Artificial Intelligent - Logic

Dr Salem Ameen
S.A.AMEEN1@SALFORD.AC.UK

Lecture- Part A: Logical Agents Knowledge-based agents

Lecture- **Part A**: Logic

Lecture- **Part B**: Fuzzy Logic

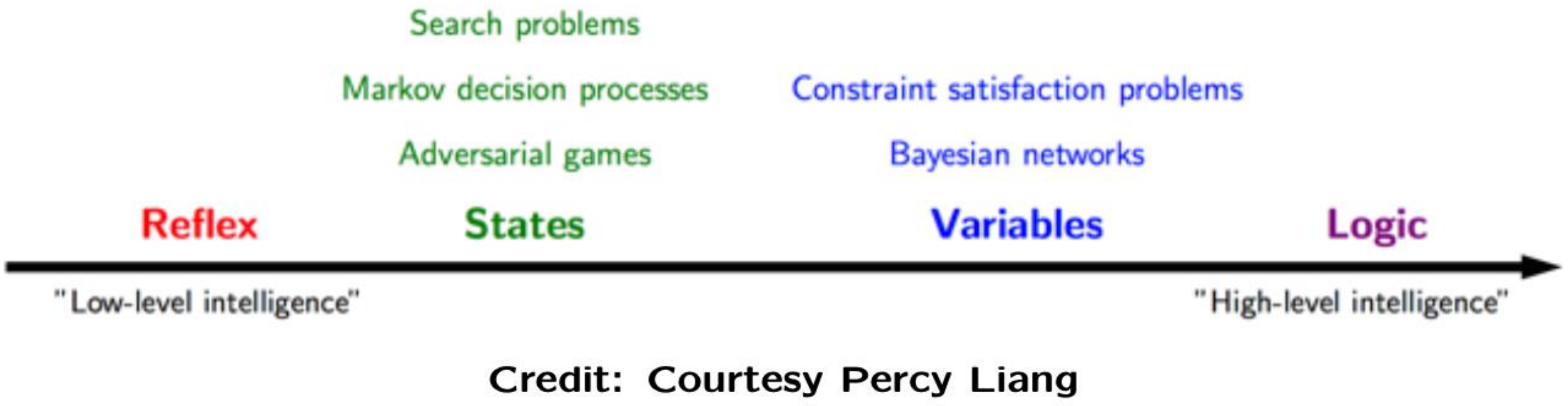
Lecture- Part A: Logical Agents Knowledge-based agents

Logical AI:

The idea is that an agent can represent knowledge of its world, its goals and the current situation by sentences in logic and decide what to do by inferring that a certain action or course of action is appropriate to achieve its goals.

John McCarthy in Concepts of logical AI, 2000.

Lecture- Part A: Logical Agents Knowledge-based agents

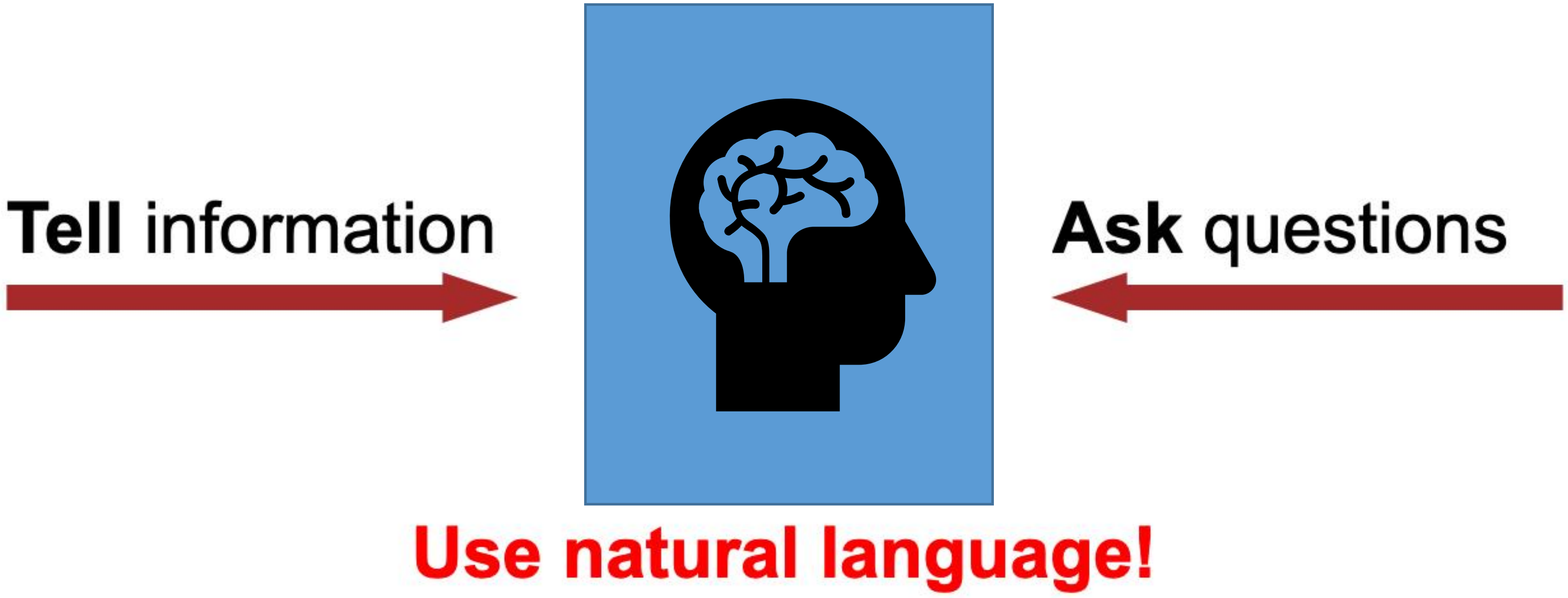


Lecture- Part A: Logical Agents
A historical note

Logic was dominant paradigm in AI before 1990s

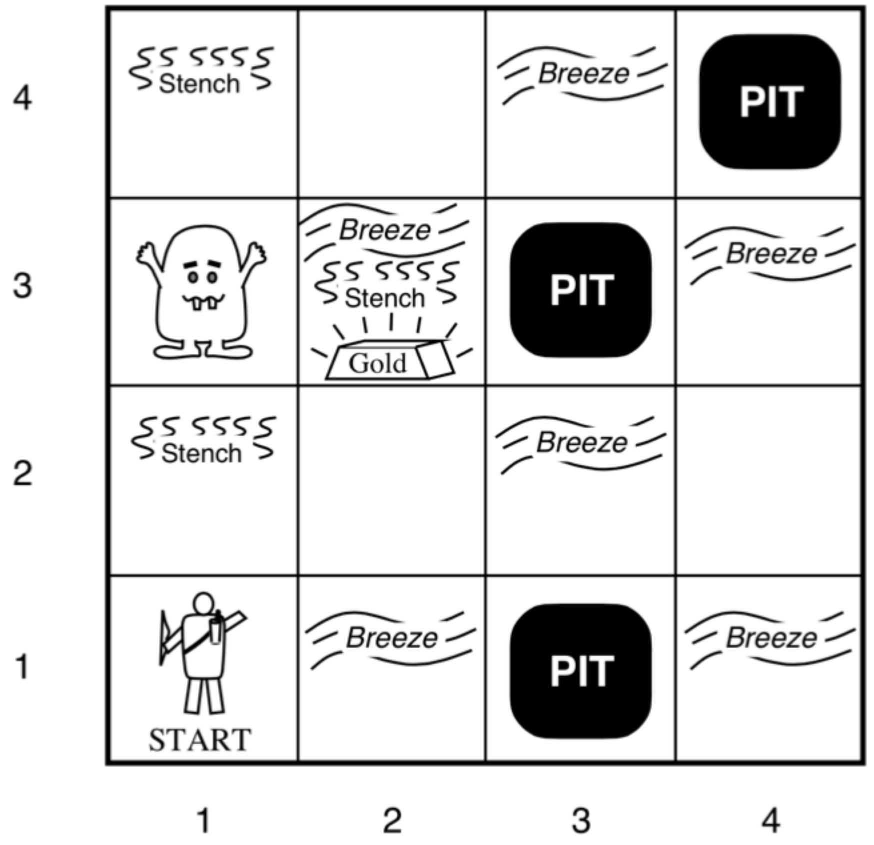
- Problem 1: deterministic, didn't handle uncertainty (probability addresses this)
- Problem 2: rule-based, didn't allow fine tuning from data (machine learning addresses this)

Lecture- Part A: Logical Agents
Motivation: smart personal assistant



Lecture- Part A: Logical Agents
The Wumpus World

Gregory Yob (1975)



Lecture- Part A: Logical Agents Exploring Wumpus World

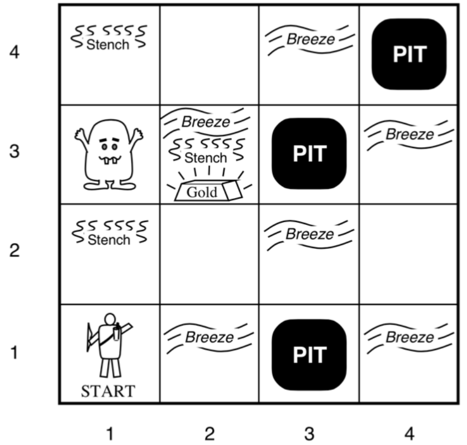
University of Salford MANCHESTER

Demo

Lecture- Part A: Logical Agents
The Wumpus World

- 4X4gridofrooms
- Squares adjacent to Wumpus are smelly and squares adjacent to pit are breezy
- Glitter iff gold is in the same square Shooting kills Wumpus if you are facing it
- Wumpus emits a horrible scream when it is killed that can be heard anywhere
- Shooting uses up the only arrow
- Grabbing picks up gold if in same square
- Releasing drops the gold in same square

Gregory Yob (1975)



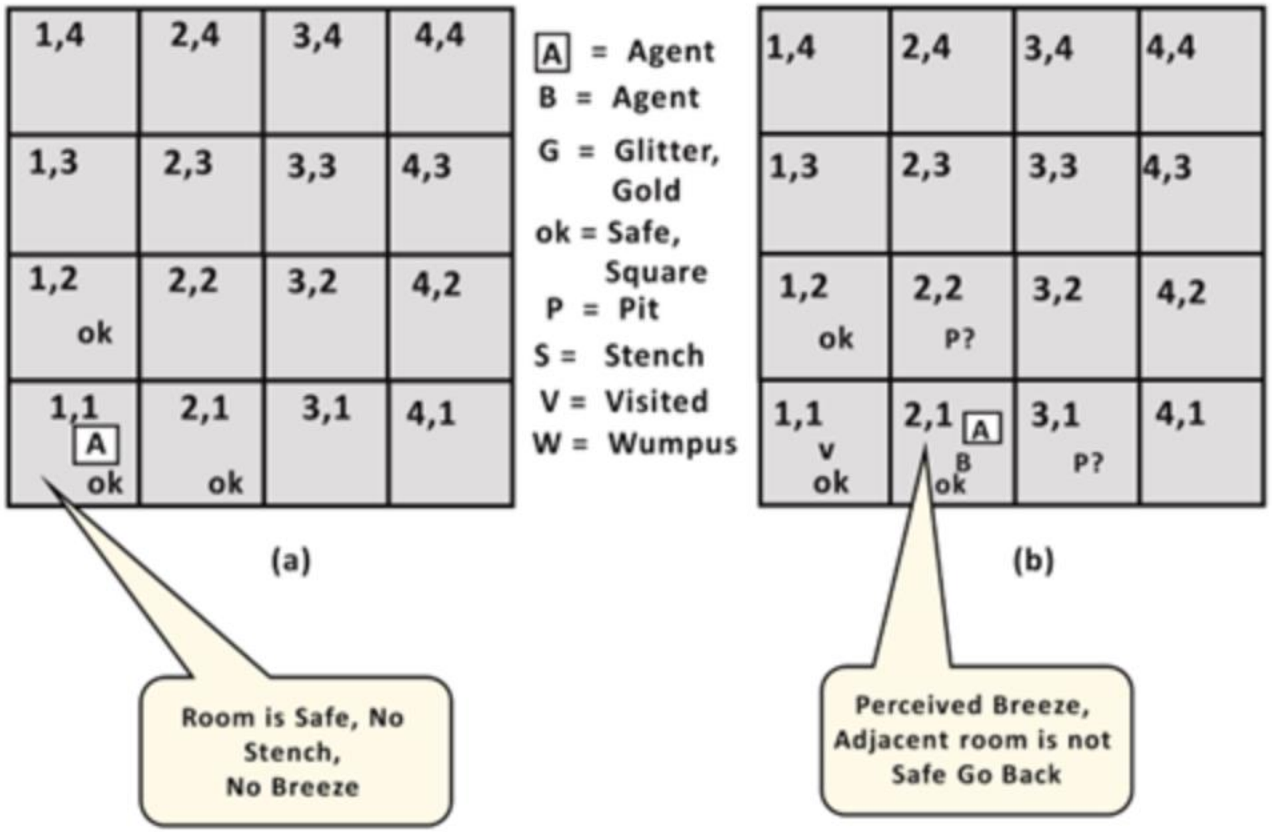
Lecture- Part A: Logical Agents
The Wumpus World

University of Salford MANCHESTER

- **Performance measure**: gold +1000, -10 for using the arrow. The games ends either when the agent dies or comes out of the cave.
- **Environment**
  - 4X4 grid of rooms
  - Agent starts in square [1,1] facing to the right
  - Locations of the gold, and Wumpus are chosen randomly with a uniform distribution from all squares except [1,1]
  - Each square other than the start can be a pit with probability of 0.2

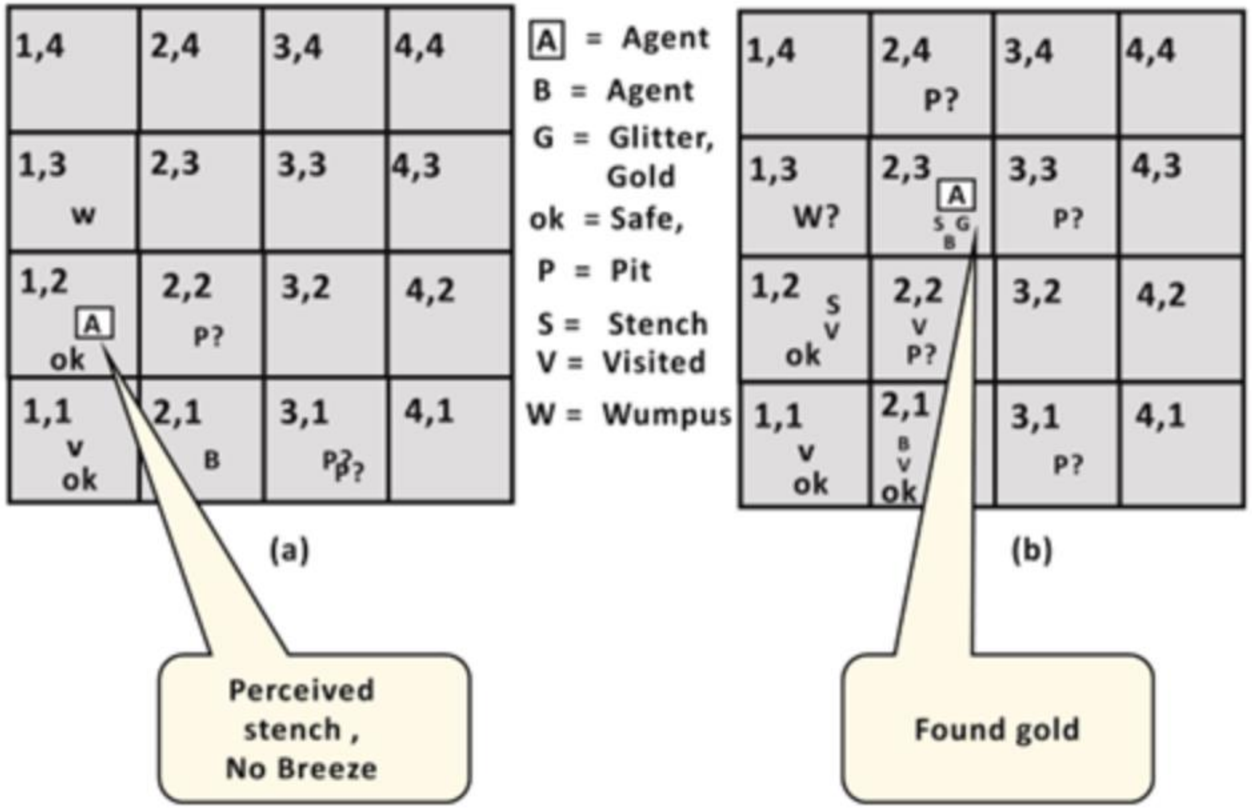
Lecture- Part A: Logical Agents
The Wumpus World

- **Agent's First step:**



Lecture- Part A: Logical Agents
The Wumpus World

- **Agent's later steps:**



Lecture- Part A: Logical Agents Logic

- **Knowledge base**: a set of sentences in a formal representation, logic
- **Logics**: are formal languages for representing knowledge to extract conclusions
  - **Syntax**: defines well-formed sentences in the language
  - **Semantic**: defines the truth or meaning of sentences in a world
- **Inference**: a procedure to derive a new sentence from other ones.
- **Logical entailment**: is a relationship between sentences. It means that a sentence follows logically from other sentences $KB \models \alpha$

Lecture- Part A: Logical Agents Types of formal mathematical logic

- Propositional logic
  - Propositions are interpreted as true or false
  - Infer truth of new propositions
- First order logic
  - Contains predicates, quantifiers and variables
    - E.g. Philosopher(a) → Scholar(a)
    - $\forall x$, King(x) $\wedge$ Greedy (x) $\Rightarrow$ Evil (x)
  - Variables range over individuals (domain of discourse)
- Second order logic
  - Quantify over predicates and over sets of variables

Lecture- Part A: Logical Agents
Other logics

- Temporal logic
  - Truths and relationships change and depend on time
- Fuzzy logic
  – Uncertainty, contradictions

Lecture- Part A: Logical Agents Logic / Propositional logic

University of Salford MANCHESTER

Propositional logic (PL) is the simplest logic.

- **Syntax of PL**: defines the allowable sentences or propositions.
- **Definition (Proposition):** A proposition is a declarative statement that's either True or False. (2+2=4 is true proposition, $W_{1,3}$ is true if Wumpus in [1,3])
  - **Atomic proposition**: single proposition symbol. Each symbol is a proposition. Notation: upper case letters and may contain subscripts. (2+2=4, $W_{1,3}$)
  - **Compound proposition**: constructed from from atomic propositions using parentheses and logical connectives. ($W_{1,3} \wedge P_{3,1}$)

Lecture- Part A: Logical Agents Propositional logic/ Atomic proposition

University of Salford MANCHESTER

Examples of atomic propositions:

- 2+2=4 is a true proposition
- $W_{1,3}$ is a proposition. It is true if there is a Wumpus in [1,3]
- "If there is a stench in [1,2] then there is a Wumpus in [1,3]" is a proposition
- "How are you?" or "Hello!" are not propositions. In general, statement that are questions, commands, or opinions are not propositions.

Lecture- Part A: Logical Agents Propositional logic/Compound propo.

University of Salford MANCHESTER

Examples of compound/complex propositions:

Let p, p_1, and p_2 be propositions

- **Negation** $\neg p$ is also a proposition. We call a **literal** either an atomic proposition or its negation. E.g., $W_{1,3}$ is a positive literal, and $\neg W_{1,3}$ is a negative literal.
- **Conjunction** $p_1 \wedge p_2$. E.g., $W_{1,3} \wedge P_{3,1}$
- **Disjunction** $p_1 \vee p_2$ E.g., $W_{1,3} \vee P_{3,1}$
- **Implication** $p_1 \rightarrow p_2$. E.g., $W_{1,3} \wedge P_{3,1} \rightarrow \neg W_{2,2}$
- **If and only if** $p_1 \leftrightarrow p_2$. E.g., $W_{1,3} \leftrightarrow \neg W_{2,2}$

Lecture- Part A: Logical Agents Model

A **model** in propositional logic is an assignment of truth values to propositional symbols.

