Experiment 8 - Implementation of Fuzzy Control System using Fuzzy Tool

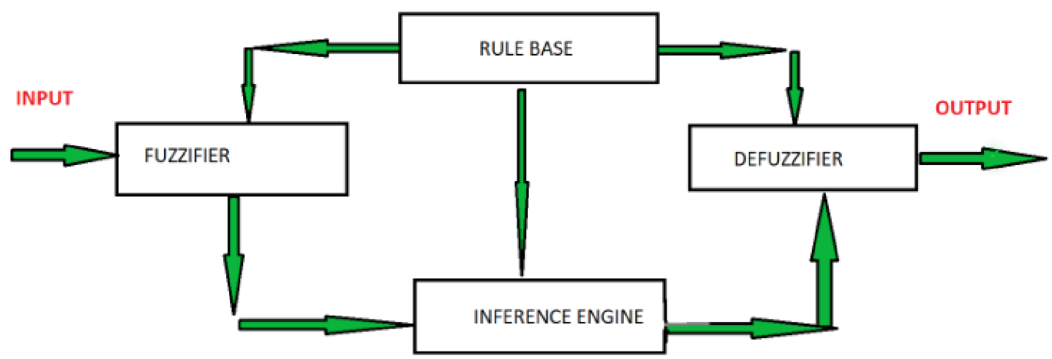
Roll No.	37
Name	Mikil Lalwani
Class	D20 B
Subject	Data Science Lab
LO Mapped	L3: Implement a fuzzy controller system

Aim: To Design a Fuzzy Control System using Fuzzy Tool.

Fuzzy Control System:

A fuzzy control system, often referred to as a fuzzy logic controller (FLC), is a type of control system that uses fuzzy logic to model and control complex and uncertain systems. Unlike traditional control systems that rely on precise mathematical models and crisp (binary) logic, fuzzy control systems are designed to handle imprecise, uncertain, and vague information.

Fuzzy Control System Architecture:



FUZZY LOGIC ARCHITECTURE

Components and concepts of a fuzzy control system:

1. Membership Functions: Fuzzy control systems use membership functions to represent the degree of membership of a variable in a fuzzy set. These membership functions describe how different input variables relate to linguistic terms (e.g., "low," "medium," "high") and output variables in the system.
2. Fuzzy Rules: Fuzzy control systems use a set of fuzzy rules to relate the input variables to the output variables. These rules are typically expressed in

the form of "if-then" statements that specify how to adjust the system's output based on the input variables and the fuzzy logic.

3. Fuzzy Inference Engine: The fuzzy inference engine processes the input data based on the fuzzy rules and membership functions to determine the appropriate output. It uses fuzzy logic to make decisions and calculate the degree to which each rule is applicable.

4. Defuzzification: After the fuzzy inference engine produces a fuzzy output, the result needs to be converted into a crisp (real) value. This process is called defuzzification and typically involves calculating a weighted average of the fuzzy output values.

5. Feedback Loop: In many control systems, there is a feedback loop that continuously updates the input and adjusts the system's output based on the actual performance and desired setpoints. Fuzzy control systems can incorporate feedback to improve their control decisions.

Fuzzy Tool Used:

skfuzzy:

The skfuzzy library is a Python library that provides tools for working with fuzzy logic and fuzzy systems. Fuzzy logic is a mathematical framework used to handle uncertainty and imprecision in data and decision-making. skfuzzy allows you to create and simulate fuzzy systems, which are particularly useful in situations where traditional, crisp (binary) logic and control systems are insufficient due to the complexity or uncertainty of the problem.

Code and Observation:

