = ctrl.Rule(
    room_temperature['hot'] | temperature['cold'] | humidity['dry'],
    fan_speed['slow']
)

rule4a = ctrl.Rule(
    room_temperature['warm'] | temperature['cold'] | humidity['humid'],
    fan_speed['maximum']
)
rule4b = ctrl.Rule(
    room_temperature['warm'] | temperature['cold'] | humidity['comfortable'],
    fan_speed['fast']
)
rule4c = ctrl.Rule(
    room_temperature['warm'] | temperature['cold'] | humidity['refreshing'],
    fan_speed['medium']
)
rule4d = ctrl.Rule(
    room_temperature['warm'] | temperature['cold'] | humidity['dry'],
    fan_speed['slow']
)

rule5a = ctrl.Rule(
    room_temperature['cold'] | temperature['warm'] | humidity['humid'],
    fan_speed['maximum']
)
rule5b = ctrl.Rule(
    room_temperature['cold'] | temperature['warm'] | humidity['comfortable'],
    fan_speed['fast']
)
rule5c = ctrl.Rule(
    room_temperature['cold'] | temperature['warm'] | humidity['refreshing'],
    fan_speed['medium']
)
rule5d = ctrl.Rule(
    room_temperature['cold'] | temperature['warm'] | humidity['dry'],
    fan_speed['slow']
)

return [
    rule1a, rule1b, rule1c, rule1d,
    rule2a, rule2a, rule2b, rule2d,
```

```
        rule3a, rule3b, rule3c, rule3d,
        rule4a, rule4b, rule4c, rule4d,
        rule5a, rule5b, rule5c, rule5d
    ]


def get_compressor_speed_control_rules():
    rule1a = ctrl.Rule(
        room_temperature['very-hot'] & temperature['very-cold'] & humidity['humid'],
        compressor_speed['maximum']
    )
    rule1b = ctrl.Rule(
        room_temperature['very-hot'] & temperature['very-cold'] & humidity['comfortable'],
        compressor_speed['maximum']
    )
    rule1c = ctrl.Rule(
        room_temperature['very-hot'] & temperature['very-cold'] & humidity['refreshing'],
        compressor_speed['maximum']
    )
    rule1d = ctrl.Rule(
        room_temperature['very-hot'] | temperature['very-cold'] | humidity['dry'],
        compressor_speed['maximum']
    )

    rule2a = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['humid'],
        compressor_speed['fast']
    )
    rule2b = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['comfortable'],
        compressor_speed['fast']
    )
    rule2c = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['refreshing'],
        compressor_speed['fast']
    )
    rule2d = ctrl.Rule(
        room_temperature['very-hot'] | temperature['cold'] | humidity['dry'],
        compressor_speed['fast']
    )

    rule3a = ctrl.Rule(
        room_temperature['hot'] | temperature['cold'] | humidity['humid'],
        compressor_speed['fast']
    )
    rule3b = ctrl.Rule(
        room_temperature['hot'] | temperature['cold'] | humidity['comfortable'],
        compressor_speed['fast']
    )
    rule3c = ctrl.Rule(
        room_temperature['hot'] | temperature['cold'] | humidity['refreshing'],
        compressor_speed['fast']
    )
    rule3d = ctrl.Rule(
```

```
        room_temperature['hot'] | temperature['cold'] | humidity['dry'],
        compressor_speed['fast']
    )

    rule4a = ctrl.Rule(
        room_temperature['warm'] | temperature['cold'] | humidity['humid'],
        compressor_speed['medium']
    )
    rule4b = ctrl.Rule(
        room_temperature['warm'] | temperature['cold'] | humidity['comfortable'],
        compressor_speed['medium']
    )
    rule4c = ctrl.Rule(
        room_temperature['warm'] | temperature['cold'] | humidity['refreshing'],
        compressor_speed['medium']
    )
    rule4d = ctrl.Rule(
        room_temperature['warm'] | temperature['cold'] | humidity['dry'],
        compressor_speed['medium']
    )