Example:

- 3 propositional symbols: A, B, C
- $2^3 = 8$ possible models w:

$$\begin{array}{c}
\{A:0, B:0, C:0\} \\
\{A:0, B:0, C:1\} \\
\{A:0, B:1, C:0\} \\
\{A:0, B:1, C:1\} \\
\{A:1, B:0, C:0\} \\
\{A:1, B:0, C:1\} \\
\{A:1, B:1, C:0\} \\
\{A:1, B:1, C:1\}
\end{array}$$

Lecture- Part A: Logical Agents
Truth tables / Building propositions

Negation:

p	$\neg p$
T	F
F	T

Conjunction:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Exclusive or:

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Implication:

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Biconditional or If and only if (IFF):

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Lecture- Part A: Logical Agents Truth tables / Building propositions

University of Salford MANCHESTER

Negation:

p	$\neg p$
T	F
F	T

Conjunction:

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Disjunction:

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Implication:

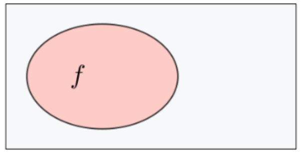
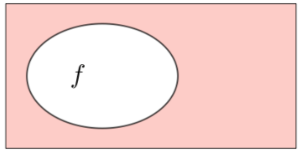
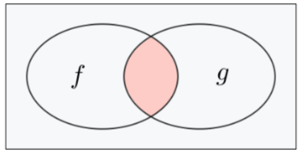
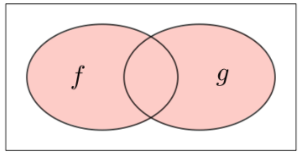
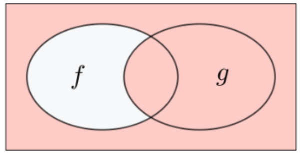
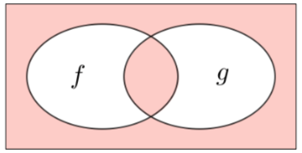
p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

Exclusive or:

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Biconditional or If and only if (IFF):

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Name	Symbol	Meaning	Illustration
Affirmation	f	f	
Negation	$\neg f$	not f	
Conjunction	$f \wedge g$	f and g	
Disjunction	$f \vee g$	f or g	
Implication	$f \rightarrow g$	if f then g	
Biconditional	$f \leftrightarrow g$	f, that is to say g	

Lecture- Part A: Logical Agents
Truth tables / Building propositions

p	q	r	¬r	p ∨ q	p ∨ q → ¬r
T	T	T	F	T	F
T	T	F	T	T	T
T	F	T	F	T	F
T	F	F	T	T	T
F	T	T	F	T	F
F	T	F	T	T	T
F	F	T	F	F	T
F	F	F	T	F	T

Lecture- Part A: Logical Agents
Truth tables / Properties of operators

- Commutativity: $p \wedge q = q \wedge p \qquad p \vee q = q \vee p$
- Associativity: $(p \wedge q) \wedge r = p \wedge (q \wedge r) \qquad (p \vee q) \vee r = p \vee (q \vee r)$
- Identity element: $p \wedge True = p \qquad p \vee True = True$
- $\neg(\neg p) = p$
- $p \wedge p = p \qquad p \vee p = p$
- Distributivity:
  $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
  $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$
- $p \wedge (\neg p) = False$ and $p \vee (\neg p) = True$
- DeMorgan's laws:
  $\neg(p \wedge q) = (\neg p) \vee (\neg q)$
  $\neg(p \vee q) = (\neg p) \wedge (\neg q)$

Lecture- Part A: Logical Agents Tautology and contradiction

- **Tautology** is a proposition which is always true
- **Contradiction** is a proposition which is always false
- **Contingency** is a proposition which is neither a tautology or a contradiction

P	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Lecture- Part A: Logical Agents Contrapositive, inverse, etc.

- Given an **implication** $p \to q$
- The **converse** is: $q \to p$
- The **contrapositive** is: $\neg q \to \neg p$
- The **inverse** is: $\neg p \to \neg q$

Example: Hot is a sufficient condition for my going to the beach.

- The **implication** is:
- The **converse** is:
- The **contrapositive** is:
- The **inverse** is:

Lecture- Part A: Logical Agents Model Checking

University of Salford MANCHESTER

Truth Table for inference

- Model: assignment T/F to every propositional symbol.
- Check that α is true in every model in which KB is true.

Lecture- Part A: Logical Agents Inference / Modus Ponens

$$\frac{p \qquad p \to q}{q}$$

$$p \wedge (p \to q) \Rightarrow q$$

$$\frac{\mathit{warm} \qquad \mathit{warm} \to \mathit{sunny}}{\mathit{sunny}}$$

Modus Ponens: "If A is true, then B is true. A is true. Therefore, B is true."

If it is a car, then it has wheels. It is a car. Therefore, it has wheels.

Lecture- Part A: Logical Agents Inference / Modus Tollens

$$\frac{\neg q \qquad p \rightarrow q}{\neg p}$$

$$\frac{\neg beach \qquad hot \rightarrow beach}{\neg hot}$$

Modus Ponens: "If A is true, then B is true. A is true. Therefore, B is true."
Modus Tollens: "If A is true, then B is true. B is not true. Therefore, A is not true

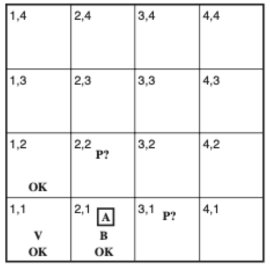
If it is a car, then it has wheels. It does not have wheels. Therefore, it is not a car.

Lecture- Part A: Logical Agents Inference / Common Rules

- Addition: $$\frac{p}{p \vee q}$$
- Simplification: $$\frac{p \wedge q}{q}$$
- Disjunctive-syllogism: $$\frac{\begin{array}{c} p \vee q \\ \neg p \end{array}}{q}$$
- Hypothetical-syllogism: $$\frac{\begin{array}{c} p \rightarrow q \\ q \rightarrow r \end{array}}{p \rightarrow r}$$

Lecture- Part A: Logical Agents Entailment / Wumpus world KB

Let's build the KB for the reduced Wumpus world.



- Let $P_{i,j}$ be true if there is a pit in $[i, j]$
- Let $B_{i,j}$ be true if there is a breeze in $[i, j]$

R_1: $\neg P_{1,1}$

- "A square is breezy if and only if there is an adjacent pit"

R_2: $B_{1,1} \Leftrightarrow P_{1,2} \lor P_{2,1}$

R_3: $B_{2,1} \Leftrightarrow P_{1,1} \lor P_{2,2} \lor P_{3,1}$

R_4: $\neg B_{1,1}$

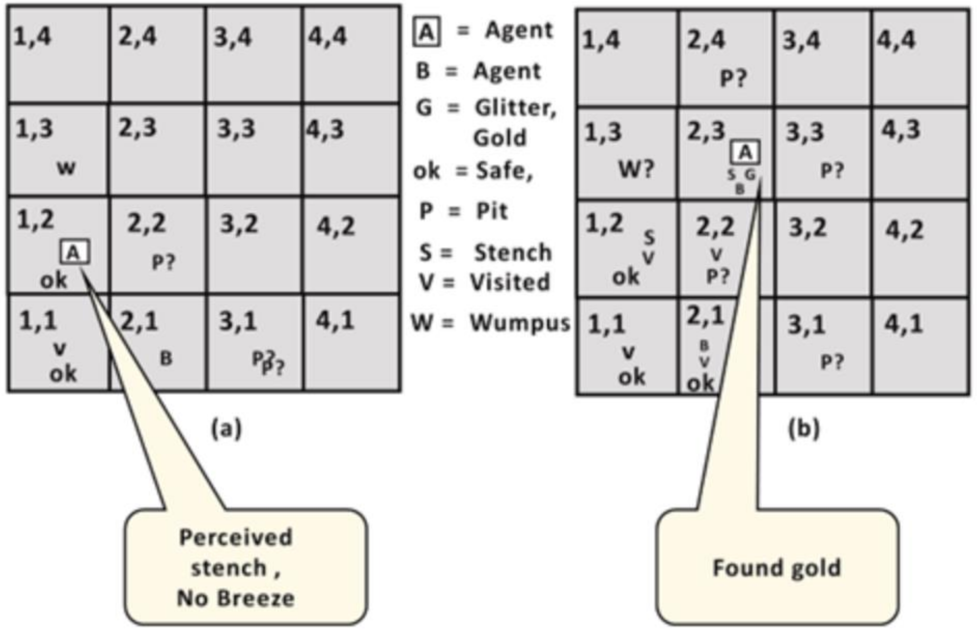
R_5: $B_{2,1}$

Questions: Based on this KB, is $KB \models P_{1,2}$? Is $KB \models P_{2,2}$?

Lecture- Part A: Logical Agents
Wumpus world KB / Model Checking

- Truth Table for inference
- Model: assignment T/F to every propositional symbol.
- Check that α is true in every model in which KB is true.

$B_{1,1}$	$B_{2,1}$	$P_{1,1}$	$P_{1,2}$	$P_{2,1}$	$P_{2,2}$	$P_{3,1}$	R_1	R_2	R_3	R_4	R_5	KB
false	*false*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*false*	*false*
false	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*false*	*true*	*false*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
false	*true*	*false*	*false*	*false*	*false*	*false*	*true*	*true*	*false*	*true*	*true*	*false*
false	*true*	*false*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*false*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*false*	*true*	*true*	*true*	*true*	*true*	*true*	*true*	*true*
false	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*false*	*false*	*true*	*true*	*false*
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
true	*true*	*true*	*true*	*true*	*true*	*true*	*false*	*true*	*true*	*false*	*true*	*false*

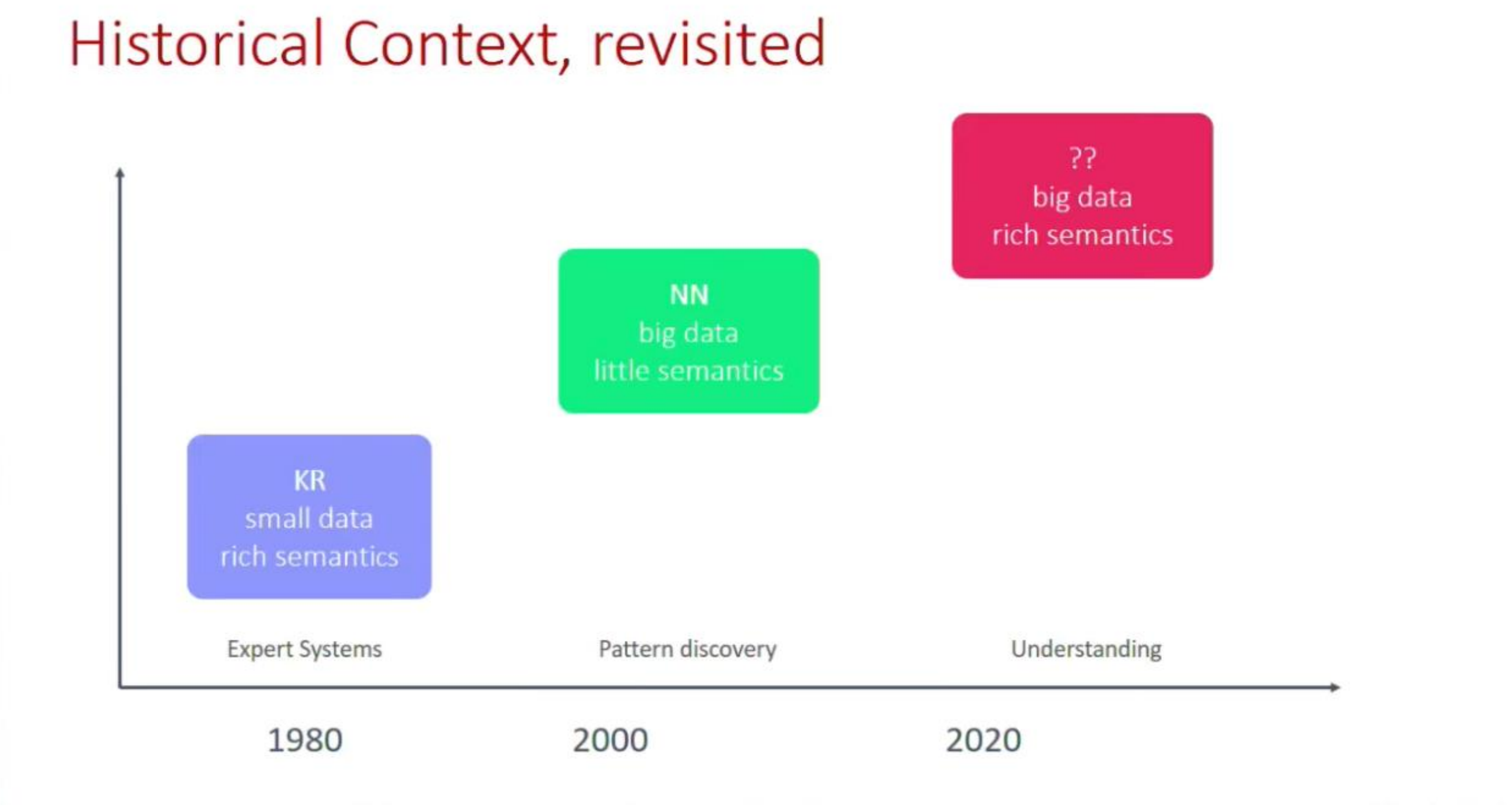


Questions: Based on this KB, is $KB \models P_{1,2}$? Is $KB \models P_{2,2}$?

Lecture- Part A: Logical Agents Summary/Logic

- Building logical agents was a main research trend in AI before the mid-nineties
- Logic is used in AI to represent the environment of the agent and reason about that environment
- **Pros** and **cons** of logical agents:
- \- Do not handle uncertainty, probability does
  \- Rule-based and do not use data, ML does
  \- It is hard to model every aspect of the world
- \+ Intelligibility of models: models are encoded explicitly

Lecture- Part A: Logical Agents Summary/Logic



Lecture- Part B: Fuzzy Logic Outline

- Background
- Fuzzy Set Theory
- Fuzzy Logic Controller
- Applications

Lecture- Part B: Fuzzy Logic Background

University of Salford MANCHESTER

- The idea of fuzzy sets and fuzzy logic were not accepted at that time (some of the underlying mathematics had not yet been explored).

Lotfi Aliasker Zadeh (February 4, 1921 – September 6, 2017)

Lecture- Part B: Fuzzy Logic Background

- "Fuzzy theory is wrong, wrong, and pernicious. What we need is more logical thinking, not less. The danger of fuzzy logic is that it will encourage the sort of imprecise thinking that has brought us so much trouble. Fuzzy logic is the cocaine of science."

  -Professor William Kahan UC Berkeley

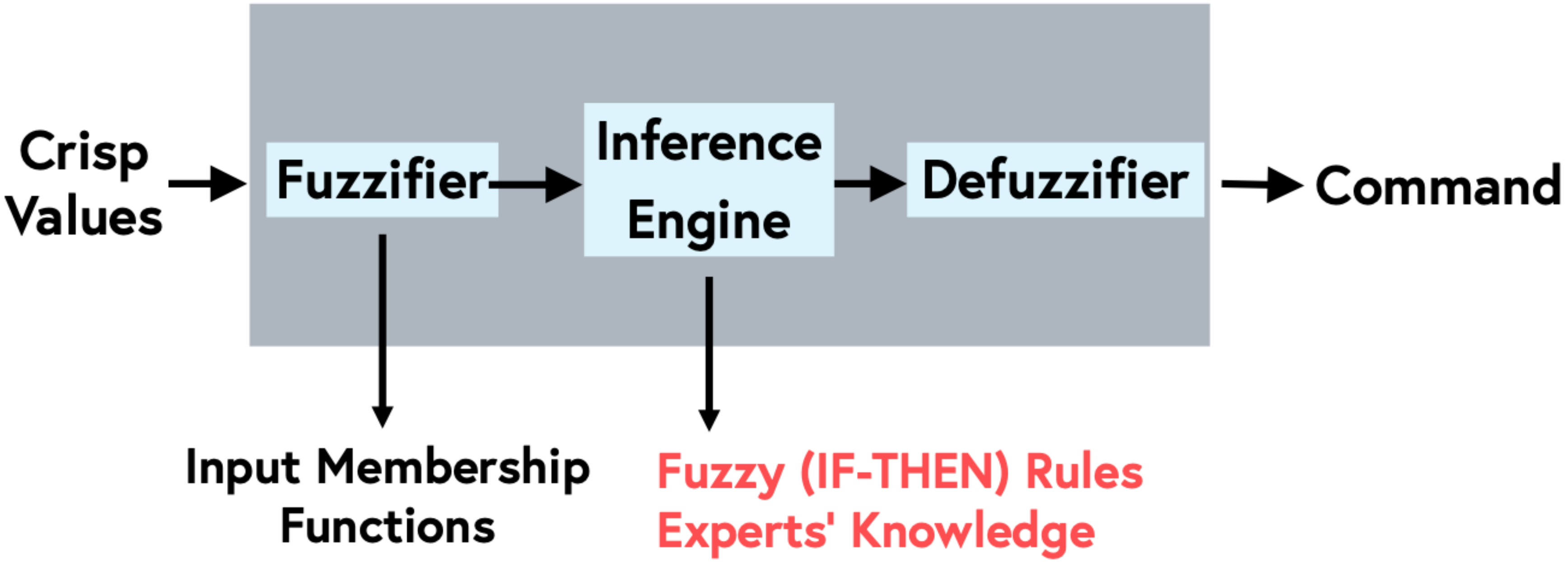
- In Japan, specifically, fuzzy logic was fully accepted and implemented in products.

- Revival in fuzzy logic in the US in the late 80s.

Lecture- Part B: Fuzzy Logic Background

- Fuzzy Logic = Computing with words
- Linguistic variables, that is, variables whose values are not number but words or sentences in a natural or artificial language
- Temperature
- T = { Very cold, Cold, Cool, Warm, Hot}

Lecture- Part B: Fuzzy Logic
Fuzzy Logic Controller (FLC)



Lecture- Part B: Fuzzy Logic Membership Function

University of Salford MANCHESTER

$$A = \{x \mid x \in X \text{ and } x \text{ is } \Delta\}$$

Δ comes from Temp.

& t is very hot

$$A_{Fuzzy} = \{(x, \mu_A(x)) \mid x \in X \text{ and } \mu_A(x) \text{ is } [0,1]\}$$

Membership Function: It quantifies the degree of membership of the element in X to the fuzzy set A.

- **x-axis** represents the universe of discourse.
- **y-axis** represents the degrees of membership in the [0, 1] interval.

Lecture- Part B: Fuzzy Logic Membership Function

- Temperature

- T = { Very cold, Cold, Cool, Warm, Hot}

.