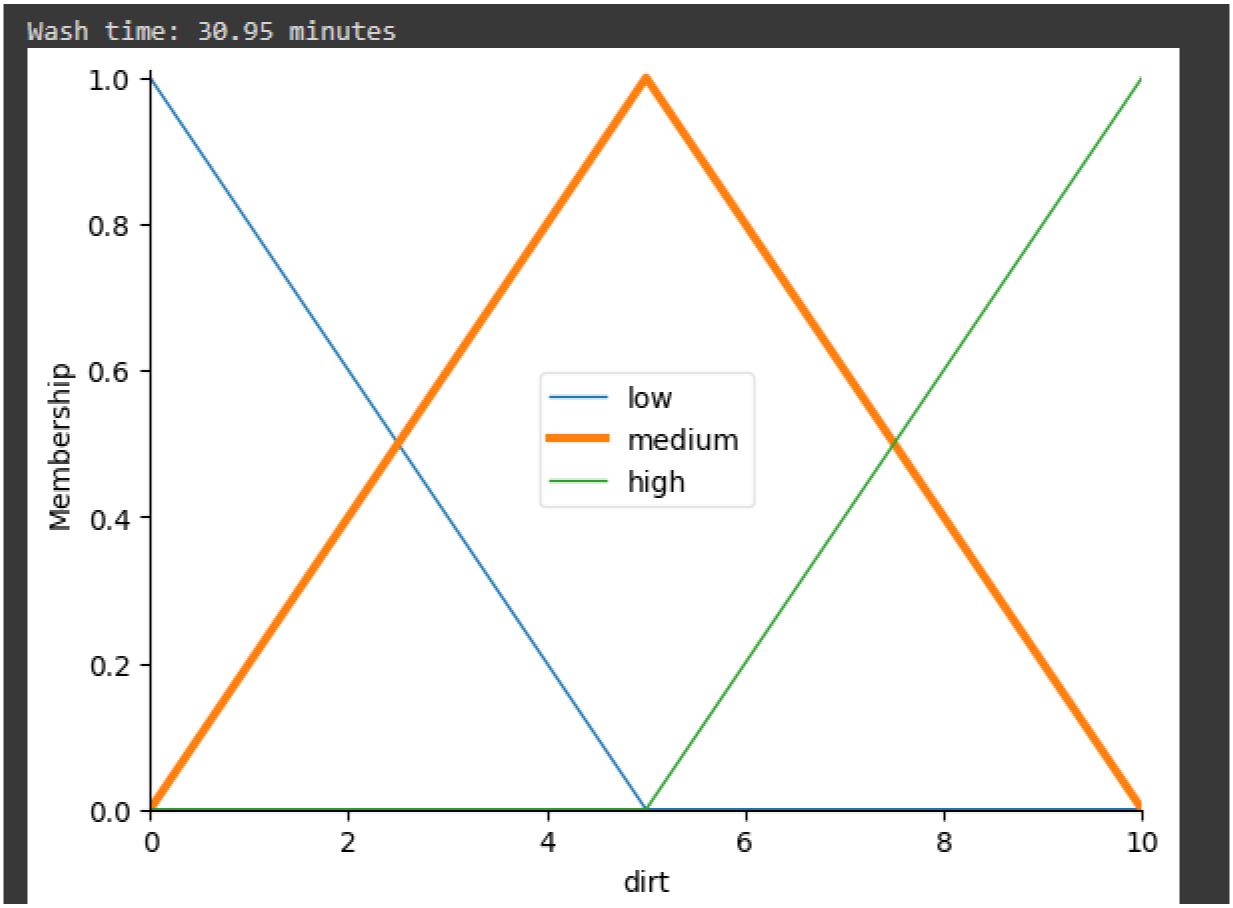
Code:

```
# Install scikit-fuzzy if not already installed
!pip install -q scikit-fuzzy

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

# Define input variables
dirt = ctrl.Antecedent(np.arange(0, 11, 1), 'dirt')
grease = ctrl.Antecedent(np.arange(0, 11, 1), 'grease')

# Define output variable
wash_time = ctrl.Consequent(np.arange(0, 61, 1), 'wash_time')

# Define membership functions for input variables
dirt['low'] = fuzz.trimf(dirt.universe, [0, 0, 5])
dirt['medium'] = fuzz.trimf(dirt.universe, [0, 5, 10])
dirt['high'] = fuzz.trimf(dirt.universe, [5, 10, 10])

grease['low'] = fuzz.trimf(grease.universe, [0, 0, 5])
grease['medium'] = fuzz.trimf(grease.universe, [0, 5, 10])
grease['high'] = fuzz.trimf(grease.universe, [5, 10, 10])

# Define membership functions for output variable
wash_time['very_short'] = fuzz.trimf(wash_time.universe, [0, 0, 15])
wash_time['short'] = fuzz.trimf(wash_time.universe, [0, 15, 30])
wash_time['medium'] = fuzz.trimf(wash_time.universe, [15, 30, 45])
wash_time['long'] = fuzz.trimf(wash_time.universe, [30, 45, 60])
wash_time['very_long'] = fuzz.trimf(wash_time.universe, [45, 60, 60])
```

