Name : Keith Fernandes Roll No. : 25

St. Francis Institute of Technology, Mumbai-400 103
Department Of Information Technology

A.Y. 2025-2026
Class: BE-ITA/B, Semester: VII
Subject: Data Science Lab

Experiment – 2

1. **Aim:** To implement Fuzzy set operation and Fuzzy membership functions and design a Fuzzy Controller

2. **Objectives:** Students should be able to understand basics of Fuzzy Logic

3. **Prerequisite:** Python basics

4. **Requirements:** PC, Python 3.9, Windows 10/ MacOS/ Linux, IDLE IDE

5. **Pre-Experiment Exercise:**
   Theory:
   Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO.

   The conventional logic block that a computer can understand takes precise input and produces a definite output as TRUE or FALSE, which is equivalent to human's YES or NO.

   The inventor of fuzzy logic, Lotfi Zadeh, observed that unlike computers, the human decision making includes a range of possibilities between YES and NO, such as −

   CERTAINLY YES
   POSSIBLY YES
   CANNOT SAY
   POSSIBLY NO
   CERTAINLY NO
   The fuzzy logic works on the levels of possibilities of input to achieve the definite output.

   ## Fuzzy Operations:

   Union :

   Consider 2 Fuzzy Sets denoted by A and B, then let's consider Y be the Union of them, then for every member of A and B, Y will be:

   degree_of_membership(Y)= max(degree_of_membership(A), degree_of_membership(B))
   Intersection :

   Consider 2 Fuzzy Sets denoted by A and B, then let's consider Y be the Intersection of them, then for every member of A and B, Y will be:

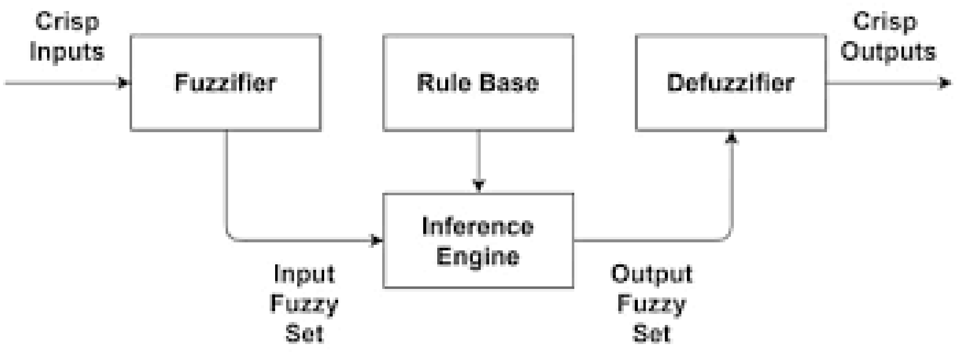
   degree_of_membership(Y)= min(degree_of_membership(A), degree_of_membership(B))

   Complement :

Consider a Fuzzy Sets denoted by A , then let’s consider Y be the Complement of it, then for every member of A , Y will be: degree_of_membership(Y)= 1 - degree_of_membership(A)

Difference :
Consider 2 Fuzzy Sets denoted by A and B, then let’s consider Y be the Intersection of them, then for every member of A and B, Y will be:
degree_of_membership(Y)= min(degree_of_membership(A), 1- degree_of_membership(B))

Fuzzy Controller:



6. **Laboratory Exercise**
   A. Procedure
   i. Use google colab for programming.
   ii. Implement following Fuzzy set operations and represent output in graphical format
      - Union
      - Intersection
      - Complement

   b. Implement Fuzzy membership functions
      - Triangular Function
      - Trapezoidal Function
      - Gaussian Function