Lecture- Part B: Fuzzy Logic Membership Function

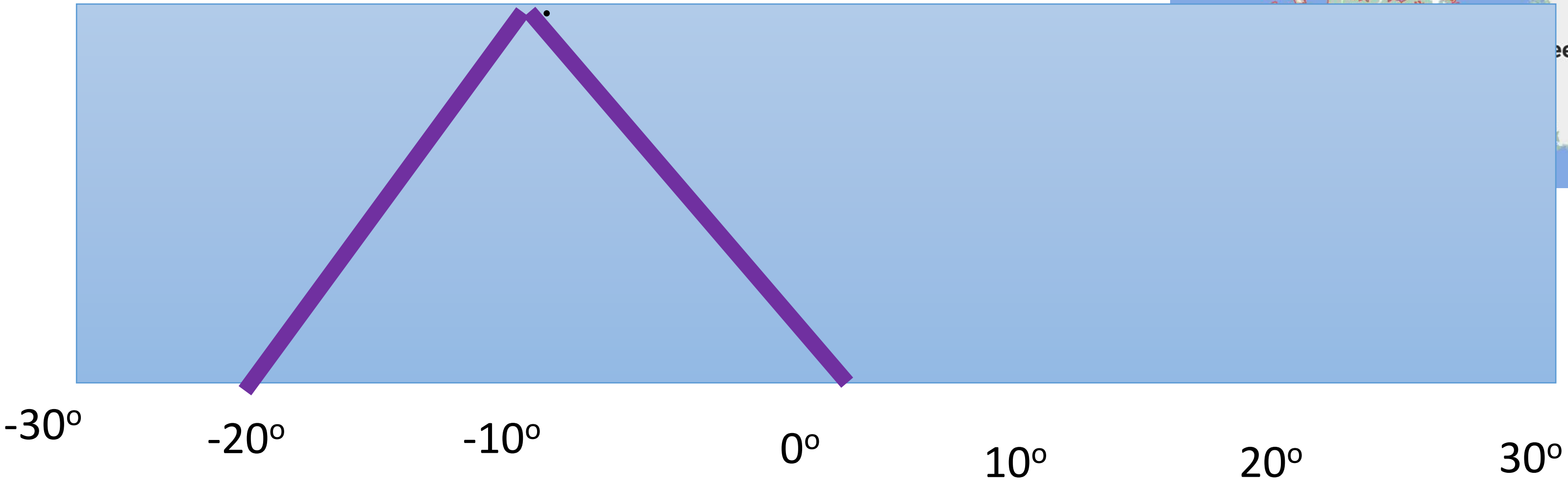
University of Salford MANCHESTER

- Temperature
- T = { **Very cold**, Cold, Cool, Warm, Hot}

Lecture- Part B: Fuzzy Logic Membership Function

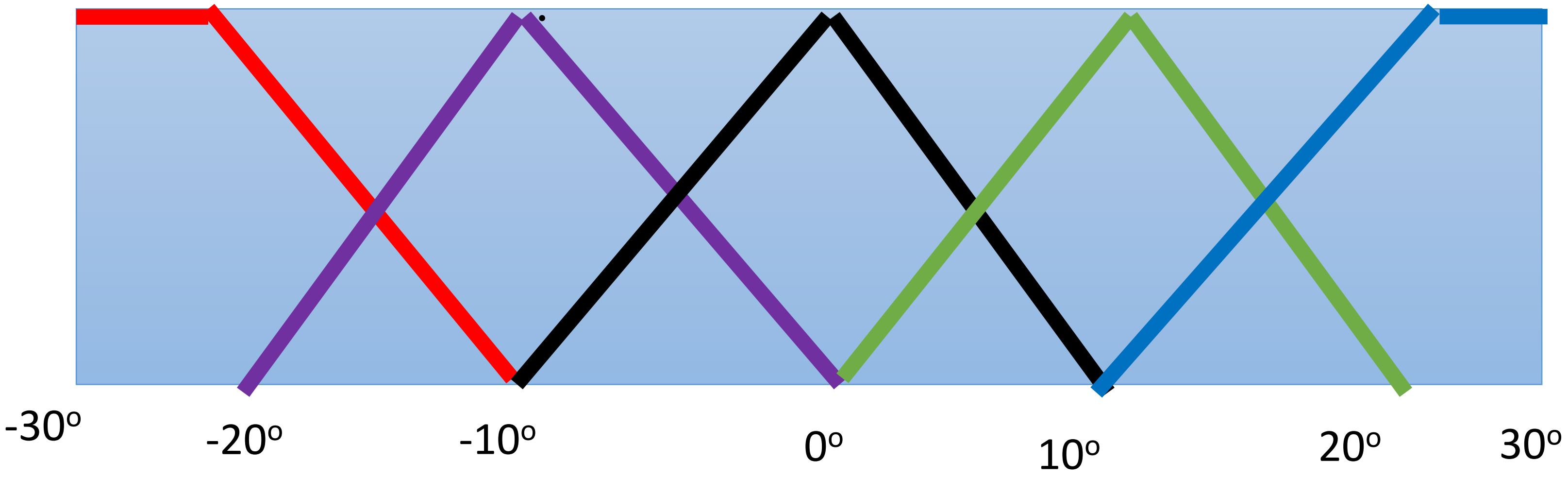
- Temperature

- T = { Very cold, **Cold**, Cool, Warm, Hot}



Lecture- Part B: Fuzzy Logic Membership Function

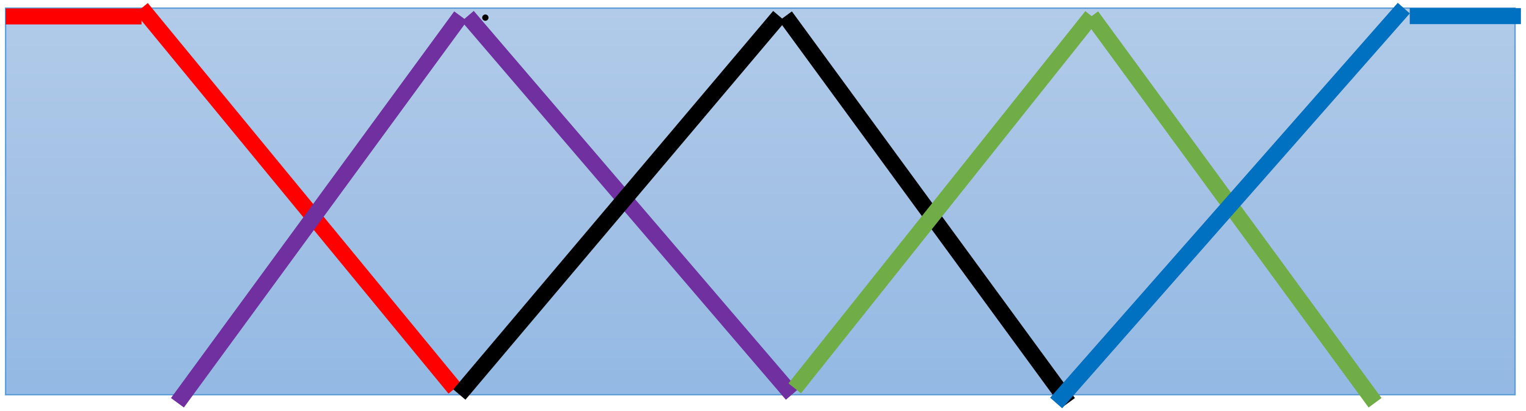
- Temperature
- T = { Very cold, Cold, Cool, Warm, Hot}



Lecture- Part B: Fuzzy Logic Membership Function

- Temperature

- T = { Very cold, Cold, Cool, Warm, Hot}



Lecture- Part B: Fuzzy Logic Membership Function

WORD

- Temperature
- T = { Very cold, Cold, Cool, Warm, Hot}

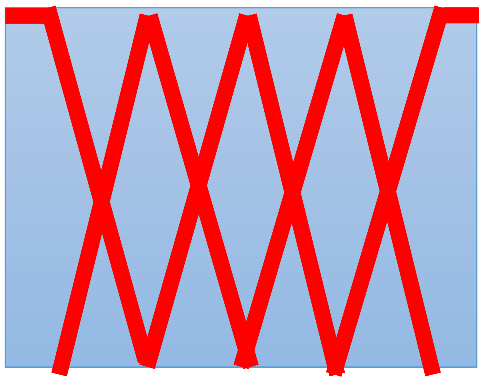
linguistic variable

Syntax

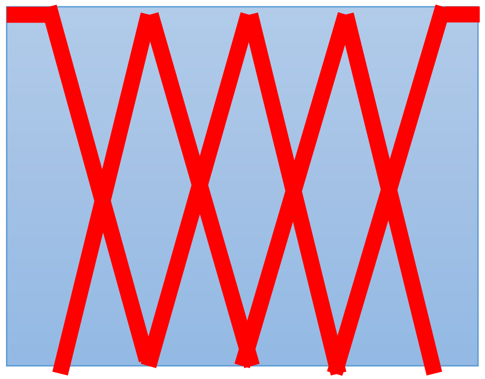
Set of terms

Semantic

FUZZY SET



-30 0 -30



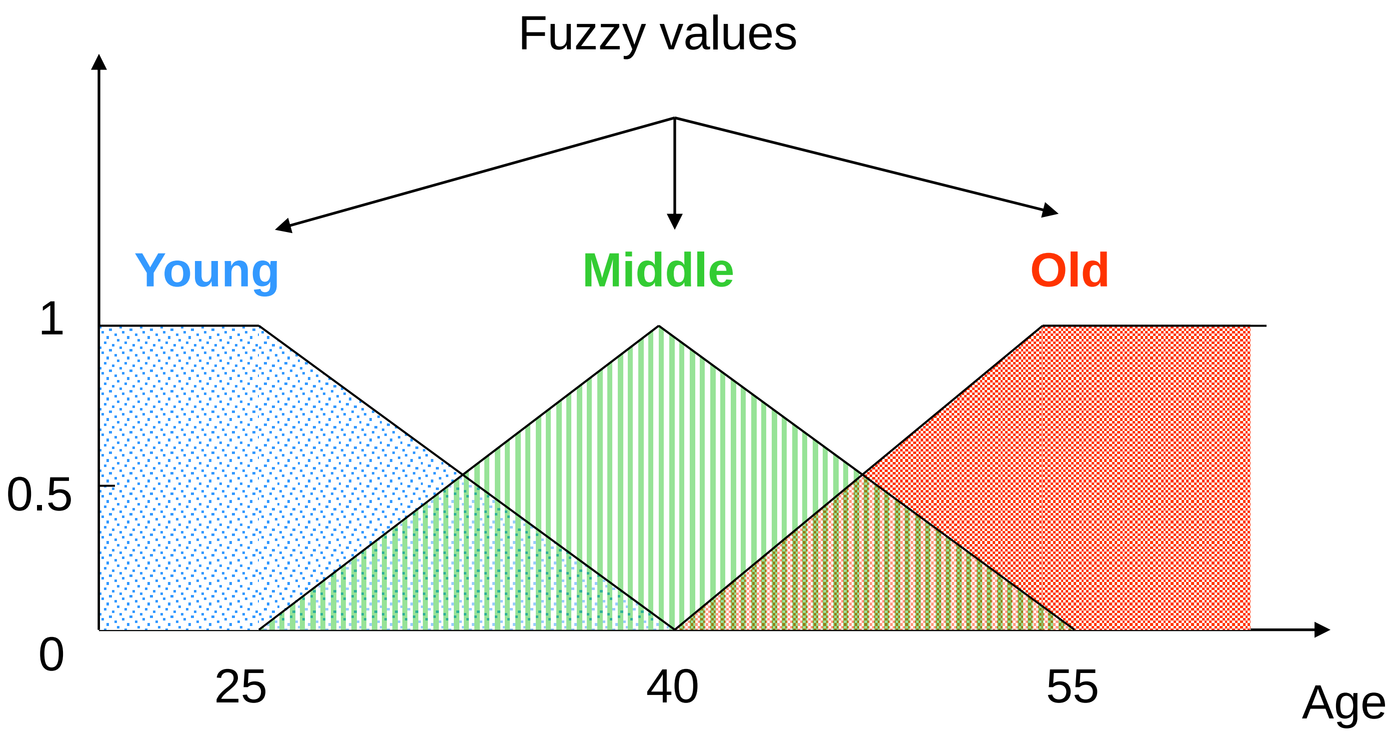
Universe of Discourse

Lecture- Part B: Fuzzy Logic Membership Function

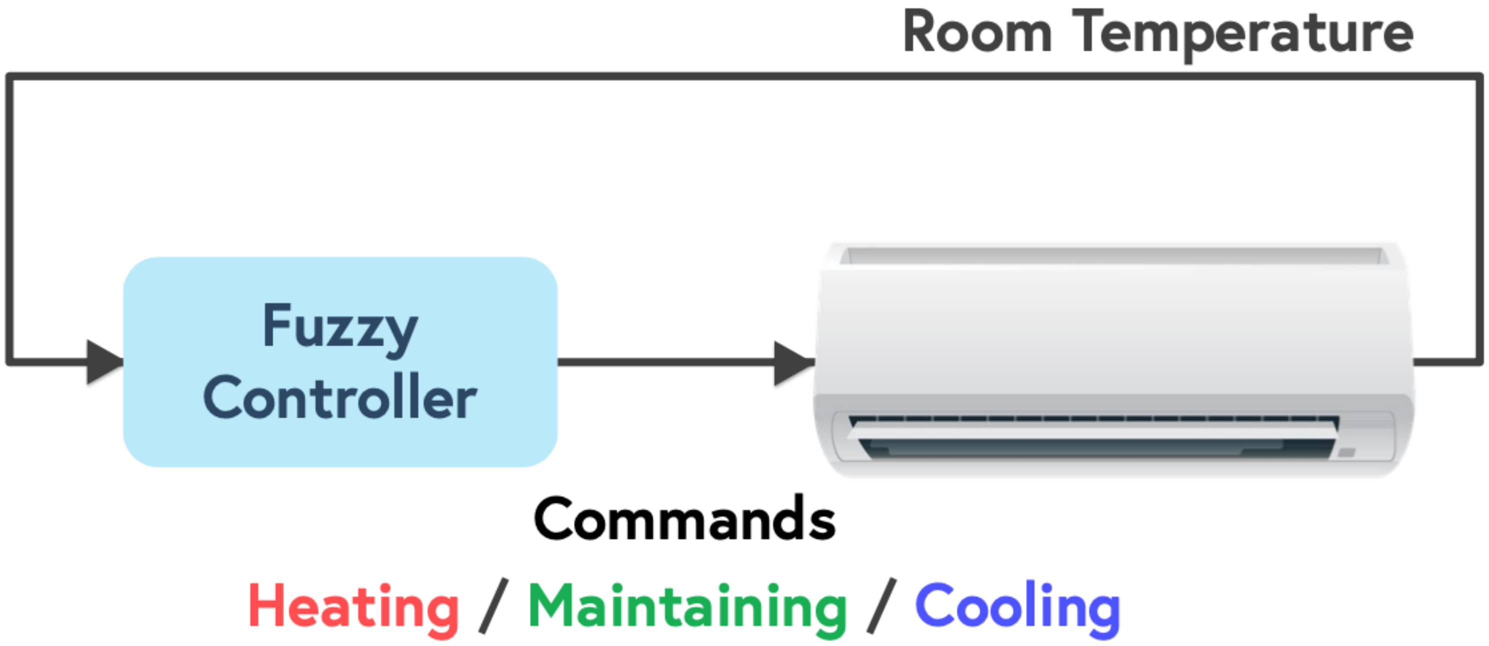
- Example **Young** :
  - Ann is 28, 0.8 in set “Young”
  - Bob is 35, 0.1 in set “Young”
  - Charlie is 23, 1.0 in set “Young”

- Unlike statistics and probabilities, the *degree* is not describing *probabilities* that the item is in the set, but instead describes *to what extent* the item is the set.

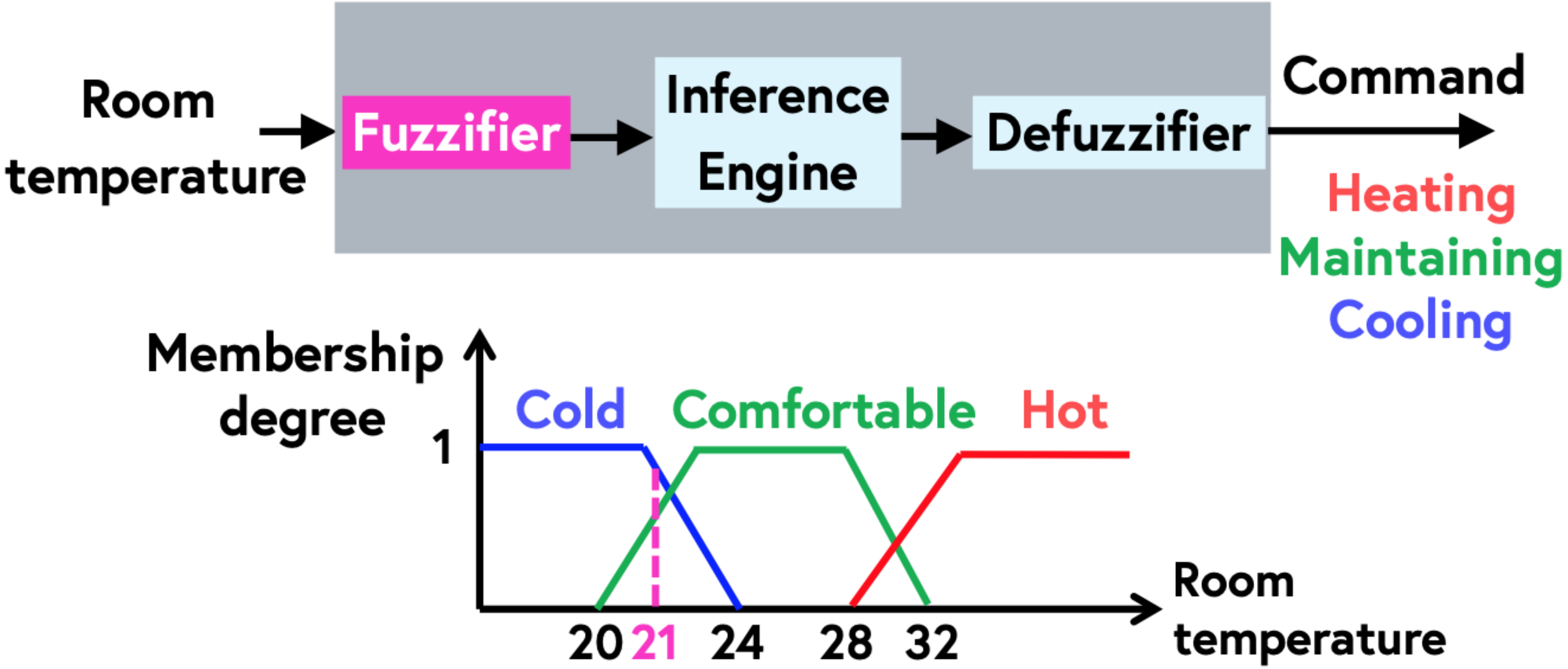
Lecture- Part B: Fuzzy Logic Membership function



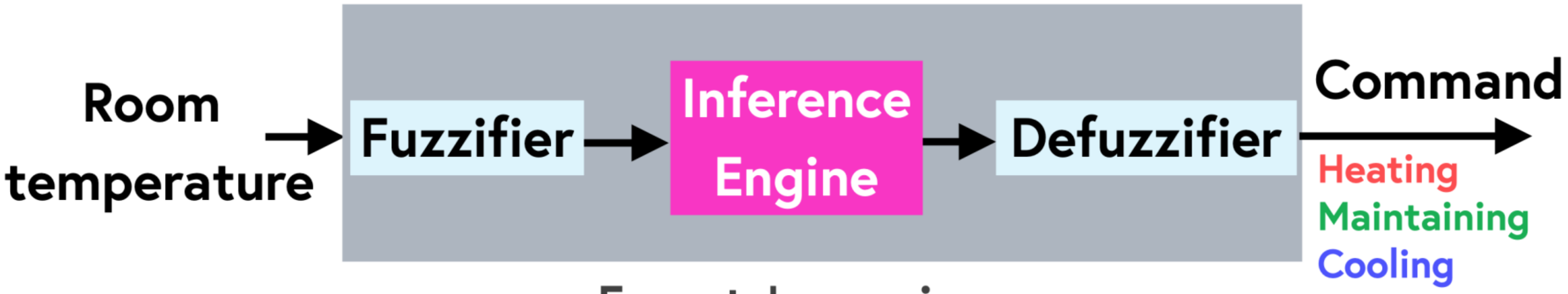
Lecture- Part B: Fuzzy Logic
Fuzzy Logic Controller (FLC) / Example



Lecture- Part B: Fuzzy Logic FLC/ Example/ Fuzzifier



Lecture- Part B: Fuzzy Logic FLC/ Example/ Inference Engine



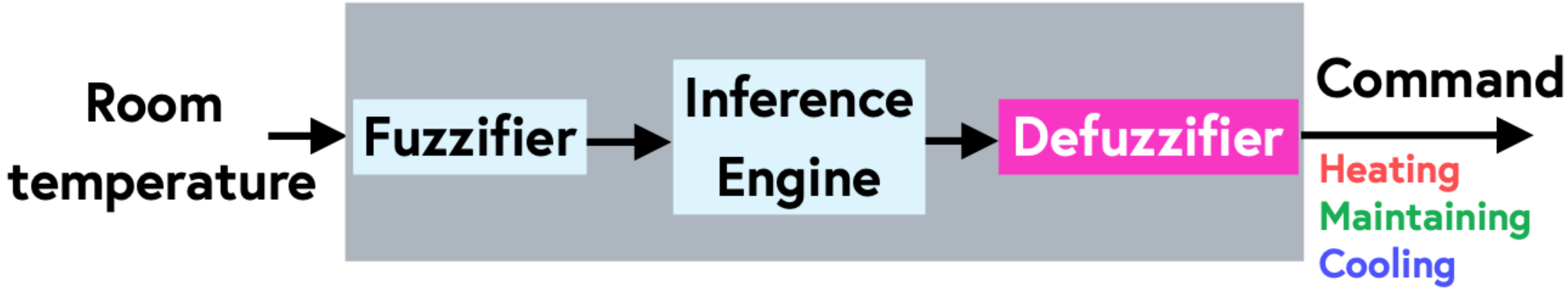
Experts' experience

- Rule 1: If the room temperature is Hot, then take the action of Cooling.
- Rule 2: If the room temperature is Comfortable, then take the action of Maintaining.
- Rule 3: If the room temperature is Cold, then take the action of Heating.