    rule5a = ctrl.Rule(
        room_temperature['cold'] | temperature['warm'] | humidity['humid'],
        compressor_speed['medium']
    )
    rule5b = ctrl.Rule(
        room_temperature['cold'] | temperature['warm'] | humidity['comfortable'],
        compressor_speed['medium']
    )
    rule5c = ctrl.Rule(
        room_temperature['cold'] | temperature['warm'] | humidity['refreshing'],
        compressor_speed['medium']
    )
    rule5d = ctrl.Rule(
        room_temperature['cold'] | temperature['warm'] | humidity['dry'],
        compressor_speed['medium']
    )

    return [
        rule1a, rule1b, rule1c, rule1d,
        rule2a, rule2a, rule2b, rule2d,
        rule3a, rule3b, rule3c, rule3d,
        rule4a, rule4b, rule4c, rule4d,
        rule5a, rule5b, rule5c, rule5d
    ]


ac_ctrl = ctrl.ControlSystem(
    get_fan_speed_control_rules() + get_compressor_speed_control_rules()
)

room_temperature.view()
in_rt = input("Enter room temperature:")
```

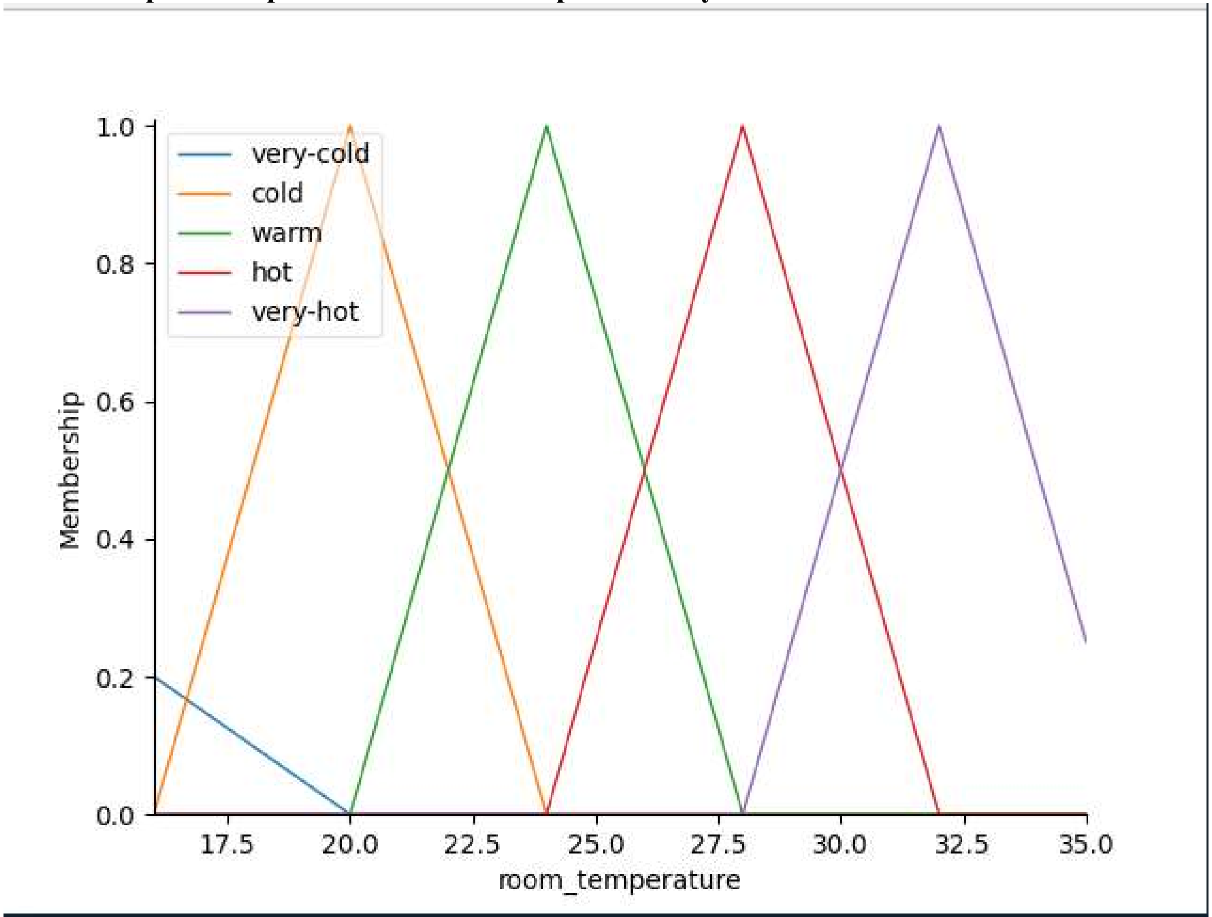
```
temperature.view()
in_tt = input("Enter target temperature:")

