AY: 2024-2025 | **M1-S1: Dept. of Electrical Engineering**
MIDTERM | **AI-ECUE122** | **Teacher: A. Mhamdi**
Nov. 2024 | **Time Limit: 1h**

This document contains **7** pages numbered from **1/7** to **7/7**. As soon as it is handed over to you, make sure it is complete. The **3** tasks are independent and can be treated in the order that suits you.

The following rules apply:

❶ **A handwritten double-sided A4** sheet is permitted.

❷ **Any electronic material**, except basic calculator, is prohibited.

❸ **Mysterious or unsupported answers** will not receive full credit.

❹ **Round results** to the nearest thousandth *(i.e., third digit after the decimal point)*.

❺ **Task N°3:** Each correct answer will grant a mark with no negative scoring.

Task N°1

⌛ 20mn | (7 points)

Examine the code that follows:

```julia
using Printf, Plots, Fuzzy
x1axis = range(0, 10, length=100)
x1 = Dict{String, Fuzzy.MF}()
x1[”S”] = GaussianMF(1, 2)
x1[”M”] = BellMF(-2, 2, 5)
x1[”L”] = GaussianMF(9, 2)
x1_chart = chart_prepare(x1, x1axis)
x2axis = range(-4, 4, length=100)
x2 = Dict{String, Fuzzy.MF}()
x2[”N”] = TrapezoidalMF(-4, -4, -3, -1)
x2[”Z”] = BellMF(-2, 2, 0)
x2[”P”] = TrapezoidalMF(1, 3, 4, 4)
x2_chart = chart_prepare(x2, x2axis)
rule1 = Rule([”S”, ”N”], [-1., -1., 3.], ”MIN”)
rule2 = Rule([”S”, ”P”], [-1., 2., 1.], ”MIN”)
rule3 = Rule([”M”, ”Z”], [.5, .5, 0.], ”MAX”)
```

```
17 rule4 = Rule(["L", "N"], [2., -1., 1.], "MIN")
18 rule5 = Rule(["L", "P"], [2., 2., 0.], "MIN")
19 rules = [rule1, rule2, rule3, rule4, rule5]
20 fis = FISSugeno([x1, x2], rules)
21 @printf "First test: %.3f\n" eval_fis(fis, [2., -2.])
22 @printf "Second test: %.3f\n" eval_fis(fis, [6.5, 3.])
```

Each membership function utilized in the code is depicted in FIG. 1, p. 4.

(a) ($\frac{1}{2}$ point) What type of FIS is used in the code.

Takagi Sugeno Kang

(b) ($2\frac{1}{2}$ points) Give the textual description of rules used in this setting.

<u>FUZZY RULE BASE:</u>

$\mathfrak{R}_1$: IF x_1 is **S** AND x_2 is **N**, THEN $y = -x_1 - x_2 + 3$.

$\mathfrak{R}_2$: IF x_1 is **S** AND x_2 is **P**, THEN $y = -x_1 + 2x_2 + 1$.

$\mathfrak{R}_3$: IF x_1 is **M** OR x_2 is **Z**, THEN $y = \dfrac{x_1 + x_2}{2}$.

$\mathfrak{R}_4$: IF x_1 is **L** AND x_2 is **N**, THEN $y = 2x_1 - x_2 + 1$.

$\mathfrak{R}_5$: IF x_1 is **L** AND x_2 is **P**, THEN $y = 2x_1 + 2x_2$.

(c) (2 points) What is the result of the instruction at line #21?

$$x_1 = 2 = \{(\mathbf{S},\ 0.86),\ (\mathbf{M},\ 0.14),\ (\mathbf{L},\ 0)\}$$
$$x_2 = -2 = \{(\mathbf{N},\ 0.50),\ (\mathbf{Z},\ 0.50),\ (\mathbf{P},\ 0)\}$$

$\mathfrak{R}_1$: x_1 is **S** $\wedge$ x_2 is **N**, $0.86 \min 0.5 = 0.5$ $(y = 3)$.

$\mathfrak{R}_2$: x_1 is **S** $\wedge$ x_2 is **P**, $0.86 \min 0 = 0$ $(y = -5)$.

$\mathfrak{R}_3$: x_1 is **M** $\vee$ x_2 is **Z**, $0.14 \max 0.5 = 0.5$ $(y = 0)$.

$\mathfrak{R}_4$: x_1 is **L** $\wedge$ x_2 is **N**, $0 \min 0 = 0$ $(y = 7)$.

$\mathfrak{R}_5$: x_1 is **L** $\wedge$ x_2 is **P**, $0 \min 0 = 0$ $(y = 0)$.

$$y^\star = \frac{0.5 \times 3 + 0.5 \times 0}{0.5 + 0.5} = 1.5$$

```
First test: 1.500
```

(d) (2 points) What is the result of the instruction at line #22?

$$x_1 = 6.5 = \{(\mathbf{S},\ 0),\ (\mathbf{M},\ 0.82),\ (\mathbf{L},\ 0.43)\}$$

$$x_2 = 3 = \{(\mathbf{N},\ 0),\ (\mathbf{Z},\ 0.14),\ (\mathbf{P},\ 1)\}$$

$\mathfrak{R}_1$: x_1 is **S** $\wedge$ x_2 is **N**, $0 \min 0 = 0$ ($y = -6.5$).