Modus Ponens: *"If A is true, then B is true. A is true. Therefore, B is true."*

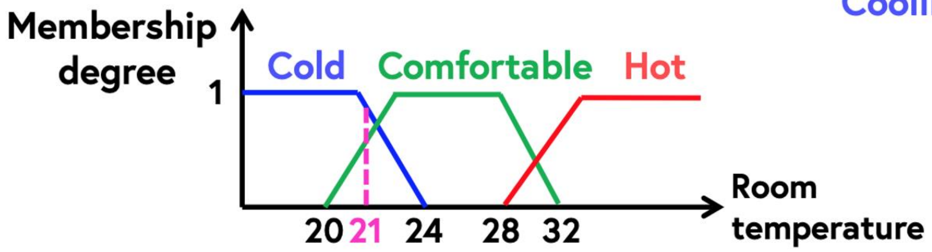
Modus Tollens: *"If A is true, then B is true. B is not true. Therefore, A is not true*

Lecture- Part B: Fuzzy Logic FLC/ Example/ Defuzzifier



Defuzzifier can determine:

- What percent of effort it would take to heat up the room temperature.
- What percent of effort it would take to cool down the room temperature.
- To maintain the room temperature.



Cooling

Center of gravity (COG).

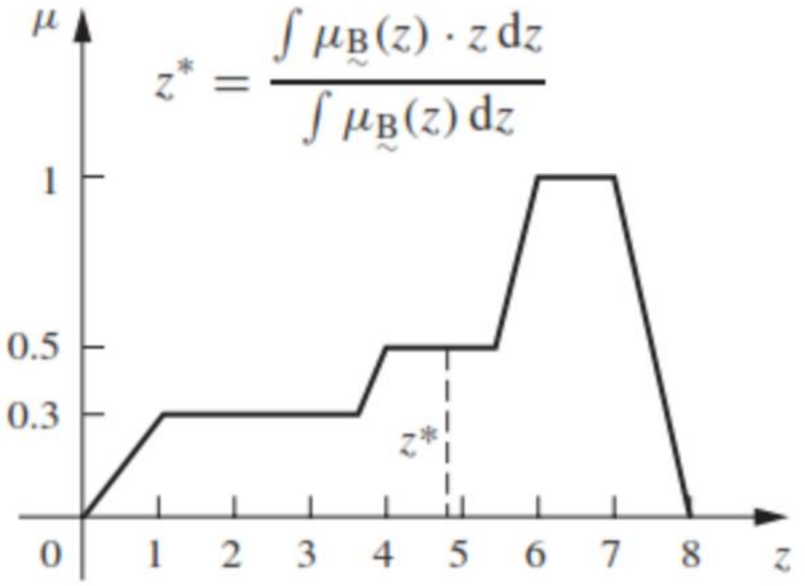
- Rule 3: If the room temperature is Cold, then take the action of Heating.

Lecture- Part B: Fuzzy Logic
Defuzzifier / Center of gravity (COG).

University of Salford MANCHESTER

$$\frac{\left[\int_0^1 (0.3z)z\,dz + \int_1^{3.6} (0.3)z\,dz + \int_{3.6}^{4} \left(\frac{z-3.0}{2}\right) z\,dz + \int_4^{5.5} (0.5)z\,dz + \int_{5.5}^{6} (z-5)z\,dz + \int_6^7 z\,dz + \int_7^8 (8-z)\,z\,dz\right]}{\left[\int_0^1 (0.3z)\,dz + \int_1^{3.6} (0.3)\,dz + \int_{3.6}^{4} \left(\frac{z-3.6}{2}\right) dz + \int_4^{5.5} (0.5)\,dz + \int_{5.5}^{6} \left(\frac{z-5.5}{2}\right) dz + \int_6^7 dz + \int_7^8 \left(\frac{7-z}{2}\right) dz\right]}$$

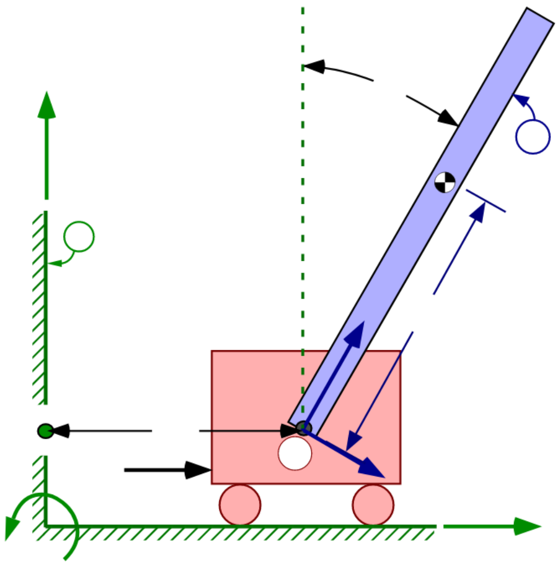
$$= 4.9\,\mathrm{m}$$

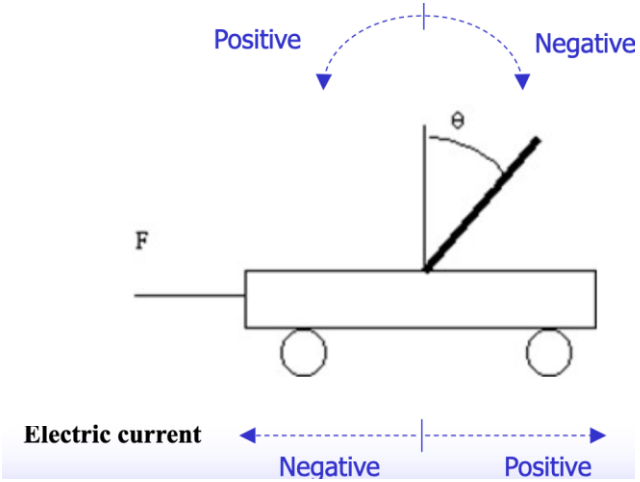


Lecture- Part B: Fuzzy Logic FLC/ Example/ Inverted Pendulum

- **Inputs**:
- 1. Error (difference) in Degree
- 2. Omega (Angular velocity)

- **Output**:
- Current (+/- for direction)



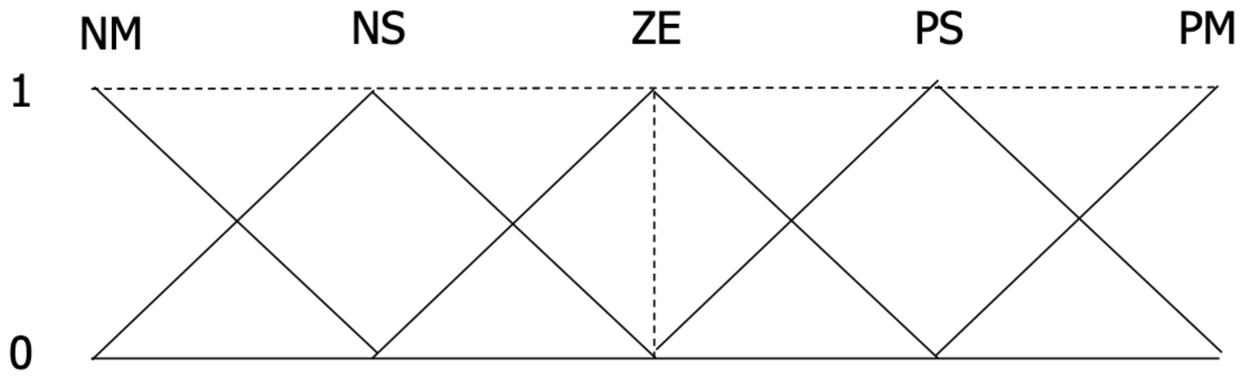


Lecture- Part B: Fuzzy Logic
The fuzzy operators

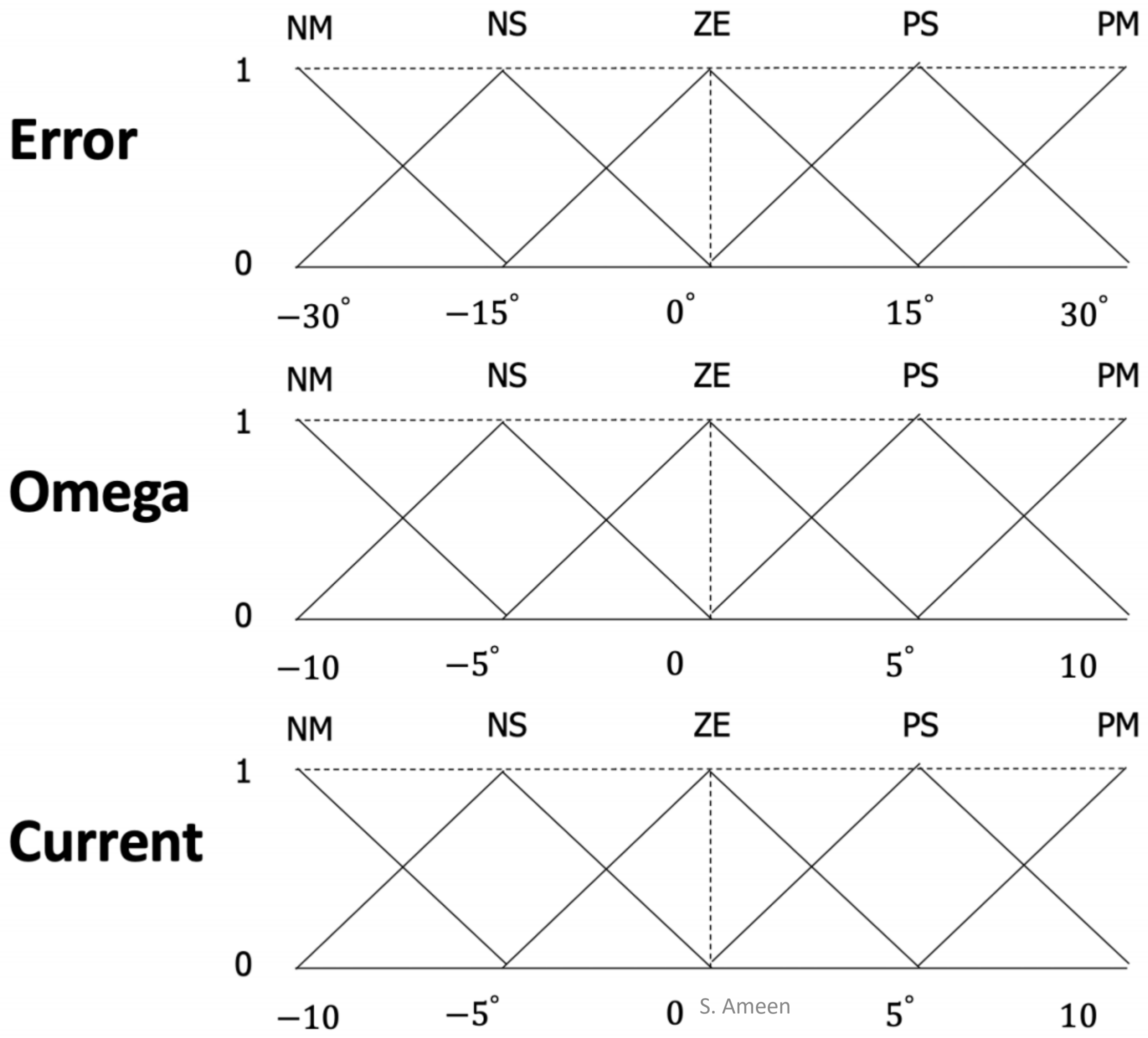
Name	Intersection AND: $\mu_{A\cap B}(x)$	Union OU: $\mu_{A\cup B}(x)$	Complement NOT: $\mu_{\bar{A}(x)}$
Zadeh Operators MIN/MAX	$min\left(\mu_A(x), \mu_B(x)\right)$	$max\left(\mu_A(x), \mu_B(x)\right)$	$1 - \mu_A(x)$

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Fuzzification

- Negative Small (NM)
- Negative Small (NS)
- Zero (ZE)
- Positive Small (PS)
- Positive Medium (PM)



Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Fuzzification



Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Inference Engine/Rules

University of Salford MANCHESTER

Error	Omega	Current
ZE	ZE	ZE
ZE	NM	PM
ZE	PS	NS
PM	**ZE**	**NM**

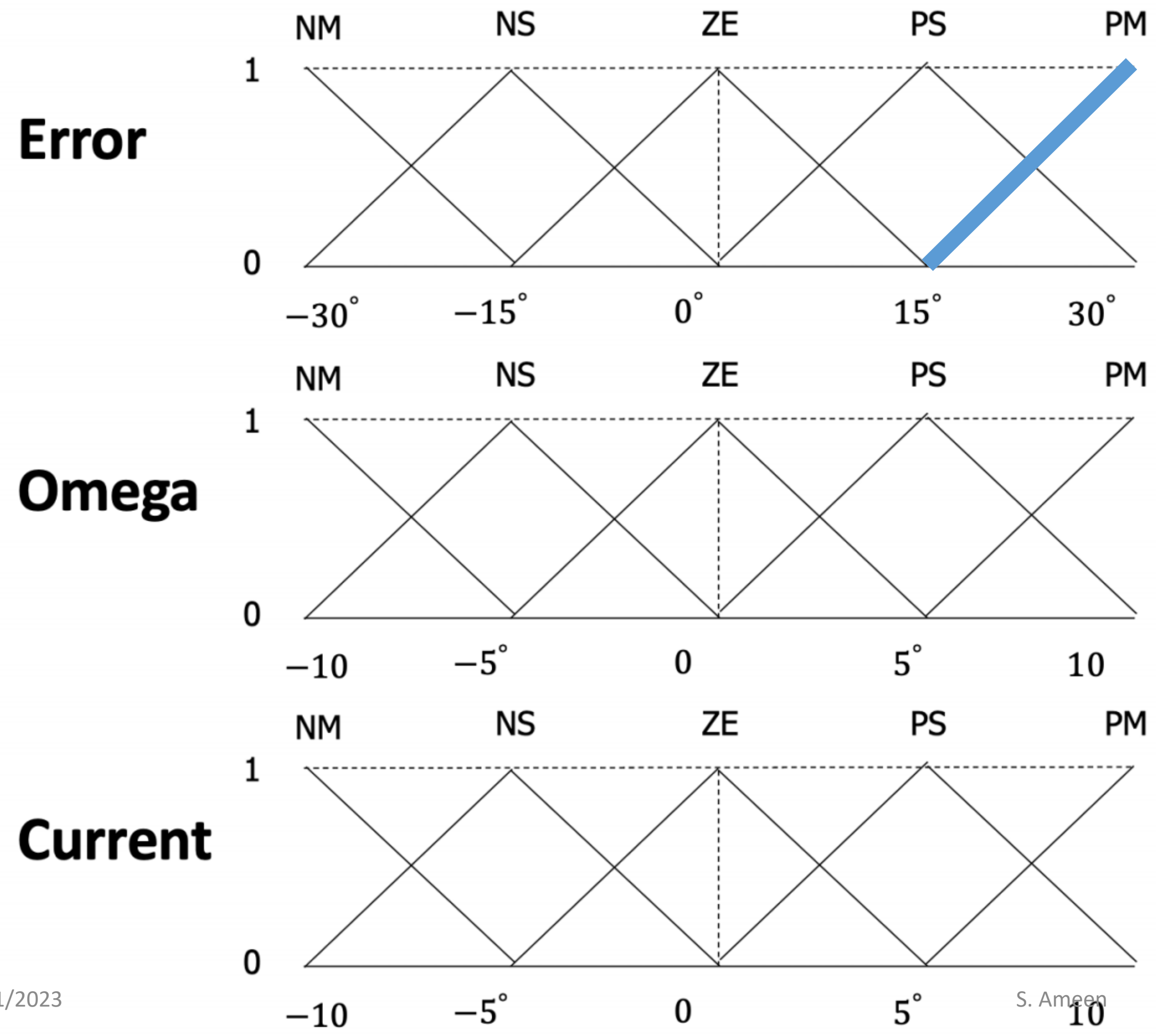
Total Number of Rules = 5^3

Error = 27 degree

Omega = -1.5 degree/Second

Current = ?

Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Defuzzifier



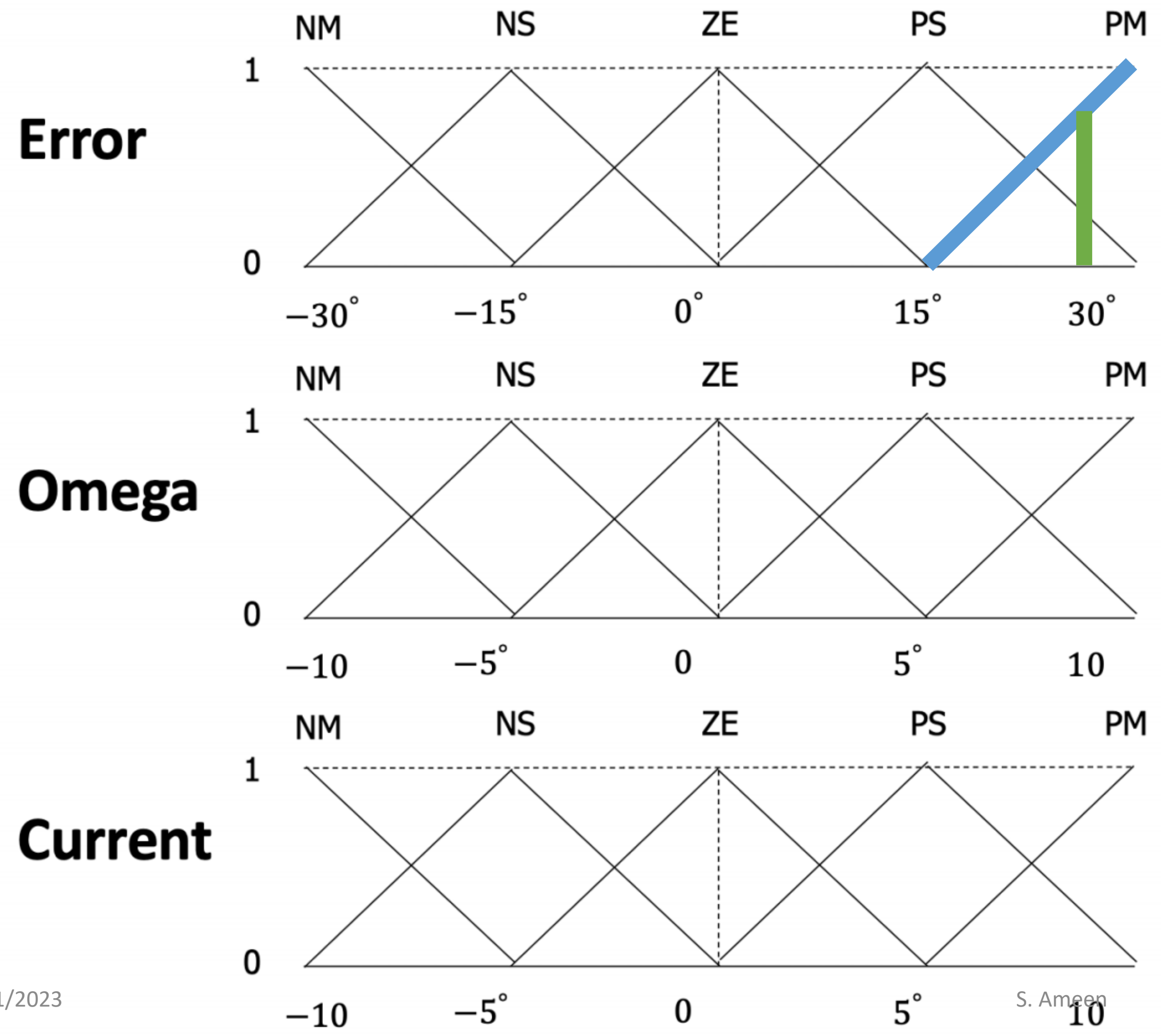
If error is PM and Omega is ZE then Current is NM

Error = 27 degree

Omega = -1.5 degree/Second

Current = ?