humidity.view()
in_hd = input("Enter humidity:")

# oxygen_level.view()
# in_ol = input("Enter oxygen level:")

fan_speed.view()
compressor_speed.view()

input('Press Enter For Processing Rules')
speed = ctrl.ControlSystemSimulation(ac_ctrl)
speed.input['room_temperature'] = int(in_rt)
speed.input['temperature'] = int(in_tt)
speed.input['humidity'] = int(in_hd)
speed.compute()

input('Press Enter to View Fan Speed')
print("Fan Speed", f"{speed.output['fan_speed']} RPM")
fan_speed.view(sim=speed)

input('Press Enter to View Compressor Speed')
print("Compressor Speed", f"{speed.output['compressor_speed']} RPM")
compressor_speed.view(sim=speed)

input('Press any key to exit!')
```

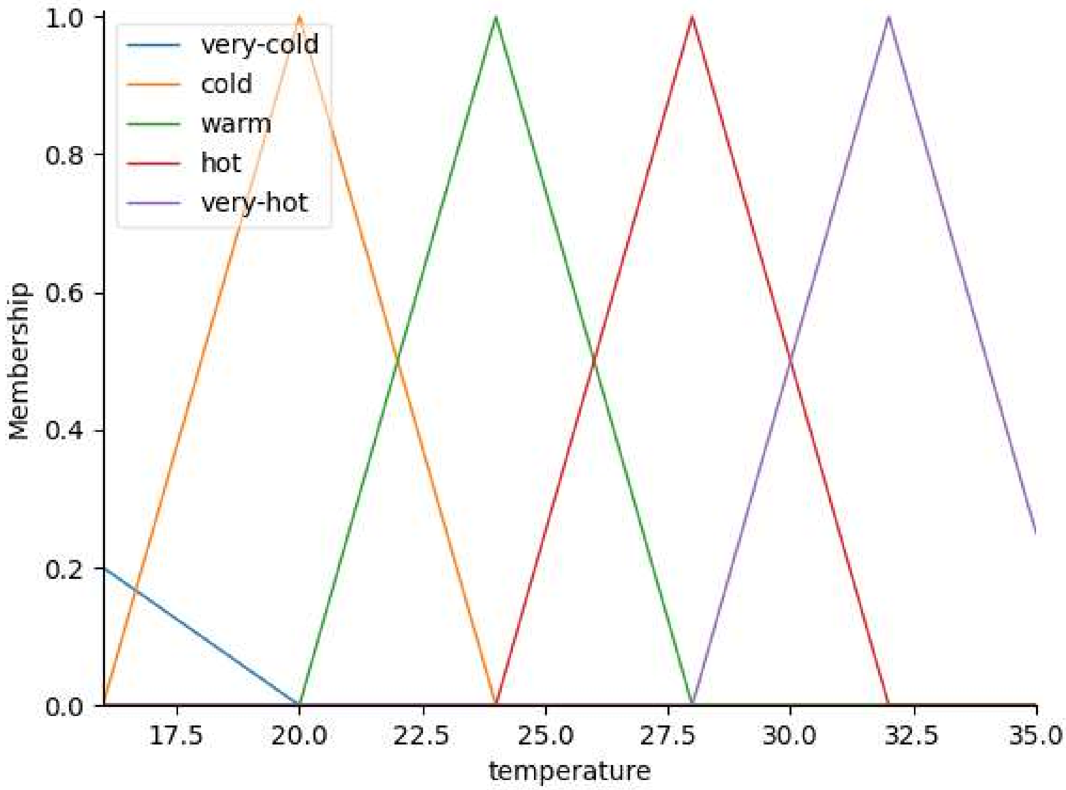
Output:

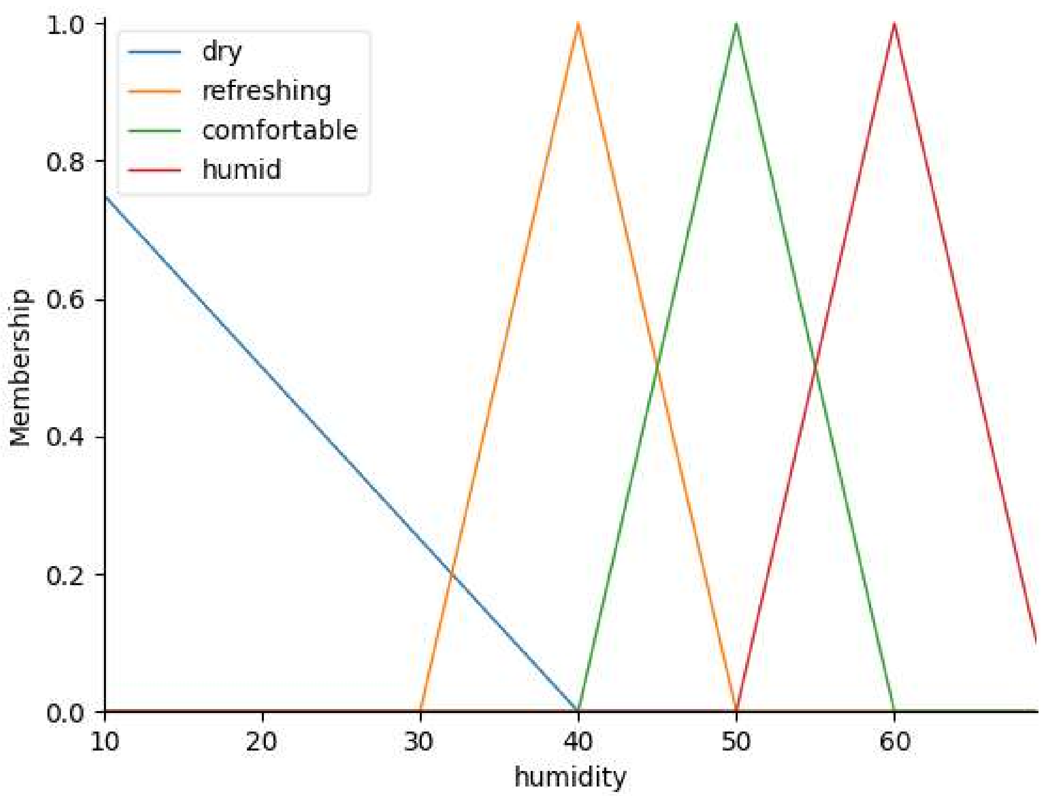
```
= RESTART: C:\Users\Administrator.MAHESHC\AppData\Local\Programs\Python\Python39\a
c.py
Enter room temperature:32
Enter target temperature:19
Enter humidity:70
Press Enter For Processing Rules
Press Enter to View Fan Speed
Fan Speed 1052.9398755669563 RPM
Press Enter to View Compressor Speed
Compressor Speed 1207.6909592196582 RPM