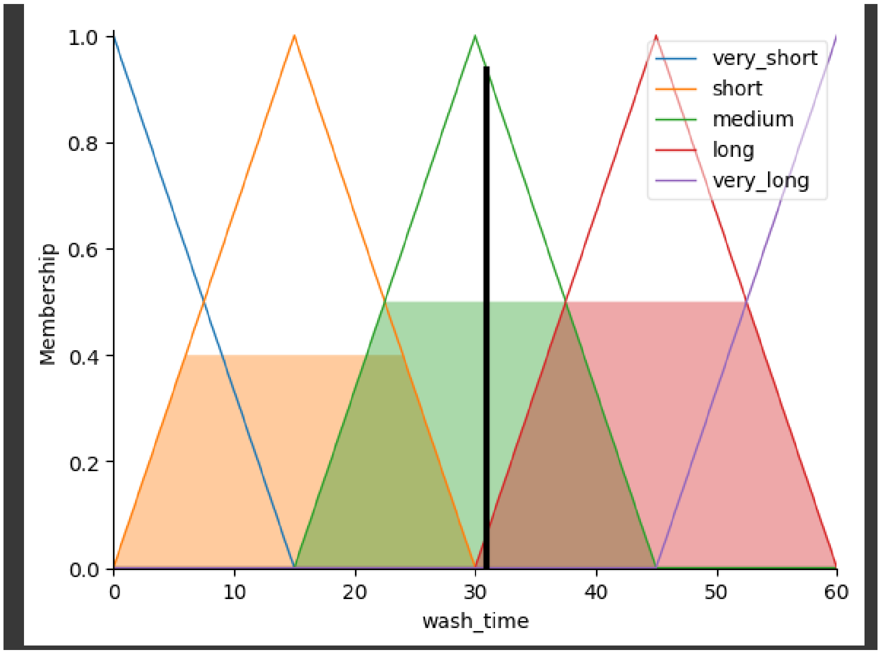
```
dirt['medium'].view()
grease['low'].view()
wash_time['long'].view()

# Define fuzzy rules
rule1 = ctrl.Rule(dirt['low'] & grease['low'], wash_time['very_short'])
rule2 = ctrl.Rule(dirt['low'] & grease['medium'], wash_time['short'])
rule3 = ctrl.Rule(dirt['low'] & grease['high'], wash_time['short'])
rule4 = ctrl.Rule(dirt['medium'] & grease['low'], wash_time['short'])
rule5 = ctrl.Rule(dirt['medium'] & grease['medium'], wash_time['medium'])
rule6 = ctrl.Rule(dirt['medium'] & grease['high'], wash_time['long'])
rule7 = ctrl.Rule(dirt['high'] & grease['low'], wash_time['medium'])
rule8 = ctrl.Rule(dirt['high'] & grease['medium'], wash_time['long'])
rule9 = ctrl.Rule(dirt['high'] & grease['high'], wash_time['very_long'])

# Create control system
wash_time_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7,
rule8, rule9])

# Create simulation
wash_time_simulation = ctrl.ControlSystemSimulation(wash_time_ctrl)

# Set input values
wash_time_simulation.input['dirt'] = 7.5  # Dirt level (0-10)
wash_time_simulation.input['grease'] = 3.0  # Grease level (0-10)

# Perform the fuzzy inference
wash_time_simulation.compute()

# Get the output value
wash_time_output = wash_time_simulation.output['wash_time']
wash_time.view(sim=wash_time_simulation)
print(f"Wash time: {wash_time_output:.2f} minutes")
```

Output

1. Membership function for Dirt:



2. Outline of Final washtime taken:



Conclusion:

In conclusion, the experiment using a Python-based fuzzy control system showcased the system's ability to handle complex, uncertain scenarios. Fuzzy logic's adaptability, human-like reasoning, and practical applications were evident. The experiment highlighted the need for fine-tuning and optimization. Overall, fuzzy control systems offer promise in addressing real-world challenges where traditional methods fall short.