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



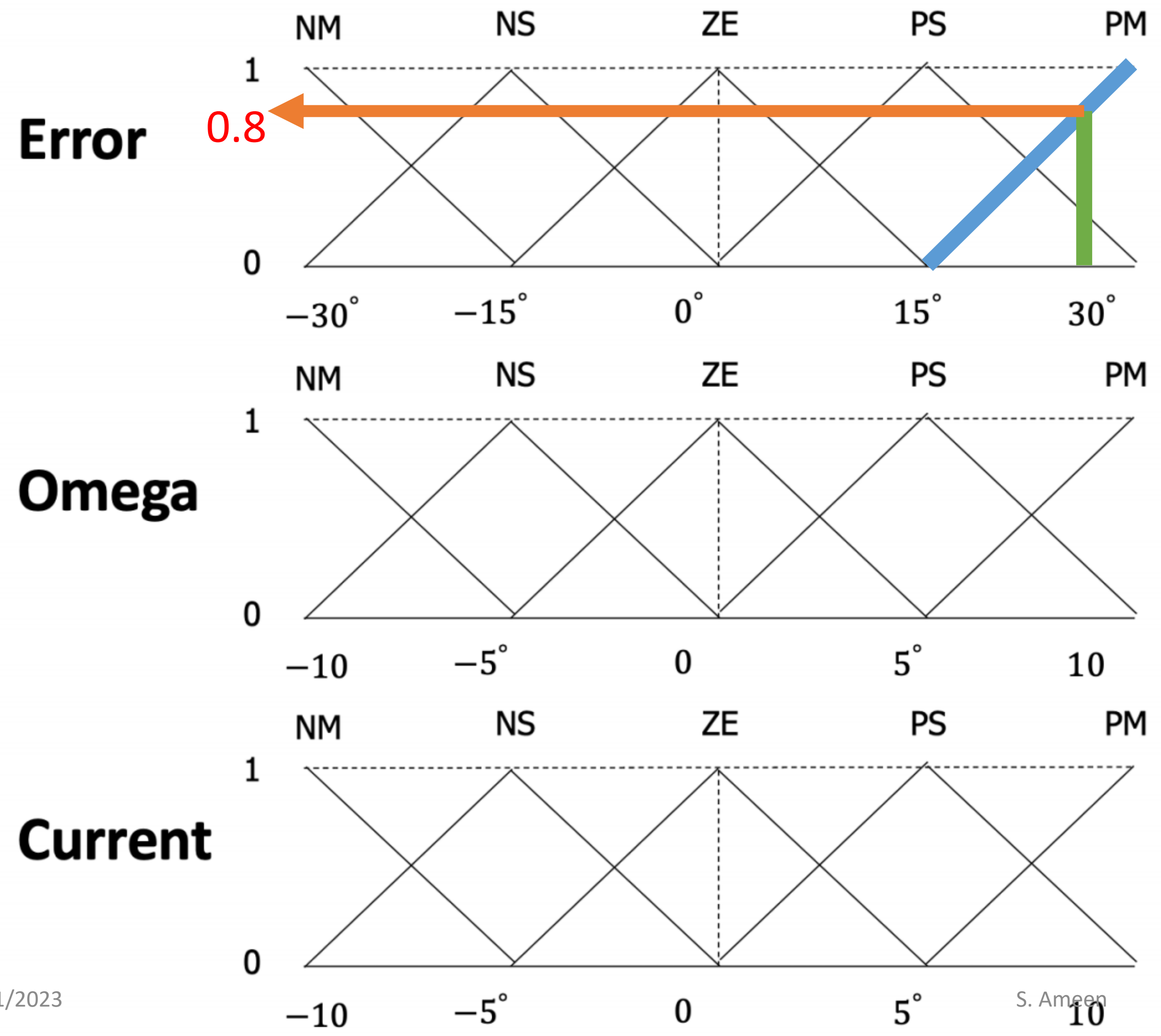
If error is PM and Omega is ZE then Current is NM

Error = 27 degree

Omega = -1.5 degree/Second

Current = ?

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



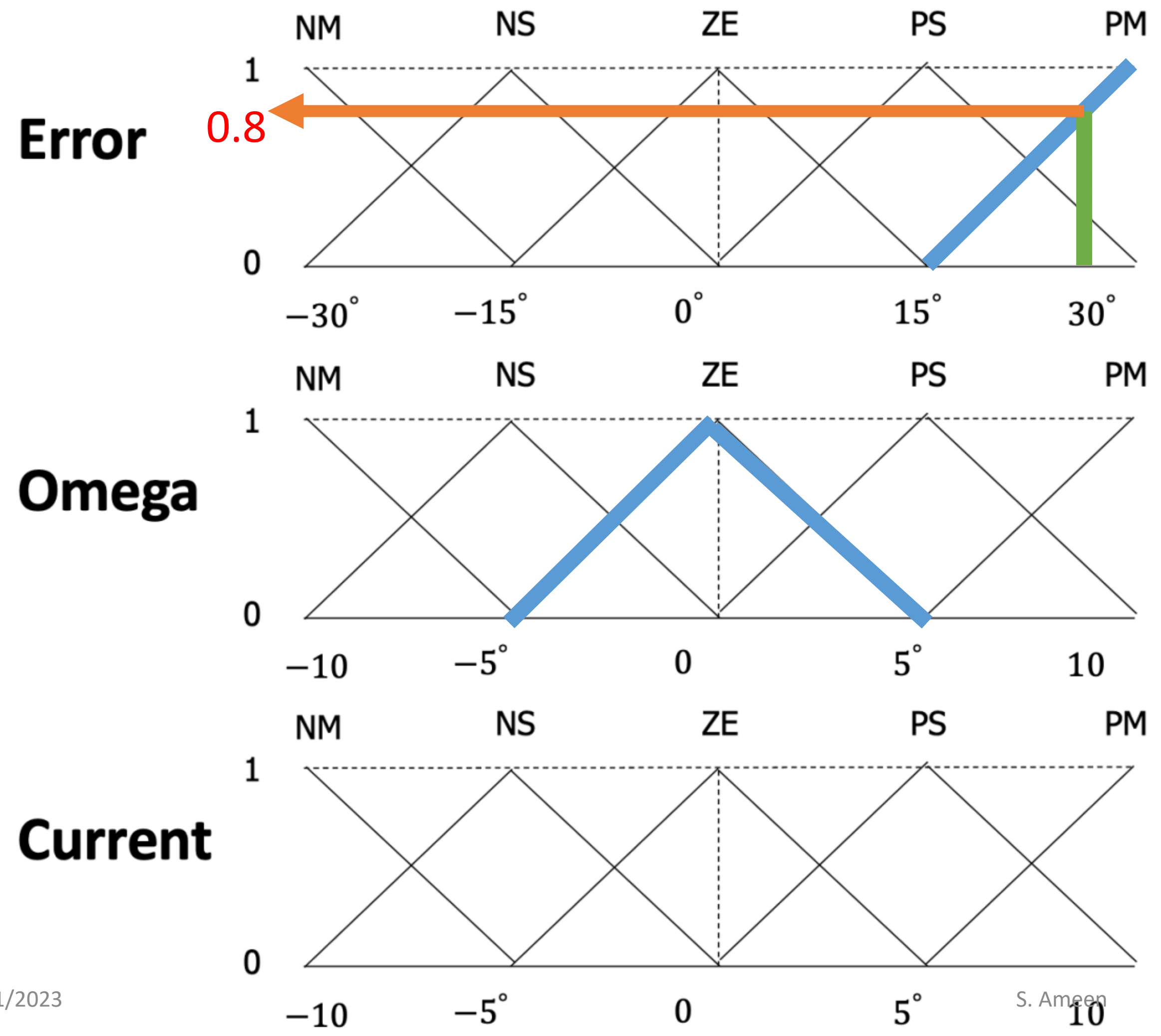
If error is PM and Omega is ZE then Current is NM

Error = 27 degree ➔ 0.8

Omega = -1.5 degree/Second

Current = ?

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



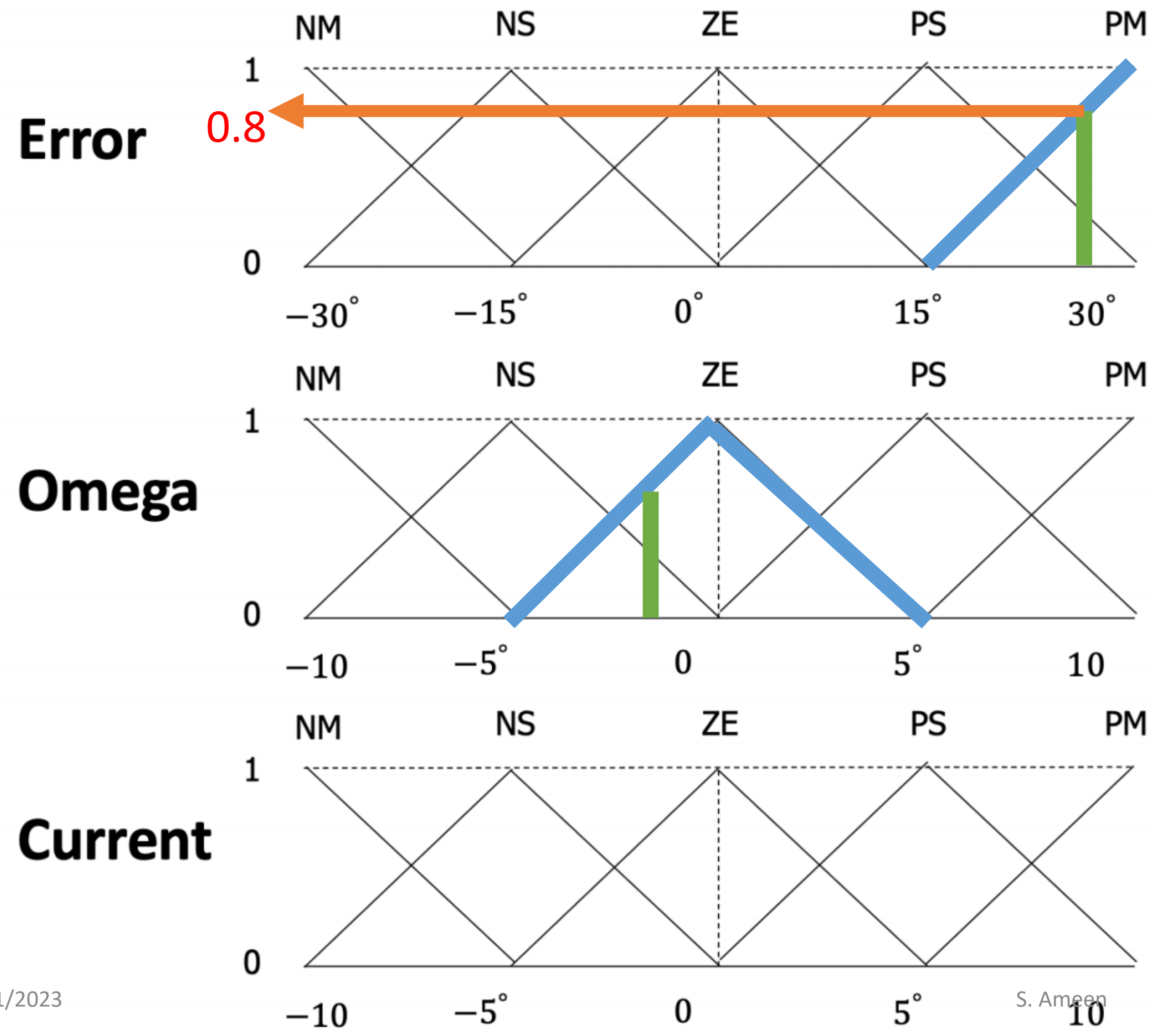
If error is PM and Omega is ZE then Current is NM

Error = 27 degree ➔ 0.8

Omega = -1.5 degree/Second

Current = ?

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



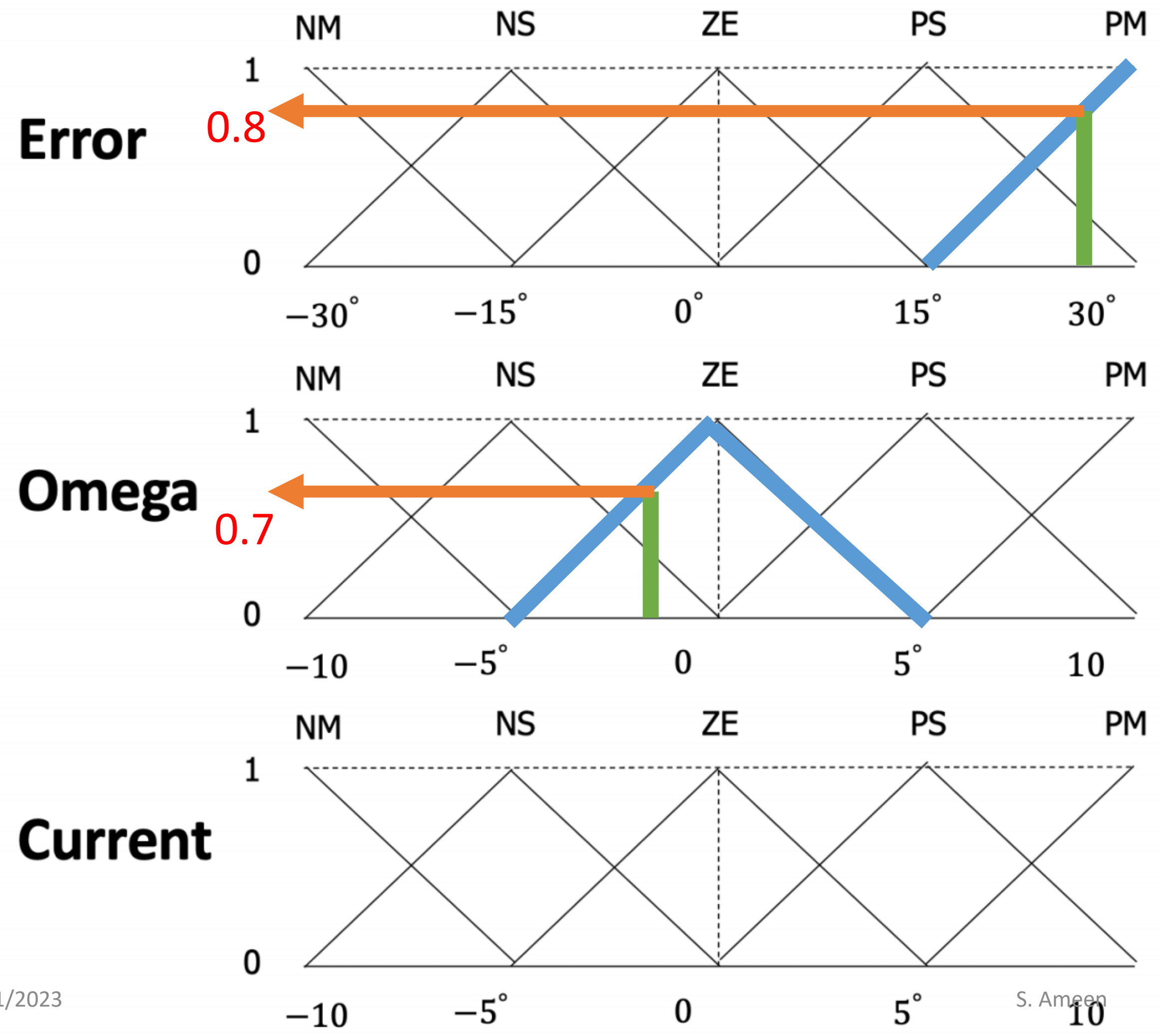
If error is PM and Omega is ZE then Current is NM

Error = 27 degree → 0.8

Omega = -1.5 degree/Second

Current = ?

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



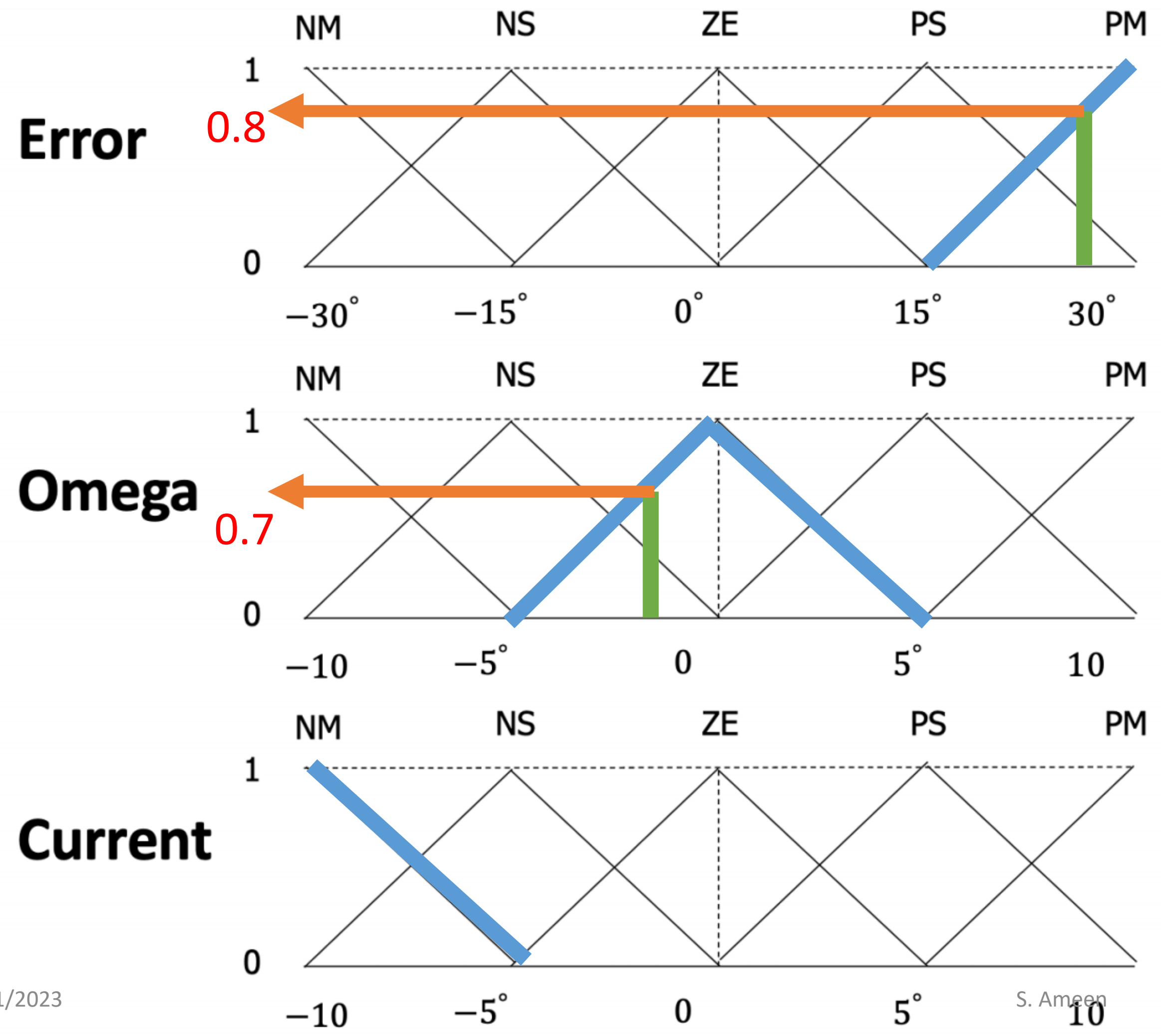
If error is PM and Omega is ZE then Current is NM

Error = 27 degree ➔ 0.8

Omega = -1.5 D/S ➔ **0.7**

Current = ?

