KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Engineering (CSE)

CSE3210: Artificial Intelligence

Laboratory Assignment Report

Submitted To:

Md Mehrab Hossain Opi

Lecturer

Department of Department of Computer Science and Engineering (CSE)

Khulna University of Engineering & Technology (KUET)

Waliul Islam Sumon

Lecturer

Department of Department of Computer Science and Engineering (CSE)

Khulna University of Engineering & Technology (KUET)

Submitted By:

Arka Braja Prasad Nath

Roll No: 2107055
Year: Third

Semester: Second

Department of Computer Science and Engineering (CSE)

Khulna University of Engineering & Technology (KUET)

Date of Submission: February 15, 2026

Problem Statement and Goal

The objective of this assignment is to design a **Fuzzy Washing Machine Controller** that determines the optimal **Wash Time** based on:

- **Dirtiness (0–100)** = Low, Medium, High
- **Load Size (0–100)** = Small, Medium, Large

Output:

- **Wash Time (0–60 minutes)** = Short, Medium, Long

Why Fuzzy?

In real-life washing systems, dirtiness and load size cannot be precisely categorized using rigid thresholds. A load with dirtiness value 49 is almost similar to 50, but crisp logic treats them differently.

Fuzzy logic allows smooth transitions between categories and models human reasoning such as:

```
"If clothes are very dirty and the drum is full, wash longer."
```

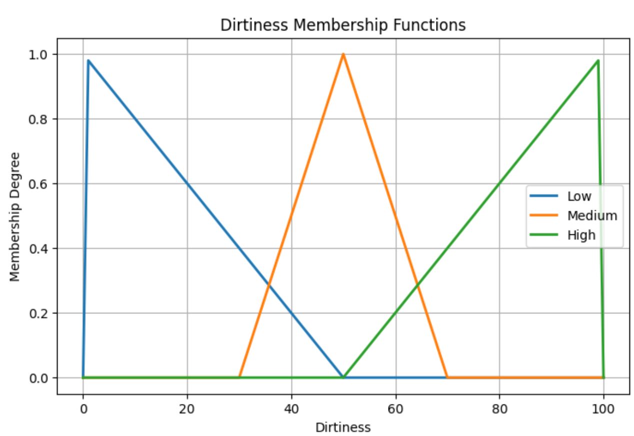
Membership Functions

Triangular membership functions were used for simplicity and smooth overlap.

(A) Dirtiness (0–100)

- Low (0, 0, 50)
- Medium (30, 50, 70)
- High (50, 100, 100)

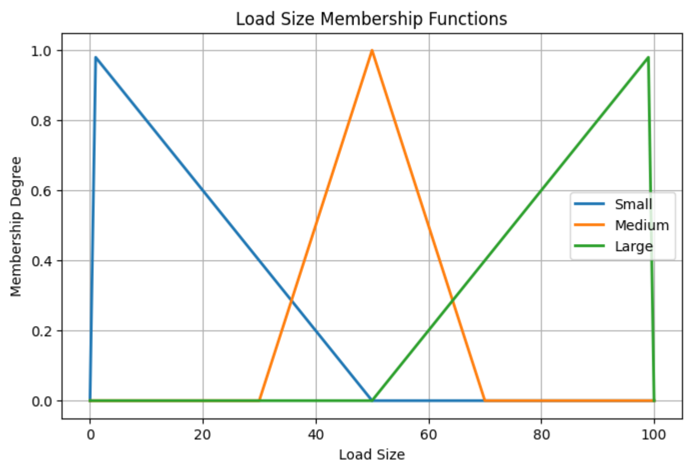
These overlaps ensure smooth transition between low–medium and medium–high dirtiness.