$\mathfrak{R}_2$: x_1 is **S** $\wedge$ x_2 is **P**, $0 \min 1 = 0$ ($y = 0.5$).

$\mathfrak{R}_3$: x_1 is **M** $\vee$ x_2 is **Z**, $0.82 \max 0.14 = 0.82$ ($y = 4.75$).

$\mathfrak{R}_4$: x_1 is **L** $\wedge$ x_2 is **N**, $0.43 \min 0 = 0$ ($y = 11$).

$\mathfrak{R}_5$: x_1 is **L** $\wedge$ x_2 is **P**, $0.43 \min 1 = 0.43$ ($y = 19$).

$$y^\star = \frac{0.82 \times 4.75 + 0.43 \times 19}{0.82 + 0.43} \approx 9.652$$

`Second test: 9.652`

Task N$^{\underline{o}}$2 ⌛ 20mn | (5 points)

Consider a fuzzy logic system used to control the output y. The two inputs are x_1 and x_2. The membership functions of the fuzzy variables are described below.

- $x_1 \in [0,\ 10]$: **(P)** $\mathcal{L}(2,\ 4)$ **(G)** $\Delta(3,\ 5,\ 7)$ **(E)** $\Gamma(6,\ 8)$.
- $x_2 \in [0,\ 10]$: **(R)** $\mathcal{L}(1,\ 3)$ **(D)** $\Gamma(7,\ 9)$.
- $y \in [0,\ 30]$: **(C)** $\Delta(0,\ 5,\ 10)$ **(A)** $\Delta(10,\ 15,\ 20)$ **(G)** $\Delta(20,\ 25,\ 30)$.

FUZZY RULE BASE:

$\mathfrak{R}_1$: IF x_1 is **P** OR x_2 is **R**, THEN y is **C**.

$\mathfrak{R}_2$: IF x_1 is **G**, THEN y is **A**.

$\mathfrak{R}_3$: IF x_1 is **E** OR x_2 is **D**, THEN y is **G**.

(a) (3 points) Draw the membership functions in FIG. 2, p. 4.

(b) (2 points) Find the output y if $x_2 = 3$ and $x_1 = 2$.

$$x_1 = 2 = \{(\mathbf{P},\ 1),\ (\mathbf{G},\ 0),\ (\mathbf{E},\ 0)\}$$

$$x_2 = 3 = \{(\mathbf{R},\ 0),\ (\mathbf{D},\ 0)\}$$

The only rule to be fired is $\mathfrak{R}_1$. Given an output increment of 1, the defuzzified output will be:

$$y^\star = \frac{1/5 \times \left(1 + 2^2 + 3^2 + 4^2 + 5^2 + 6 \times 4 + 7 \times 3 + 8 \times 2 + 9\right)}{1/5 \times (1+2+3+4+5+4+3+2+1)} = 5$$

AY: 2024-2025 **Full Name:** ....

M1-S1: Dept. of Electrical Engineering **ID:** ....

Midterm | **AI-ECUE122** **Class:** ....

Nov. 2024 **Room:** ....

Teacher: A. Mhamdi **Time Limit:** 1h

Answer Sheet

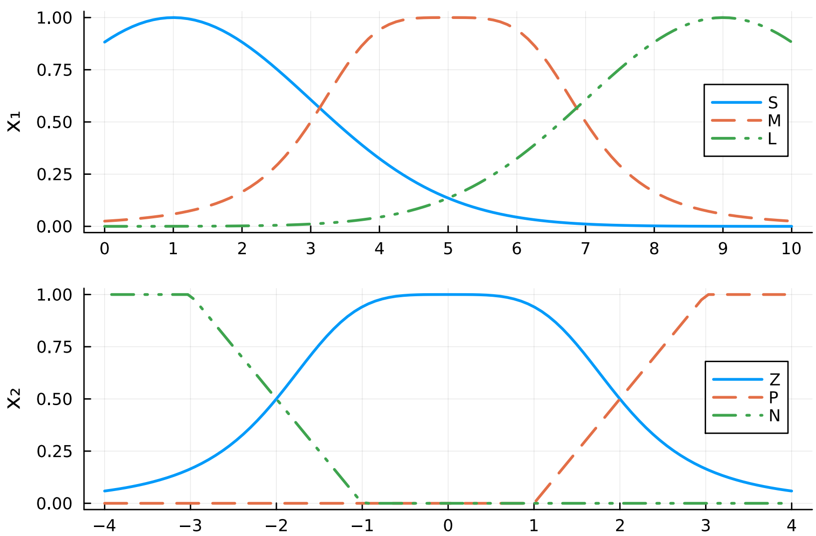

Fig. 1. Graphs of membership functions – task n°1.