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



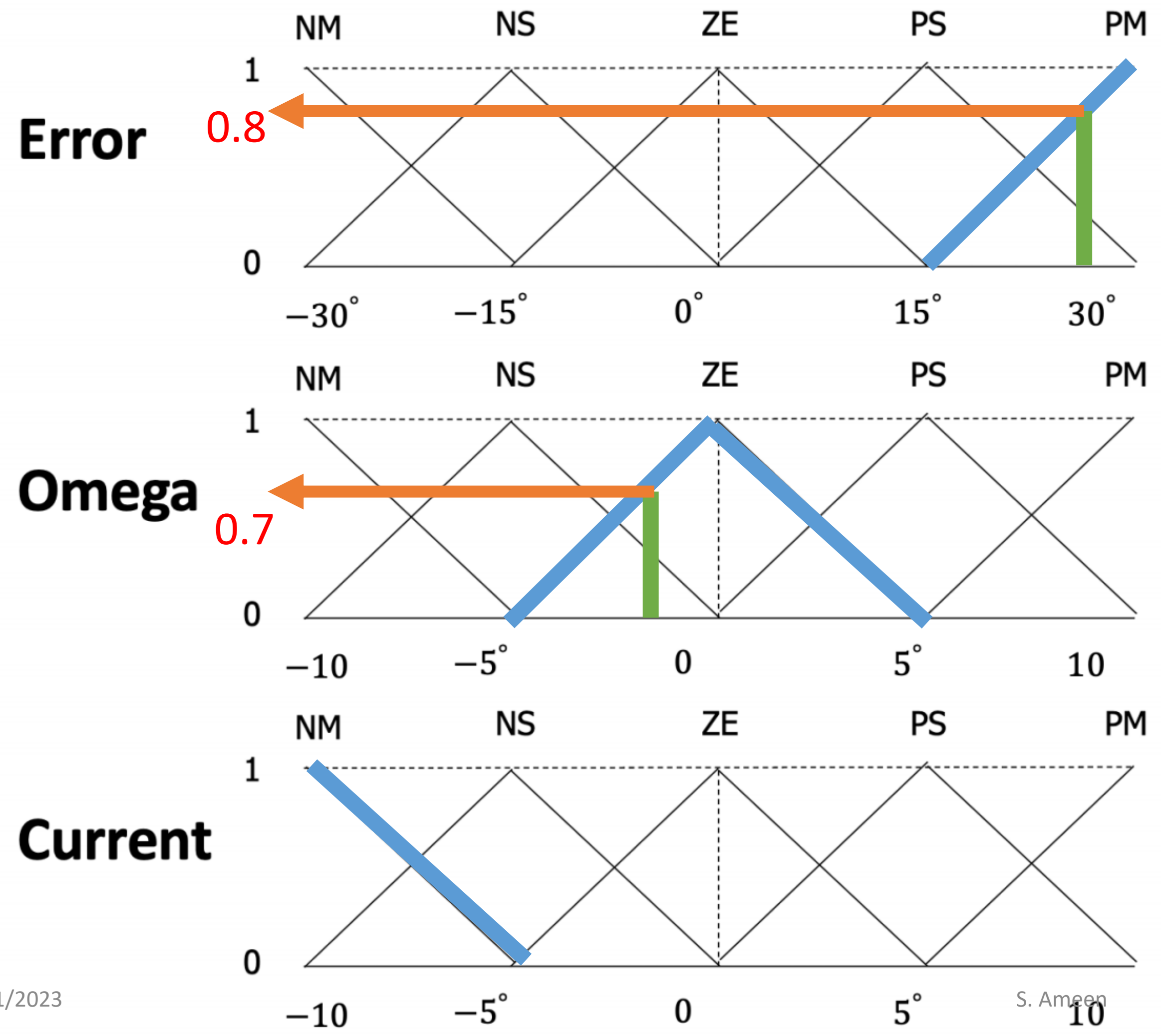
If error is PM and Omega is ZE then Current is NM

Error = 27 degree ➔ 0.8

Omega = -1.5 D/S ➔ **0.7**

Current = ?

Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Defuzzifier



If error is PM and
Omega is ZE then
Current is NM

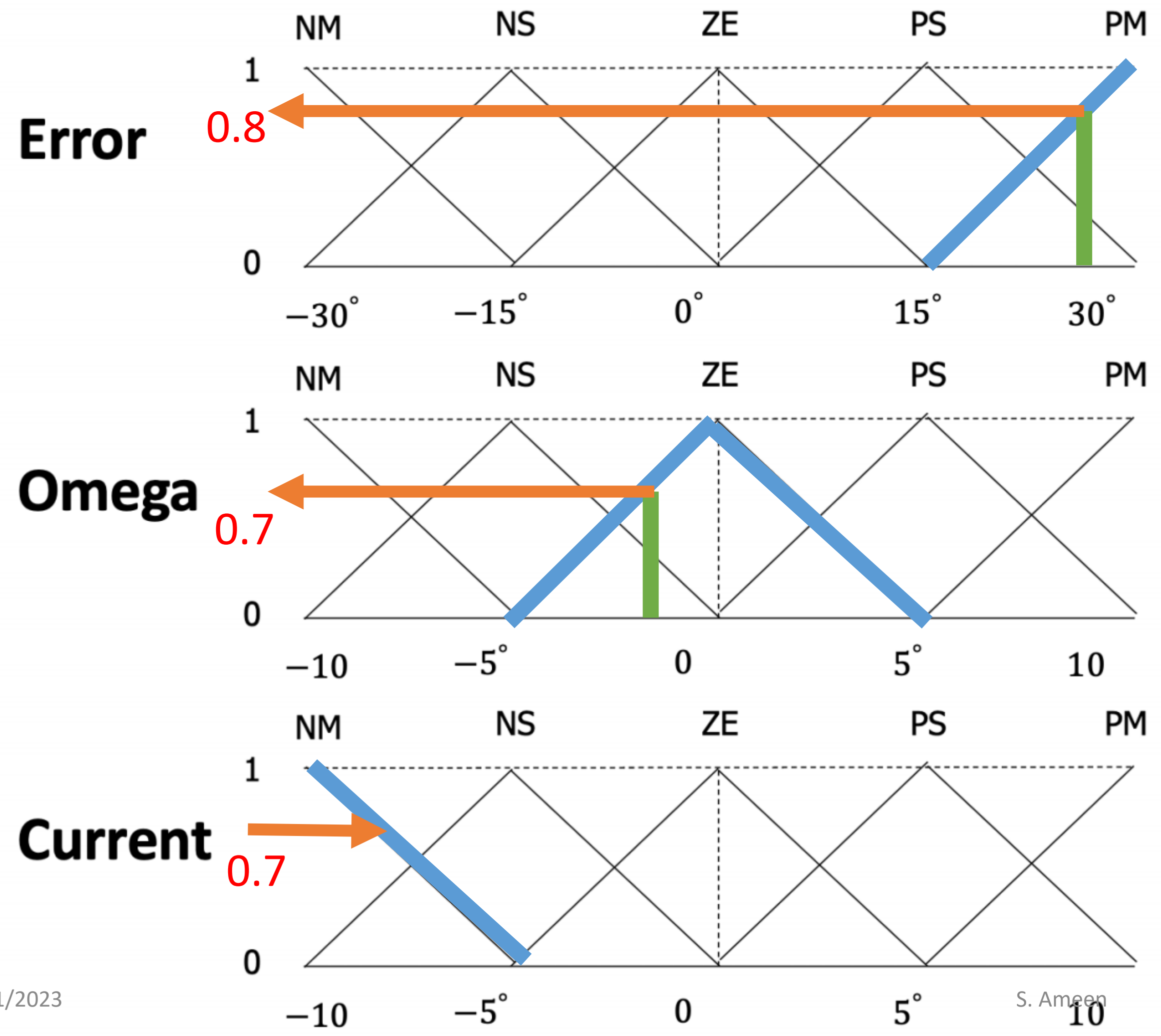
Error = 27 degree ➔ 0.8

Omega = -1.5 D/S ➔ **0.7**

Current = ?

AND = min(0.8,0.7) = 0.7

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier



If error is PM and Omega is ZE then Current is NM

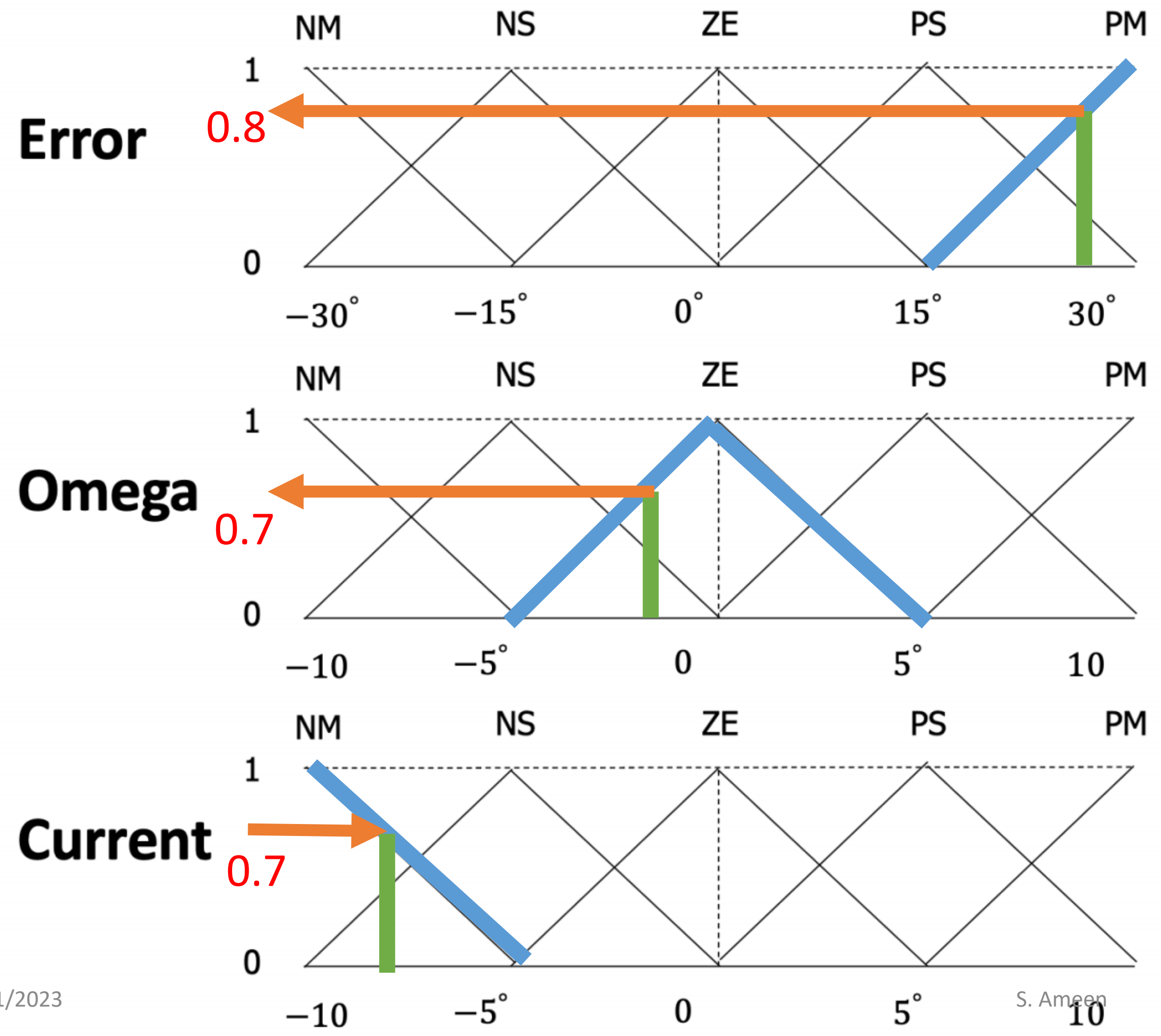
Error = 27 degree ➔ 0.8

Omega = -1.5 D/S ➔ **0.7**

Current = ?

AND = min(0.8,0.7) = 0.7

Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Defuzzifier



If error is PM and Omega is ZE then Current is NM

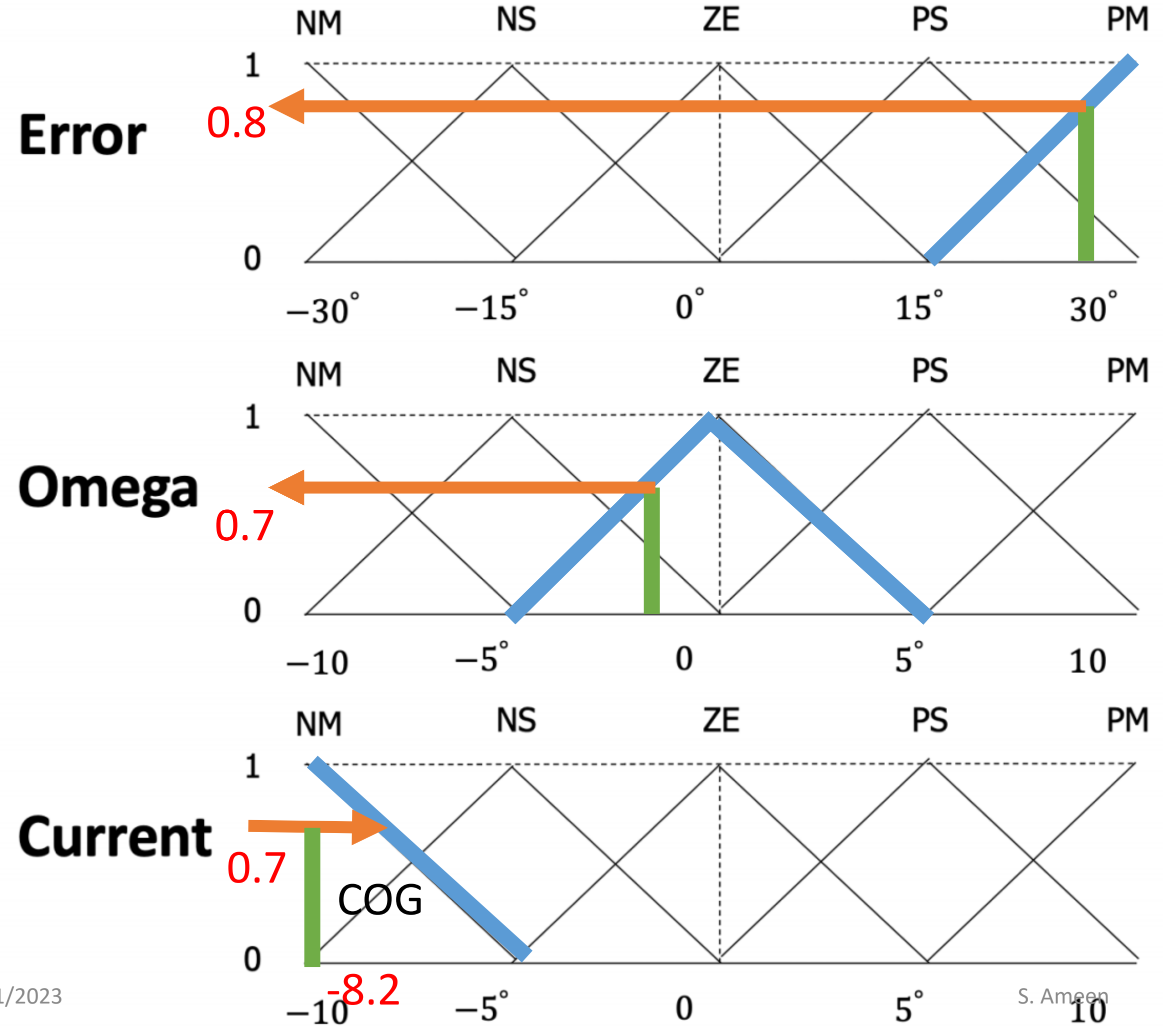
Error = 27 degree → 0.8

Omega = -1.5 D/S → **0.7**

Current = ?