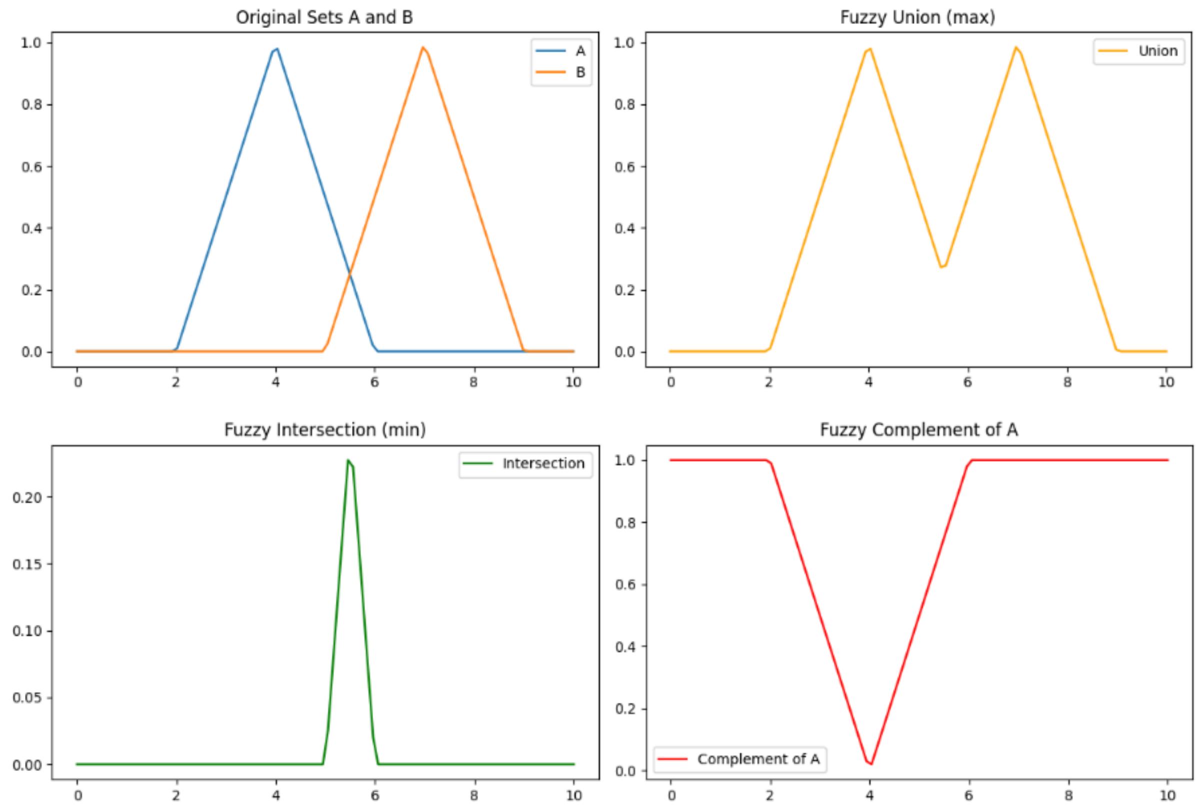
   c. Design a Fuzzy Controller

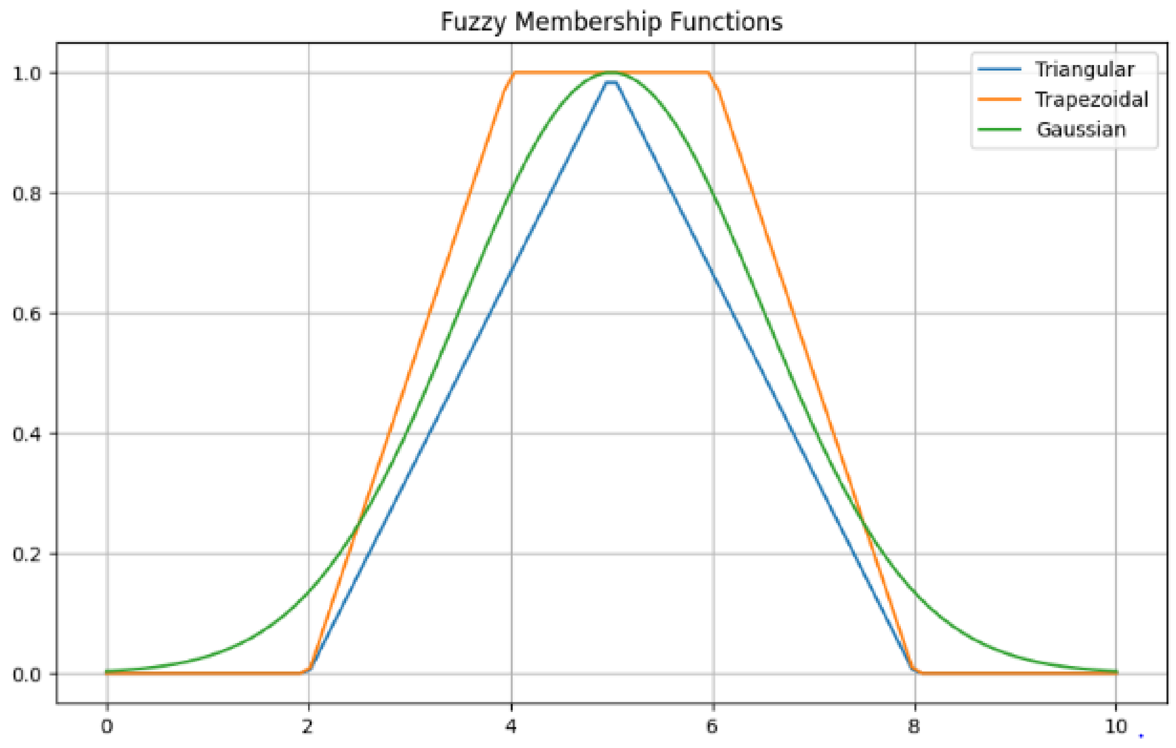
7. **Post-Experiments Exercise:**
   A. Extended Theory:
      a. Explain properties of Fuzzy sets.
      b. Write a brief note on Fuzzy Relation.
      c. Solve a problem on Fuzzy composition

   B. Conclusion:
   1. Write what was performed in the program (s) .
   2. What is the significance of program and what Objective is achieved?