Press any key to exit!
```

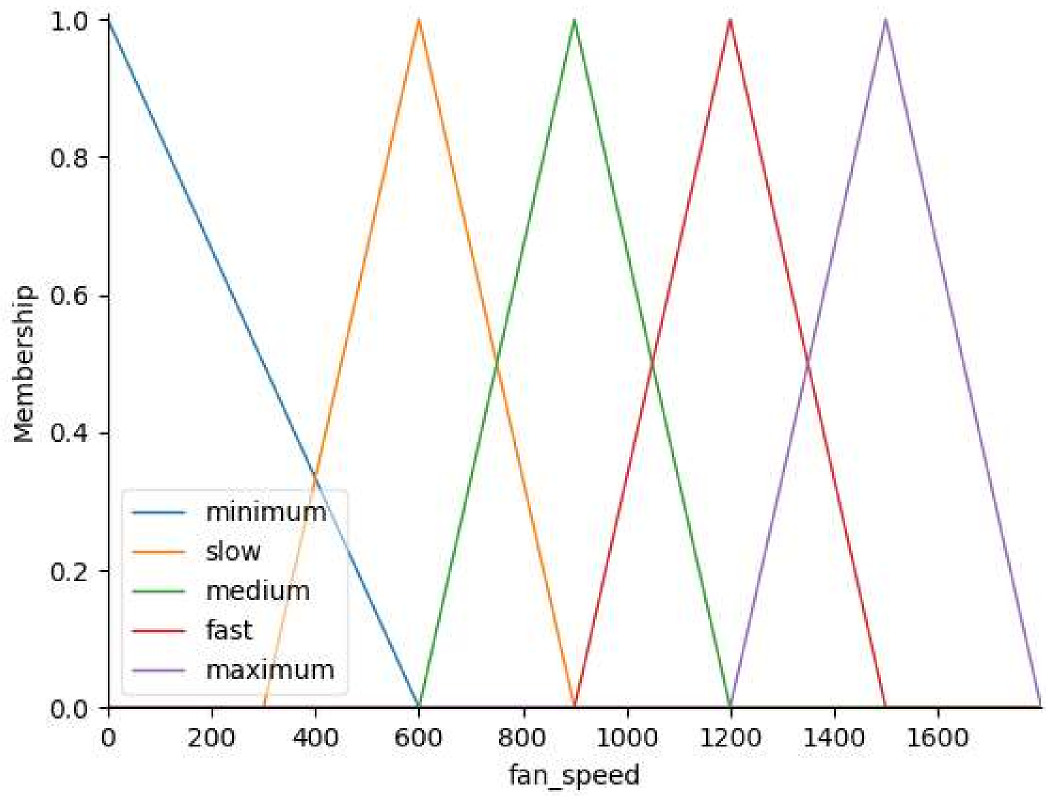
Fig: Output according to the given conditions in fuzzy operations

Graphical Representation of the output of Fuzzy controller:

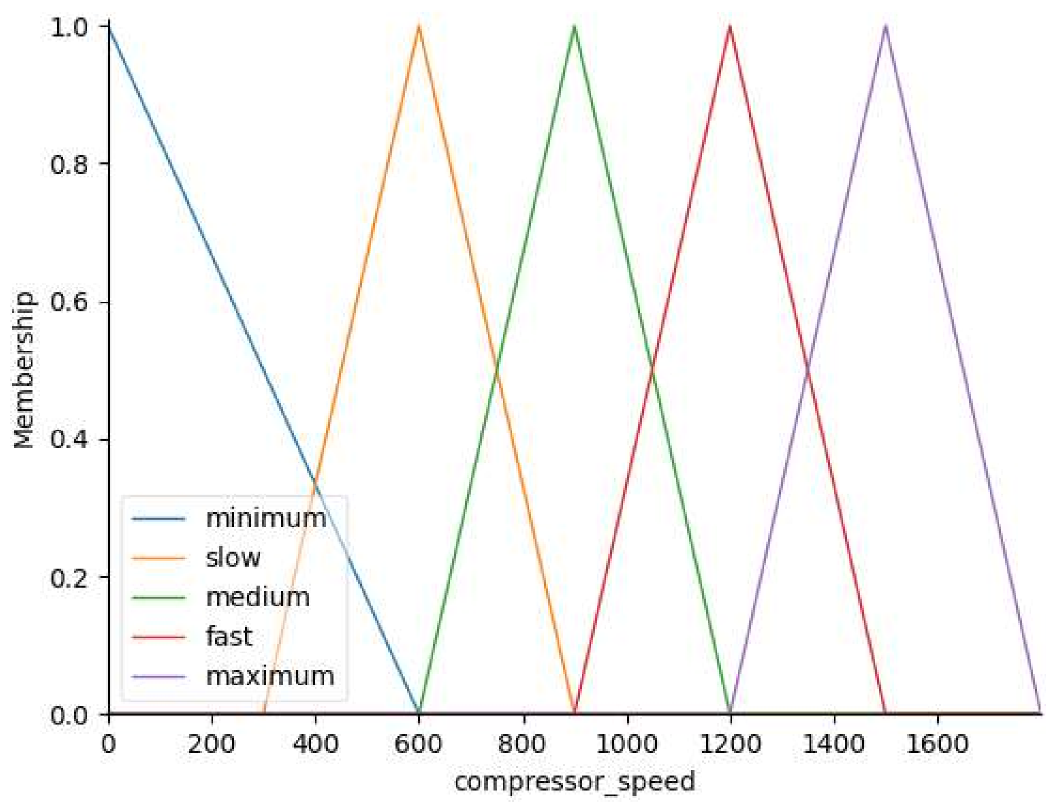



very-cold
cold
warm
hot
very-hot
Membership
temperature
0.0
0.2
0.4
0.6
0.8
1.0
17.5
20.0
22.5
25.0
27.5
30.0
32.5
35.0

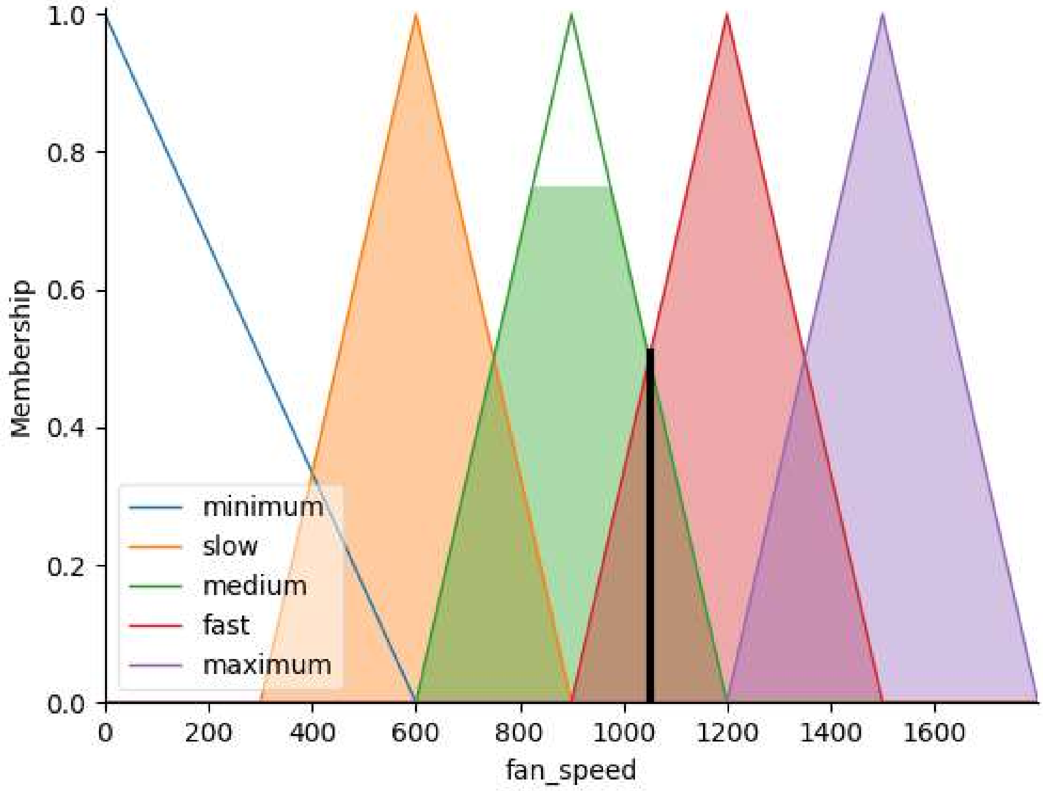

dry
refreshing
comfortable
humid
Membership
humidity
0.0
0.2
0.4
0.6
0.8
1.0
10
20
30
40
50
60

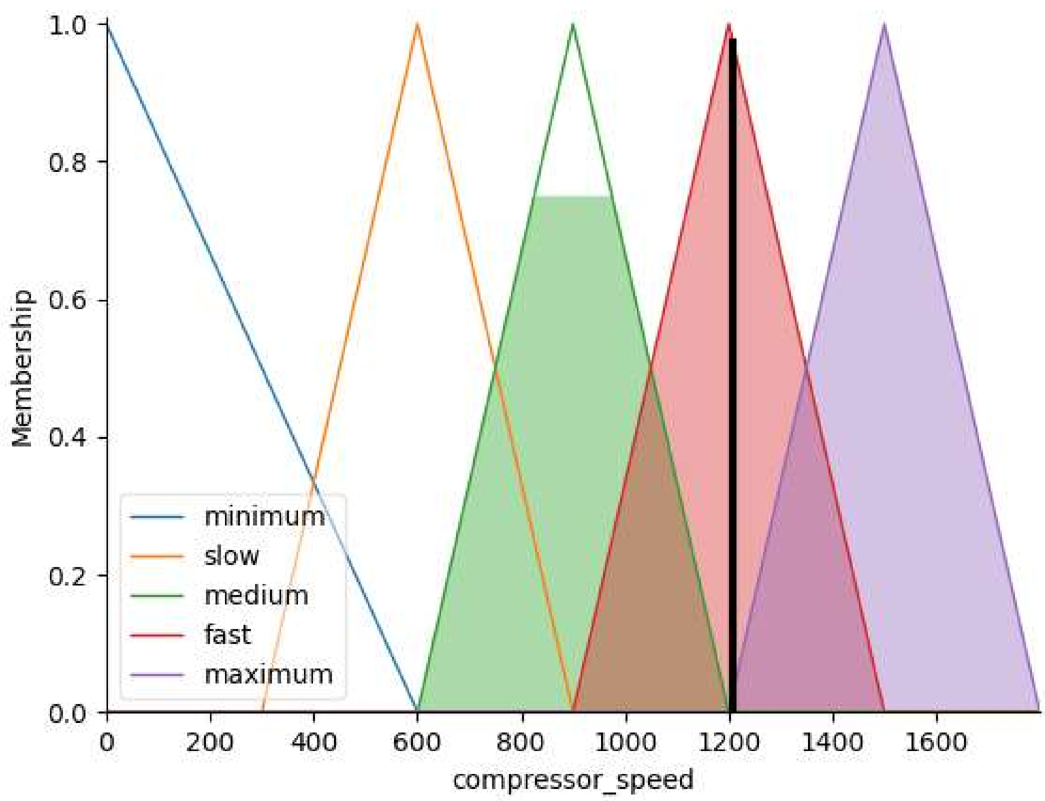


Membership
fan_speed
minimum
slow
medium
fast
maximum
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
1200
1400
1600



Membership
compressor_speed
minimum
slow
medium
fast
maximum
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
1200
1400
1600





Conclusion: In this experiment, we implemented a fuzzy inference system for Calculation of Fan and Compressor speed of AC . The inputs were grease and size of plate and output was given by the washtime. The membership function was calculated and rulebase was written for the different cases. Three cases were studied: -low grease and small plate size, medium grease and medium plate size, high grease and large plate size. The output for each was obtained successfully.

A fuzzy inference system is used in different fields, for example, information order, choice examination, master system, time arrangement forecasts, advanced mechanics, and example acknowledgment. It is otherwise called a fuzzy rule-based system, fuzzy model, fuzzy logic controller, fuzzy expert system, and fuzzy associative memory.

A code was written in python and the output was obtained successfully.