AND = min(0.8,0.7) = 0.7

Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Defuzzifier



If error is PM and Omega is ZE then Current is NM

Error = 27 degree ➔ 0.8

Omega = -1.5 D/S ➔ **0.7**

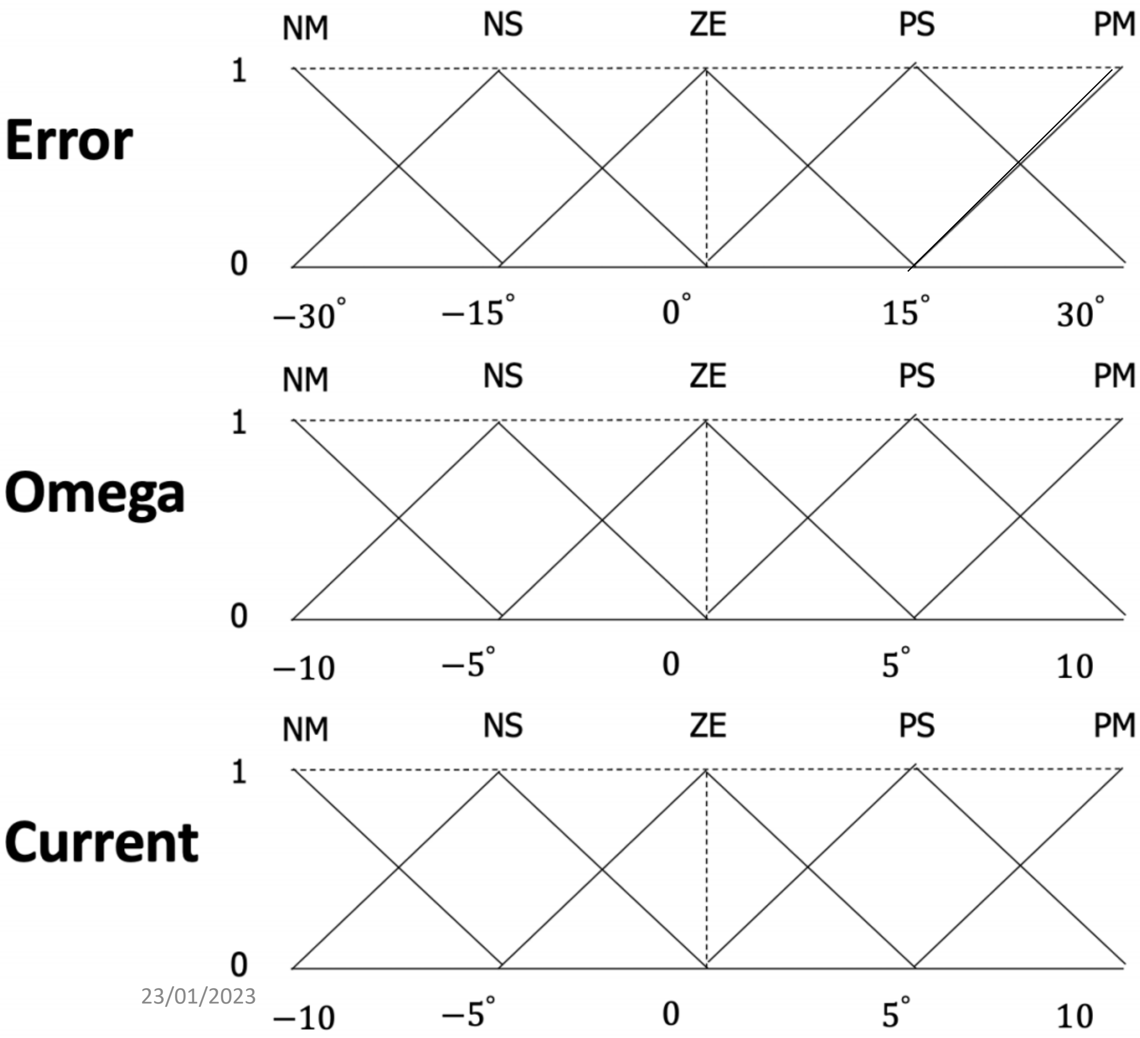
Current = -8.2

AND = min(0.8,0.7) = 0.7

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Coding

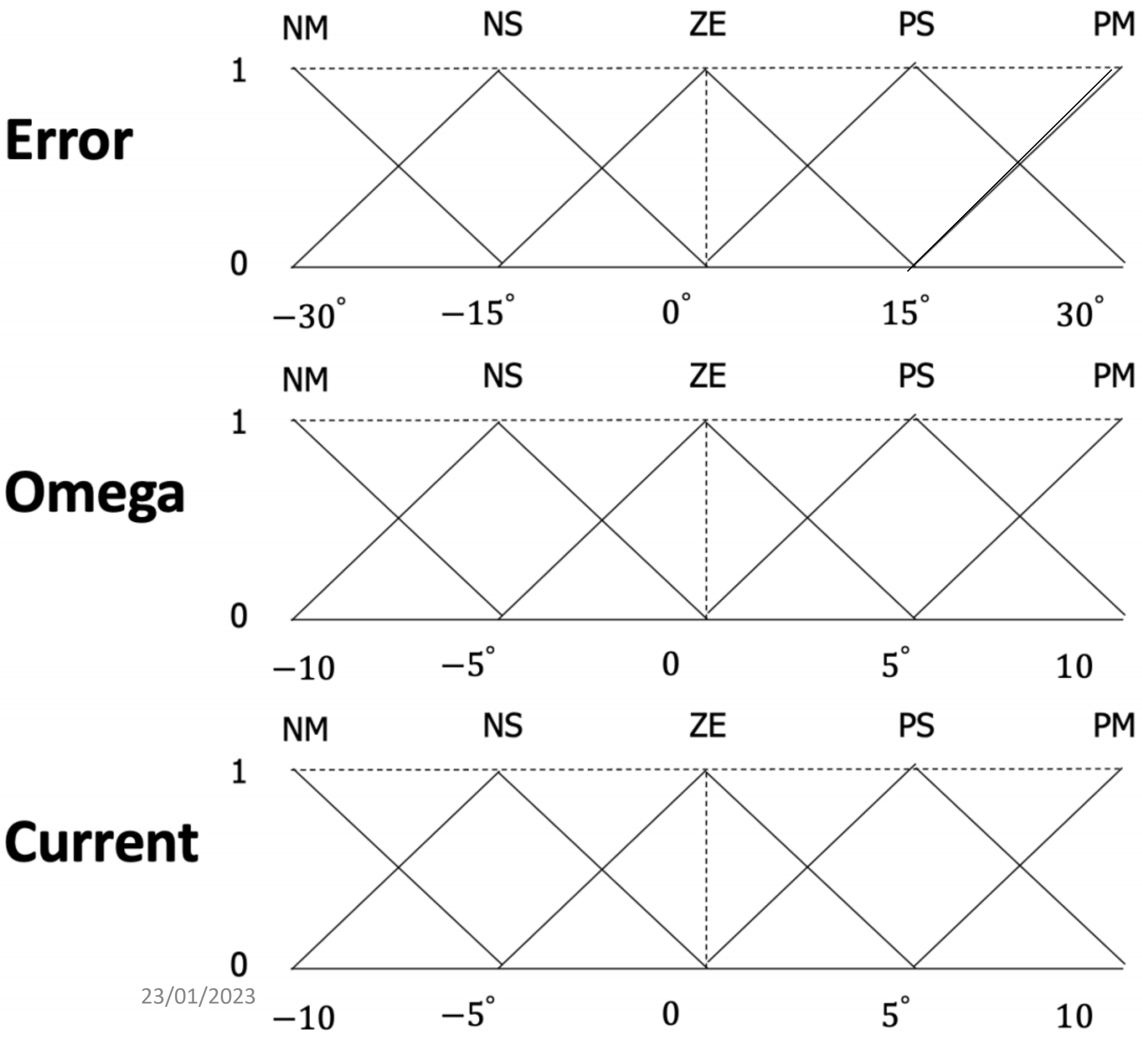
```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
%matplotlib inline
import matplotlib.pyplot as plt
```

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Coding



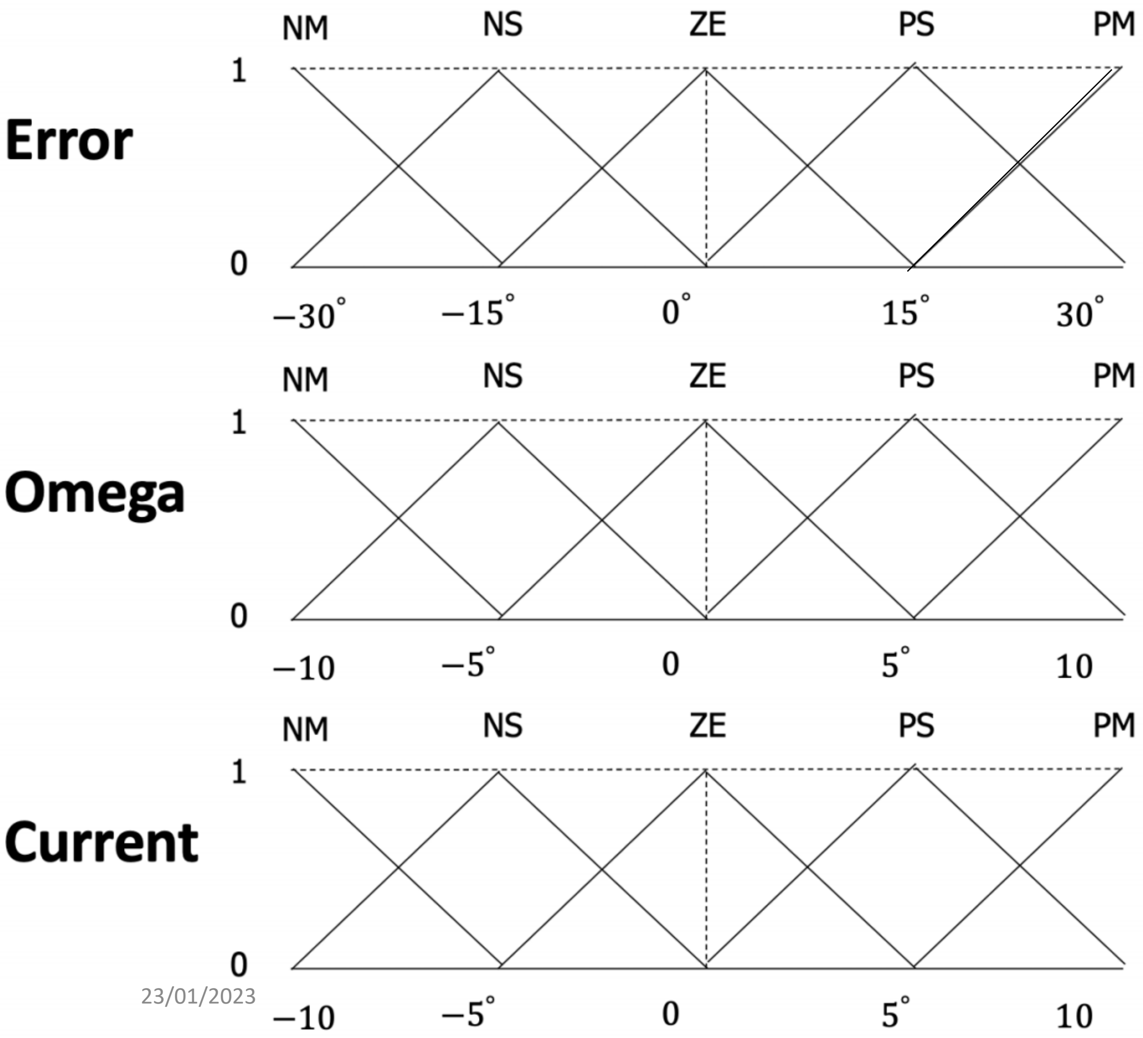

```
error = ctrl.Antecedent(np.arange(-30, 31, 1), 'error')
omega = ctrl.Antecedent(np.arange(-10, 11, 1), 'omega')
current = ctrl.Consequent(np.arange(-10, 11, 1), 'current')
```

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Coding - Inputs



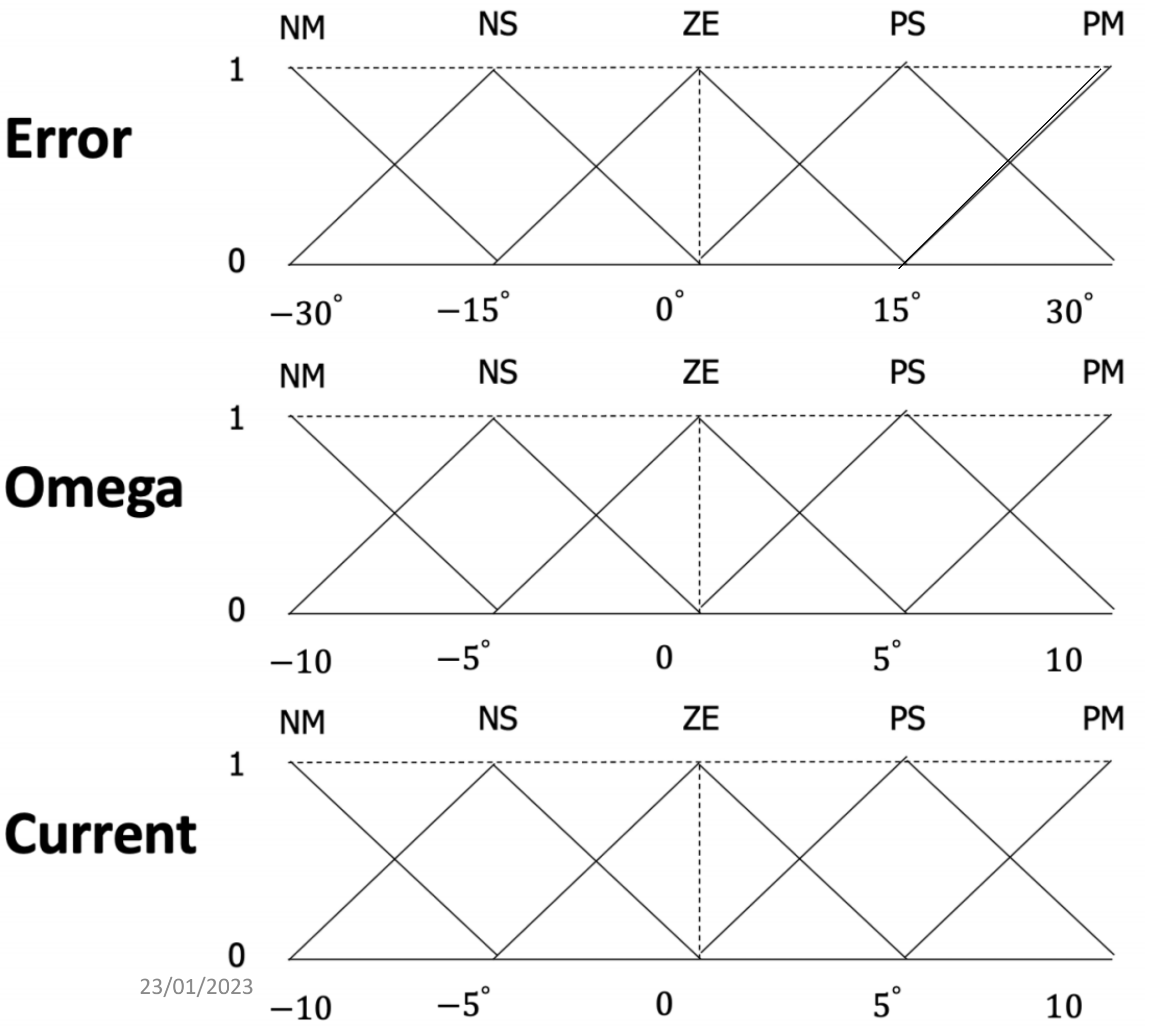

```python
names = ['NM', 'NS', 'ZE', 'PS', 'PM']
error.automf(names=names)
omega.automf(names=names)
```

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Coding - Outputs




```python
current['NM'] = fuzz.trimf(current.universe, [-10, -10, -5])
current['NS'] = fuzz.trimf(current.universe, [-10, -5, 0])
current['ZE'] = fuzz.trimf(current.universe, [-5, 0, 5])
current['PS'] = fuzz.trimf(current.universe, [0, 5, 10])
current['PM'] = fuzz.trimf(current.universe, [5, 10, 10])
```

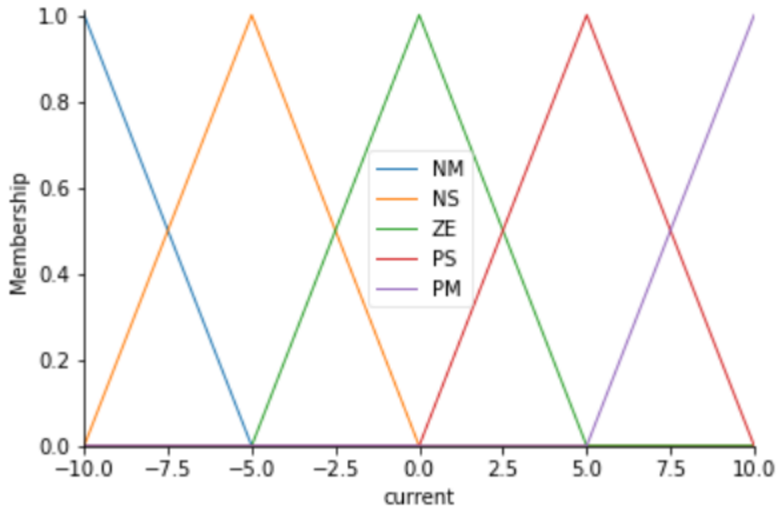
Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Coding - Outputs




```
current['NM'] = fuzz.trimf(current.universe, [-10, -10, -5])
current['NS'] = fuzz.trimf(current.universe, [-10, -5, 0])
current['ZE'] = fuzz.trimf(current.universe, [-5, 0, 5])
current['PS'] = fuzz.trimf(current.universe, [0, 5, 10])
current['PM'] = fuzz.trimf(current.universe, [5, 10, 10])
```

```
# You can see how these look with .view()
current.view()
```

```
C:\Users\Elise\anaconda3\lib\site-packages\skfuzzy\control\fuzzyv
ackend_inline, which is a non-GUI backend, so cannot show the fig
  fig.show()
```



Lecture- Part B: Fuzzy Logic Inverted Pendulum / Inference Engine Coding

Error	Omega	Current
ZE	ZE	ZE
ZE	NM	PM
ZE	PS	NS
PM	**ZE**	**NM**

```
rule1 = ctrl.Rule(error['PM'] & omega['ZE'], current['NM'])

rule1.view()

rule2 = ctrl.Rule(error['NM'] & omega['ZE'], current['PM'])

rule2.view()

rule3 = ctrl.Rule(error['ZE'] & omega['ZE'], current['ZE'])

rule3.view()
```

Lecture- Part B: Fuzzy Logic Inverted Pendulum / Inference Engine Coding

Error	Omega	Current
ZE	ZE	ZE
ZE	NM	PM
ZE	PS	NS
PM	**ZE**	**NM**

```
rule1 = ctrl.Rule(error['PM'] & omega['ZE'], current['NM'])

rule1.view()

rule2 = ctrl.Rule(error['NM'] & omega['ZE'], current['PM'])

rule2.view()

rule3 = ctrl.Rule(error['ZE'] & omega['ZE'], current['ZE'])

rule3.view()
```

```
pendulum_ctrl = ctrl.ControlSystem([rule1,rule2,rule3])
pendulum = ctrl.ControlSystemSimulation(pendulum_ctrl)
```

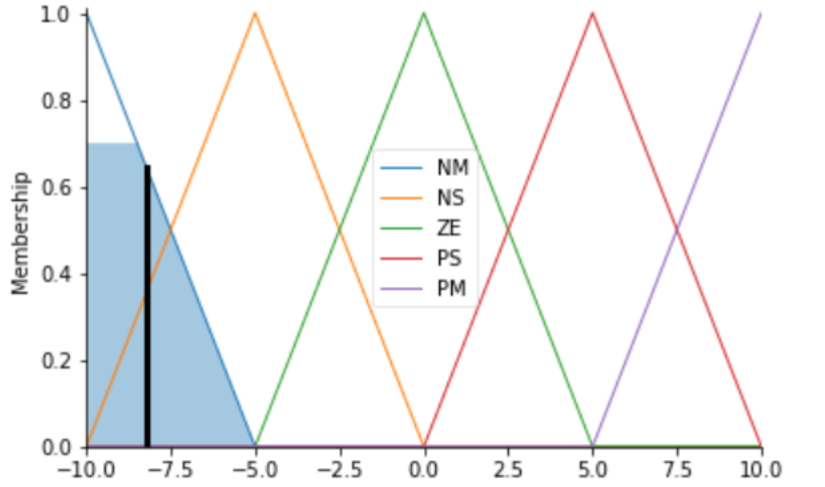
Lecture- Part B: Fuzzy Logic Inverted Pendulum / Defuzzifier

```
pendulum.input['error'] = 27
pendulum.input['omega'] = -1.5

# Crunch the numbers
pendulum.compute()
print(pendulum.output['current'])
current.view(sim=pendulum)
```

```
-8.217948717948717
```

```
C:\Users\Elise\anaconda3\lib\site-packages\skfuzzy\contro
ackend_inline, which is a non-GUI backend, so cannot show
  fig.show()
```