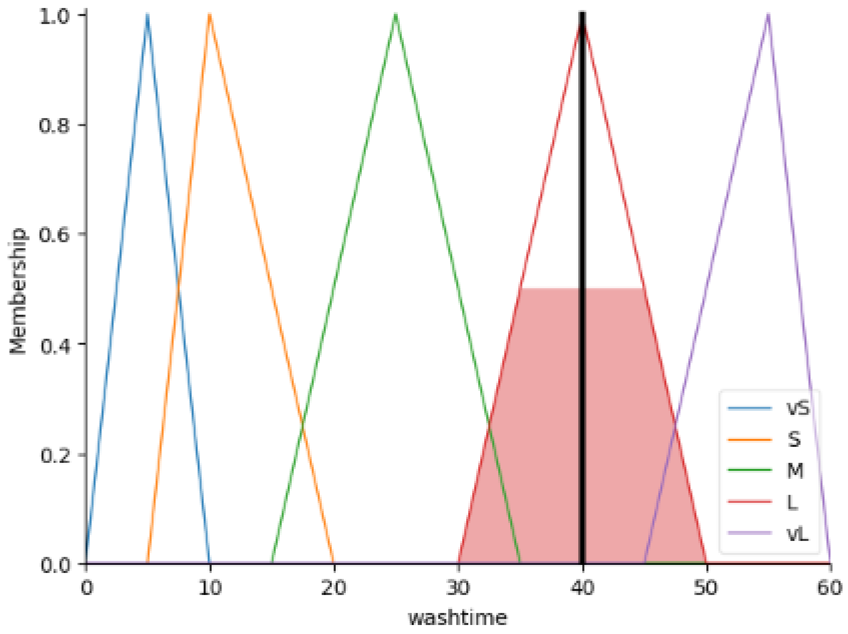
   C. References:
[1] https://drive.google.com/file/d/1MVwbyXqlMSeb-25CSQ9WAAxais9LLqhO/view?usp=sharing
[2] S.N. Sivanandam, S.N. Deepa, “Principles of Soft Computing”, Wiley Publication.

```
import numpy as np
import skfuzzy as fuzz
import matplotlib.pyplot as plt

# --------------------------------------------------
# 1. FUZZY SET OPERATIONS: Union, Intersection, Complement
# --------------------------------------------------
x = np.linspace(0, 10, 100)
A = fuzz.trimf(x, [2, 4, 6])
B = fuzz.trimf(x, [5, 7, 9])

union = np.fmax(A, B)
intersection = np.fmin(A, B)
complement_A = 1 - A

plt.figure(figsize=(12, 8))

plt.subplot(2, 2, 1)
plt.plot(x, A, label='A')
plt.plot(x, B, label='B')
plt.title('Original Sets A and B')
plt.legend()

plt.subplot(2, 2, 2)
plt.plot(x, union, label='Union', color='orange')
plt.title('Fuzzy Union (max)')
plt.legend()

plt.subplot(2, 2, 3)
plt.plot(x, intersection, label='Intersection', color='green')
plt.title('Fuzzy Intersection (min)')
plt.legend()

plt.subplot(2, 2, 4)
plt.plot(x, complement_A, label='Complement of A', color='red')
plt.title('Fuzzy Complement of A')
plt.legend()

plt.tight_layout()
plt.show()

# --------------------------------------------------
# 2. MEMBERSHIP FUNCTIONS: Triangular, Trapezoidal, Gaussian
# --------------------------------------------------
x = np.linspace(0, 10, 100)

tri_mf = fuzz.trimf(x, [2, 5, 8])
trap_mf = fuzz.trapmf(x, [2, 4, 6, 8])
gauss_mf = fuzz.gaussmf(x, 5, 1.5)

plt.figure(figsize=(10, 6))
plt.plot(x, tri_mf, label='Triangular')
plt.plot(x, trap_mf, label='Trapezoidal')
plt.plot(x, gauss_mf, label='Gaussian')
plt.title('Fuzzy Membership Functions')
plt.legend()
plt.grid(True)
plt.show()

# --------------------------------------------------
# 3. FUZZY CONTROLLER: Washing Machine
# --------------------------------------------------
from skfuzzy import control as ctrl