(B) Load Size (0–100)

- Small (0, 0, 50)
- Medium (30, 50, 70)
- Large (50, 100, 100)

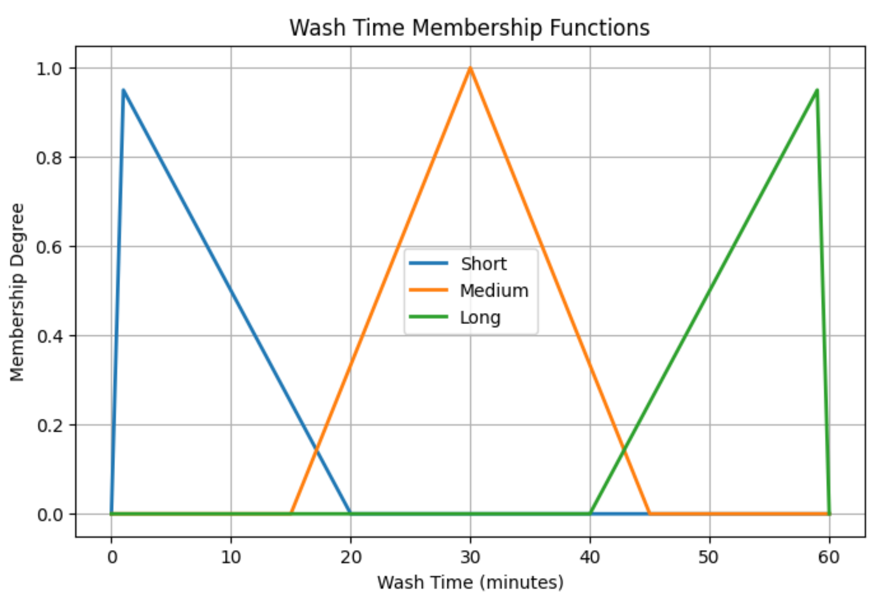
Similar overlap ensures realistic transition between load sizes.



(C) Wash Time (0–60 minutes)

- Short (0, 0, 20)
- Medium (15, 30, 45)
- Long (40, 60, 60)

Medium overlaps both short and long to allow smooth blending of rules.



Rule Base

- IF Dirtiness IS Low AND Load Size IS Small = Wash Time IS Short
- IF Dirtiness IS Low AND Load Size IS Medium = Wash Time IS Short
- IF Dirtiness IS Low AND Load Size IS Large = Wash Time IS Medium
- IF Dirtiness IS Medium AND Load Size IS Small = Wash Time IS Short
- IF Dirtiness IS Medium AND Load Size IS Medium = Wash Time IS Medium
- IF Dirtiness IS Medium AND Load Size IS Large = Wash Time IS Long
- IF Dirtiness IS High AND Load Size IS Small = Wash Time IS Medium
- IF Dirtiness IS High AND Load Size IS Medium = Wash Time IS Long
- IF Dirtiness IS High AND Load Size IS Large = Wash Time IS Long

This rule base ensures complete coverage of all possible input combinations.