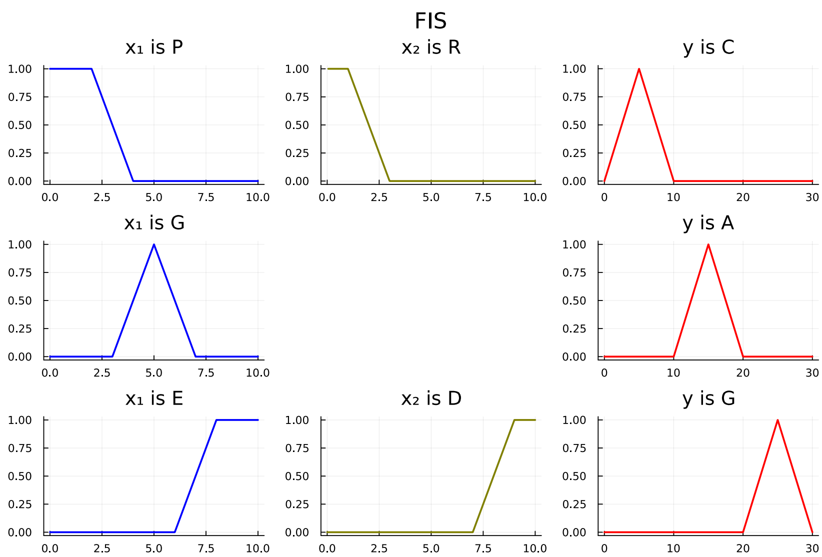

Fig. 2. Graphs of membership functions – task n°2.

Do not write anything here

Task N°3

⌛ 20mn | (8 points)

(a) ($\frac{1}{2}$ point) What is the main principle of fuzzy logic?

- ◯ Values can only be true or false.
- ◯ Logical operations are performed using binary arithmetic.
- ◯ Fuzzy logic requires crisp inputs only.
- √ A value can have degrees of truth between 0 and 1.

(b) ($\frac{1}{2}$ point) Which of the following is NOT a common operation in fuzzy logic systems?

√ Binary logic OR operation ◯ Fuzzification ◯ Defuzzification ◯ Inference

(c) ($\frac{1}{2}$ point) Which of the following statements best distinguishes Mamdani from TSK fuzzy systems?

- ◯ Mamdani systems use crisp outputs, while TSK systems use fuzzy outputs.
- ◯ TSK systems are limited to single-input, single-output models, while Mamdani can handle multi-input, multi-output models.
- √ Mamdani systems require defuzzification, whereas TSK systems can generate crisp outputs directly.
- ◯ Mamdani systems are computationally more efficient than TSK systems.

(d) ($\frac{1}{2}$ point) Which characteristic differentiates Tsukamoto fuzzy systems from Mamdani systems?

- ◯ Tsukamoto systems allow rule outputs to be represented as fuzzy sets.
- ◯ Mamdani systems use rule consequents defined by mathematical functions.
- √ In Tsukamoto systems, the rule output membership functions are always monotonic.

(e) ($\frac{1}{2}$ point) What is a key advantage of TSK fuzzy systems over Mamdani and Tsukamoto systems?

- ◯ TSK systems are better suited for handling vague and imprecise information.
- √ TSK systems are computationally more efficient because they use mathematical functions for outputs.
- ◯ TSK systems are easier to interpret due to their simpler rule base.

(f) ($\frac{1}{2}$ point) Which of the following is a key feature of `Julia`?

DO NOT WRITE ANYTHING HERE

✂ -

○ Julia is a statically-typed language.

○ Julia only supports object-oriented programming.

○ Julia cannot call functions written in other programming languages.

√ Julia combines performance close to C with dynamic typing.

(g) (½ point) What would be the output?

```julia
println("Hello" * " " * "World")
```

√ "Hello World" ○ "Hello * * World" ○ "Hello World!"

(h) (½ point) What will happen if you run the following code?

```julia
x = [1, 2, 3, 4]
println(x[5])
```

○ 0 ○ Nothing √ BoundsError ○ Null

(i) (½ point) Which of the following is true about Julia's package manager?

○ Julia does not support package management.

√ Packages are installed using the Pkg module.

○ Packages must be manually compiled before use.

○ Julia supports only a single package repository.

(j) (½ point) What is a benefit of using multiple dispatch in Julia?

○ It enforces strict inheritance rules for types.

○ It allows functions to be written with fewer arguments.

√ Efficient and modular code by selecting methods based on argument types.

○ It restricts the number of methods a function can have.

(k) (1 point) What does the following code snippet output?

```julia
a = 5
b = 5.0
println(typeof(a + b))
```

○ Number ○ Int64 √ Float64 ○ ERROR

(l) (1 point) What will the following code output?

```julia
x = [1, 2, 3, 4]
y = x
```

Do not write anything here

```
y[1] = 10
println(x)
```

○ [1, 2, 3, 4] ○ ERROR √ [10, 2, 3, 4]

(m) (1 point) What will happen if we run the following code?

```
function foo(x::Int, y::Int)
    return x * y
end

function foo(x::Float64, y::Float64)
    return x + y
end

print(foo(3, 4.), ' ', foo(2.5, 1.5))
```

○ 12 4.0 ○ 12.0 4 √ MethodError ○ 12 4