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

# Output variable
washtime = ctrl.Consequent(np.arange(0, 61, 1), 'washtime')

# Membership functions for dirt
dirt['small'] = fuzz.trimf(dirt.universe, [0, 0, 4])
dirt['medium'] = fuzz.trimf(dirt.universe, [2, 5, 8])
dirt['large'] = fuzz.trimf(dirt.universe, [6, 10, 10])

# Membership functions for grease
grease['small'] = fuzz.trimf(grease.universe, [0, 0, 4])
grease['medium'] = fuzz.trimf(grease.universe, [2, 5, 8])
grease['large'] = fuzz.trimf(grease.universe, [6, 10, 10])

# Membership functions for wash time
washtime['vS'] = fuzz.trimf(washtime.universe, [0, 5, 10])
washtime['S'] = fuzz.trimf(washtime.universe, [5, 10, 20])
washtime['M'] = fuzz.trimf(washtime.universe, [15, 25, 35])
washtime['L'] = fuzz.trimf(washtime.universe, [30, 40, 50])
washtime['vL'] = fuzz.trimf(washtime.universe, [45, 55, 60])
# Define all fuzzy rules in a list
rules = [
    ctrl.Rule(dirt['small'] & grease['small'], washtime['vS']),
    ctrl.Rule(dirt['small'] & grease['medium'], washtime['S']),
    ctrl.Rule(dirt['small'] & grease['large'], washtime['M']),

    ctrl.Rule(dirt['medium'] & grease['small'], washtime['S']),
    ctrl.Rule(dirt['medium'] & grease['medium'], washtime['M']),
    ctrl.Rule(dirt['medium'] & grease['large'], washtime['L']),

    ctrl.Rule(dirt['large'] & grease['small'], washtime['M']),
    ctrl.Rule(dirt['large'] & grease['medium'], washtime['L']),
    ctrl.Rule(dirt['large'] & grease['large'], washtime['vL']),
]

# Now define the control system
wash_ctrl = ctrl.ControlSystem(rules)
wash_sim = ctrl.ControlSystemSimulation(wash_ctrl)

# Provide inputs and compute output
wash_sim.input['dirt'] = 6
wash_sim.input['grease'] = 8
wash_sim.compute()

# Print and plot the result
print("Calculated Wash Time: {:.2f} minutes".format(wash_sim.output['washtime']))
washtime.view(sim=wash_sim
```





```
Calculated Wash Time: 40.00 minutes
```


Membership
washtime
vS
S
M
L
vL