Implementation Sketch

```
START
DEFINE range Dirtiness = 0 to 100
DEFINE range Load_Size = 0 to 100
DEFINE range Wash_Time = 0 to 60
FUNCTION TRIANGULAR(x, a, b, c)
   IF x <= a OR x >= c THEN
      RETURN 0
   ELSE IF x > a AND x <= b THEN
      RETURN (x - a) / (b - a)
   ELSE
      RETURN (c - x) / (c - b)
END FUNCTION
INPUT dirtiness_value
INPUT load_size_value
// Fuzzification
d_low    = TRIANGULAR(dirtiness_value, 0, 0, 50)
d_medium = TRIANGULAR(dirtiness_value, 30, 50, 70)
d_high   = TRIANGULAR(dirtiness_value, 50, 100, 100)
l_small  = TRIANGULAR(load_size_value, 0, 0, 50)
l_medium = TRIANGULAR(load_size_value, 30, 50, 70)
l_large  = TRIANGULAR(load_size_value, 50, 100, 100)
// Rule Evaluation (MIN for AND)
R1 = MIN(d_low, l_small)        // Short
R2 = MIN(d_low, l_medium)       // Short
R3 = MIN(d_low, l_large)        // Medium
```

```
R4 = MIN(d_medium, l_small)     // Short
R5 = MIN(d_medium, l_medium)    // Medium
R6 = MIN(d_medium, l_large)     // Long
R7 = MIN(d_high, l_small)       // Medium
R8 = MIN(d_high, l_medium)      // Long
R9 = MIN(d_high, l_large)       // Long
// Aggregation (MAX for same outputs)
Short_strength  = MAX(R1, R2, R4)
Medium_strength = MAX(R3, R5, R7)
Long_strength   = MAX(R6, R8, R9)
// Defuzzification (Centroid)
numerator = 0
denominator = 0
FOR t = 0 TO 60 STEP 1
   mu_short  = MIN(Short_strength, TRIANGULAR(t, 0, 0, 20))
   mu_medium = MIN(Medium_strength, TRIANGULAR(t, 15, 30, 45))
   mu_long   = MIN(Long_strength, TRIANGULAR(t, 40, 60, 60))

   aggregated = MAX(mu_short, mu_medium, mu_long)

   numerator = numerator + (t * aggregated)
   denominator = denominator + aggregated
END FOR
IF denominator ≠ 0 THEN
   crisp_output = numerator / denominator
ELSE
   crisp_output = 0
END IF
IF crisp_output < 18 THEN
   label = "Short"
ELSE IF crisp_output < 42 THEN
   label = "Medium"
ELSE
   label = "Long"
END IF
PRINT crisp_output
PRINT label
END
```

Test Cases

No.	Dirtiness	Load Size	Wash Time (min)	Label	Description
1	20	15	≈ Short	Short	Low dirt, small load
2	50	50	≈ Medium	Medium	Medium dirt, medium load
3	85	90	≈ Long	Long	High dirt, large load
4	30	80	Between Medium–Long	Medium/Long	Borderline dirt, large load
5	75	30	≈ Medium	Medium	High dirt, small load

Case-by-Case Analysis

Case 1: (20, 15) → Short

Low dirtiness and small load strongly activate Rule 1.

Lightly soiled small loads logically require minimal washing time.

Case 2: (50, 50) → Medium

Both inputs are centered in their ranges.

Rule 5 (Medium & Medium → Medium) dominates, producing balanced wash time.

Case 3: (85, 90) → Long

Very dirty clothes and a full drum strongly activate Rule 9.

Maximum washing effort is required.

Case 4: (30, 80) → Medium/Long

Dirtiness is between Low and Medium while load is Large.

Rules 3 and 6 partially activate, producing blended output.

This demonstrates fuzzy logic's smooth handling of borderline cases.

Case 5: (75, 30) → Medium

Clothes are very dirty but the load is small.

Rule 7 (High & Small → Medium) dominates.

Small load reduces required washing time despite high dirt.

Possible Extensions

- Assign different weights (e.g., Dirtiness more important than Load Size)

- Add new input: Fabric Type (Delicate/Normal/Heavy)
- Compare centroid with other defuzzifiers (Mean of Maxima)
- Implement on embedded systems (IoT washing machine controller)

Conclusion

The Fuzzy Washing Machine Controller is an example that is successful in modeling real-life uncertainty (washing condition) through fuzzy logic to smoothly determine the washing time, depending on dirtiness and the size of the load. Instead of the use of rigid thresholds, the system involves the use of overlapping membership functions and a well defined system of rules that are used to generate realistic and gradual transitions of the output. The centroid defuzzification method guarantees balanced and understandable crisp results. On the whole, fuzzy approach offers a flexible human-like decision making and is more effective than the crisp logic for the borderline and mixed washing scenarios.

Colab Link

https://colab.research.google.com/drive/1iPtWP3rAwLY3offTUPHDwROqC2FjCCp6?usp=sharing