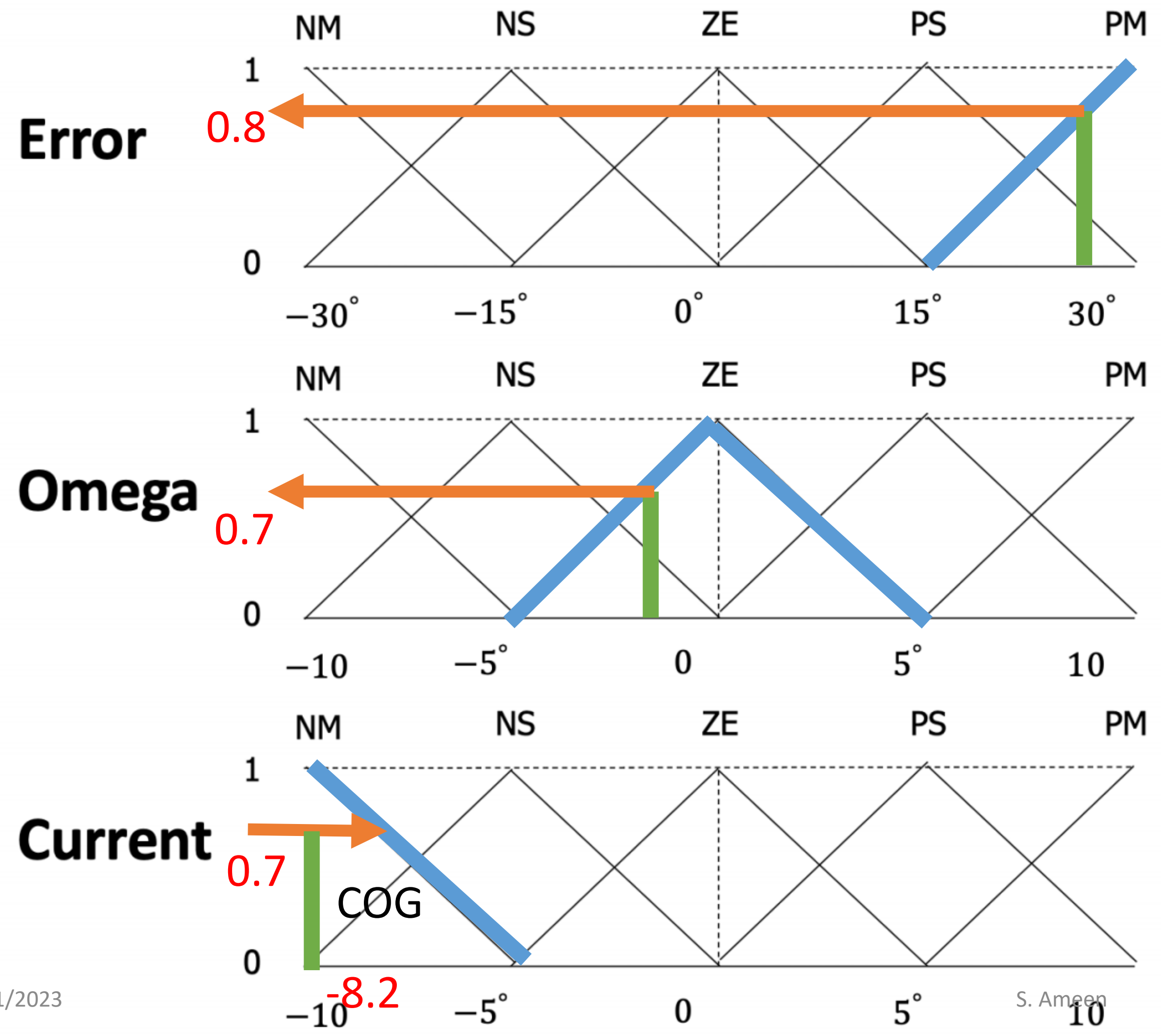
If error is PM and Omega is ZE then Current is NM

Error = 27 degree ➔ 0.8

Omega = -1.5 D/S ➔ **0.7**

Current = -8.2

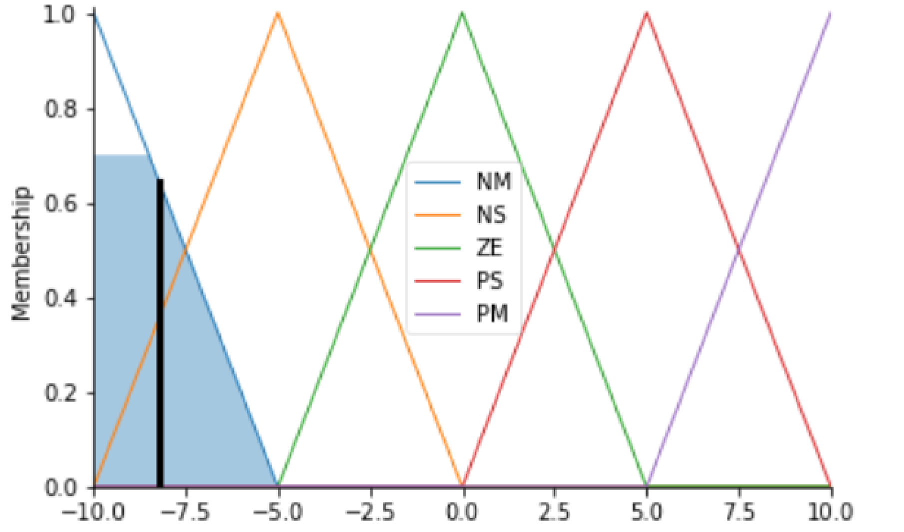
Lecture- Part B: Fuzzy Logic
Inverted Pendulum / Defuzzifier




```
pendulum.input['error'] = 27
pendulum.input['omega'] = -1.5

# Crunch the numbers
pendulum.compute()
print(pendulum.output['current'])
current.view(sim=pendulum)
```

```
-8.217948717948717
C:\Users\Elise\anaconda3\lib\site-packages\skfuzzy\contro
ackend_inline, which is a non-GUI backend, so cannot show
  fig.show()
```



Lecture- Part B: Fuzzy Logic FLC/ Applications – Master Project

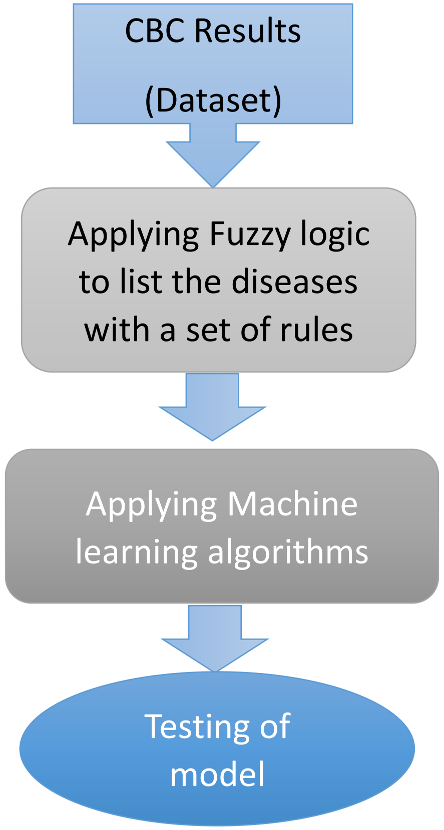
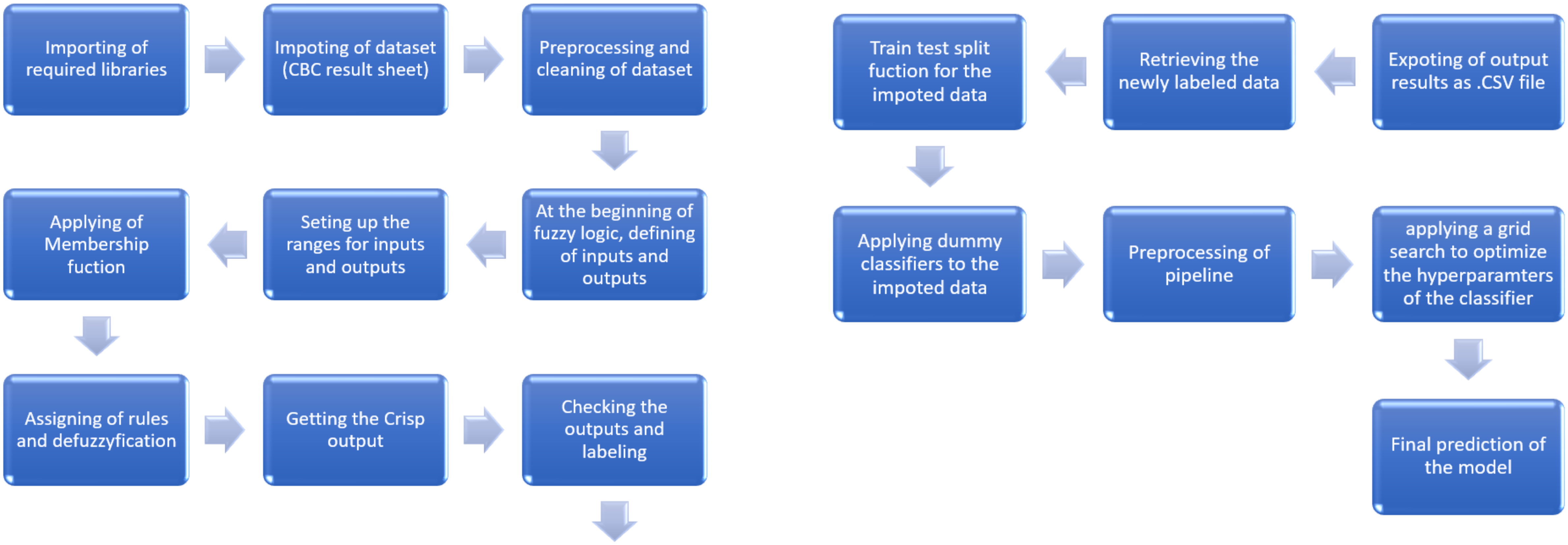


Figure 5 : Disease prediction model architecture

Lecture- Part B: Fuzzy Logic FLC/ Applications – Master Project



Lecture- Part B: Fuzzy Logic FLC/ Applications – Master Project

Specified rules for fuzzy logic

All rules were created based on previous medical histories by a Medical Laboratory Technologist with a help of doctor.

Rule 1= Septicaemia/Urine Tract Infections (UTI) are detected based on this values level that is WBC should be high, HGB should be normal, HCT should be normal, PLT should be normal.

Rule 2=Septicaemia disease is detected by the following values levels that is WBC is high, HGB is normal. HCT is normal, and PLT is high.

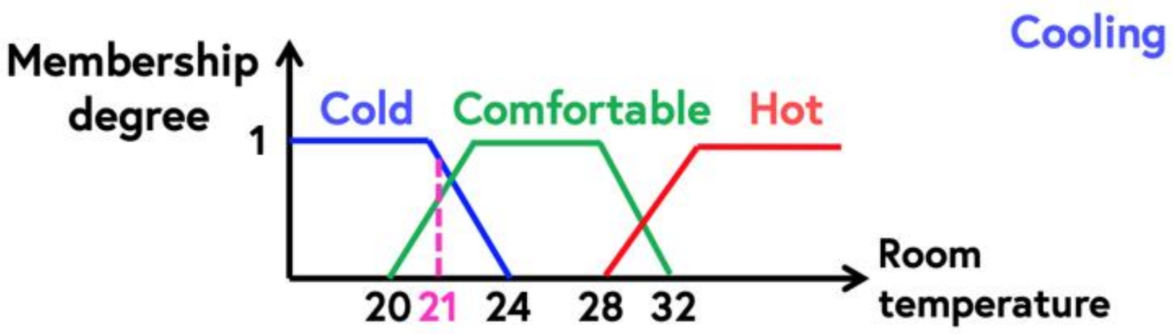
Rule 3 =If WBC is normal, HGB is high, HCT is high and PLT is low hence these values show that Dengue disease has been detected.

Rules 4=If WBC is low, HGB is high, HCT is high and PLT is low hence these values show that Dengue disease has been detected.

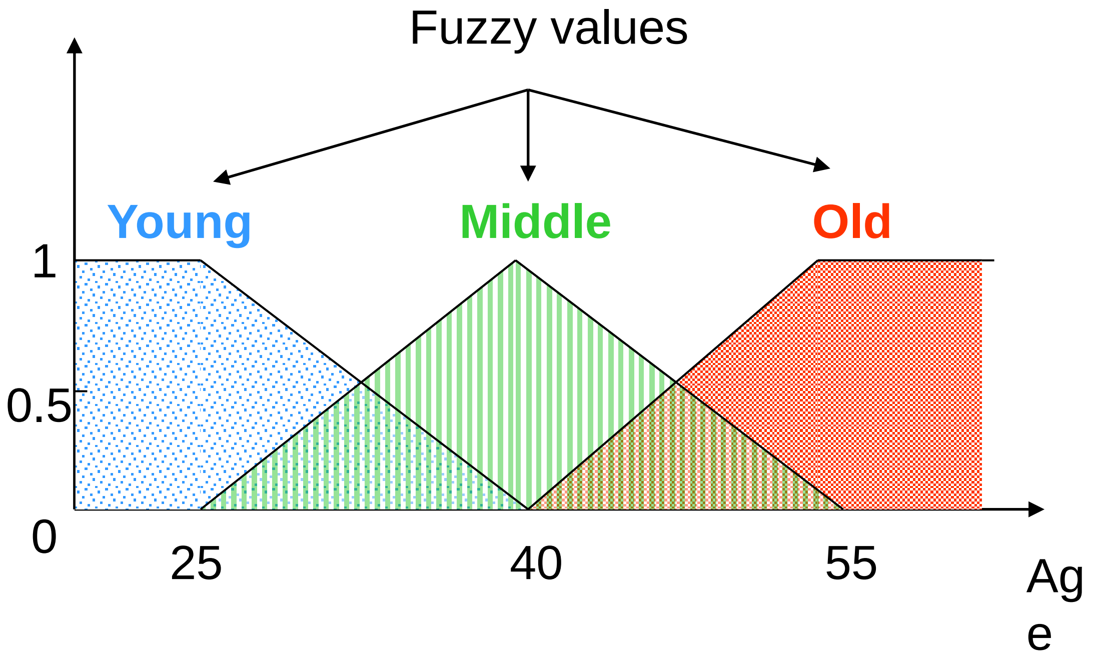
Rule 5=If WBC is low, HGB is normal, HCT is normal and PLT is low that value states the other viral fevers have been detected.

Rule 6=If WBC is normal, HGB is normal, HCT is normal, and PLT is low the values state the other viral fevers and Idiopathic Thrombocytopenic purpura (ITP) have been detected.

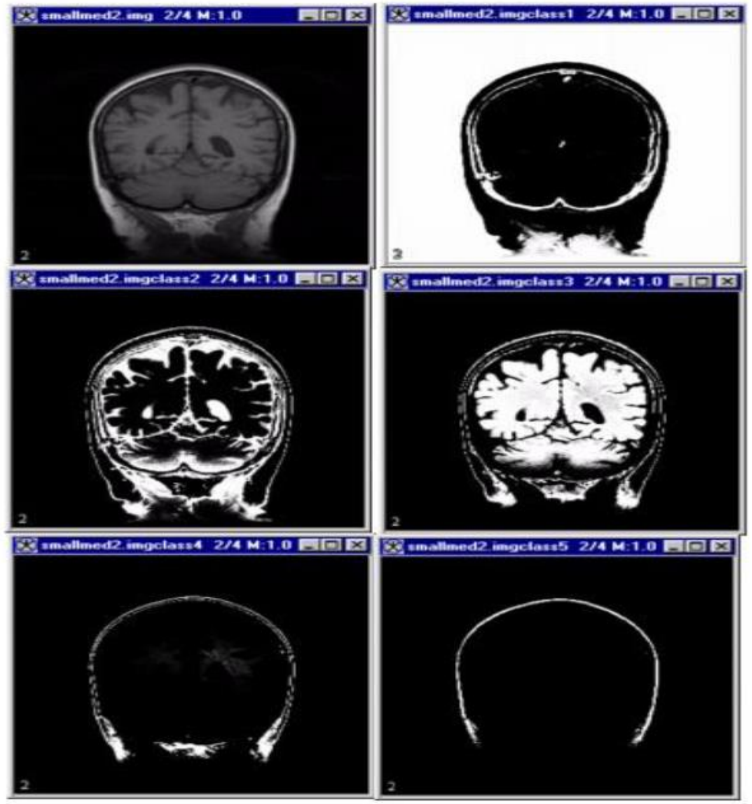
Lecture- Part B: Fuzzy Logic FLC/ Applications



Temp= fuzz.trapmf(x_house_market, [50, 100, 200, 250])

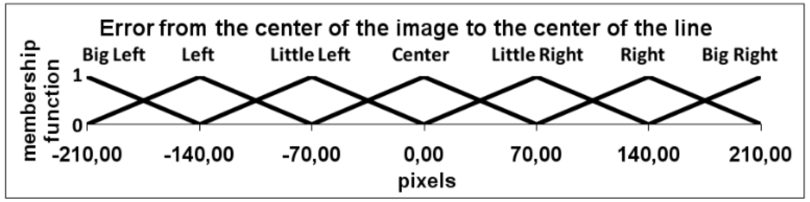


Lecture- Part B: Fuzzy Logic FLC/ Applications

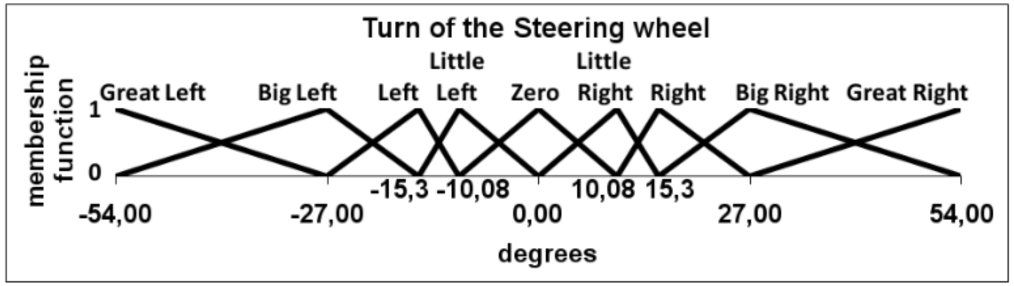


https://mipav.cit.nih.gov/pubwiki/index.php/Fuzzy_C-Means:_Multispectral_and_Single_Channel_Algorithms

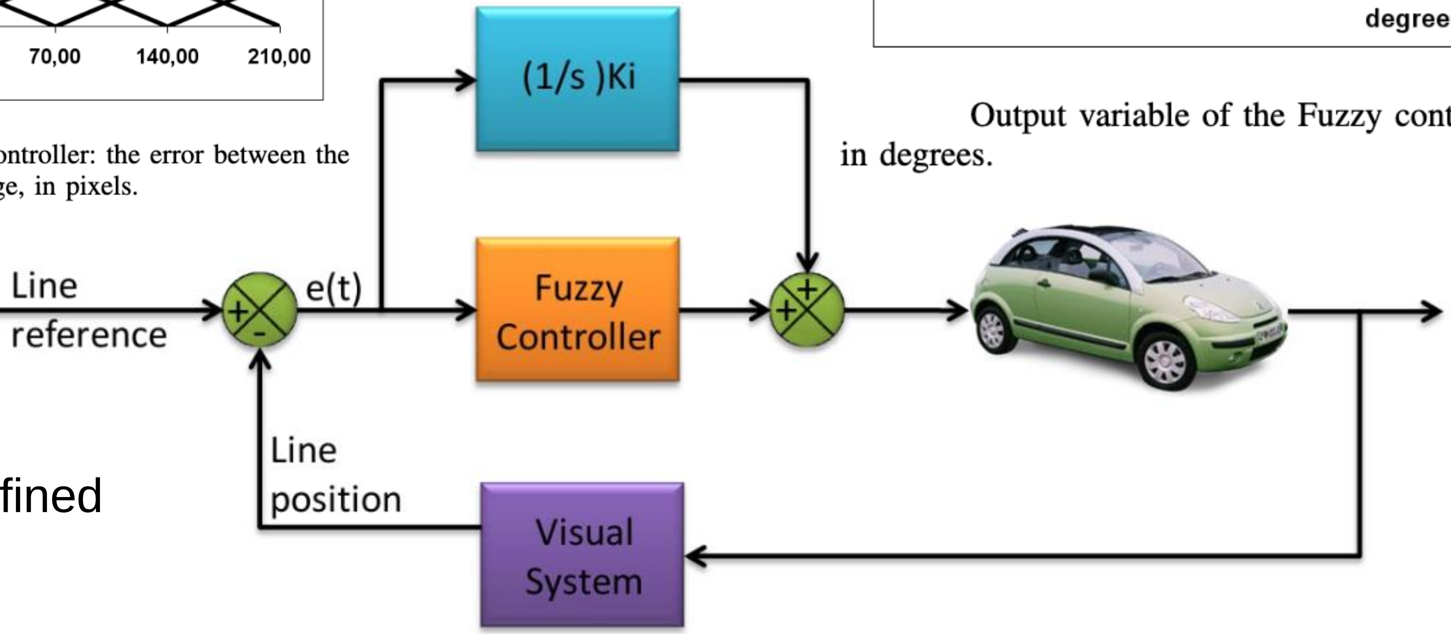
Lecture- Part B: Fuzzy Logic FLC/ Applications



First input variable of the Fuzzy controller: the error between the centre of the line and the centre of the image, in pixels.



Output variable of the Fuzzy controller: the steering wheel angle, in degrees.

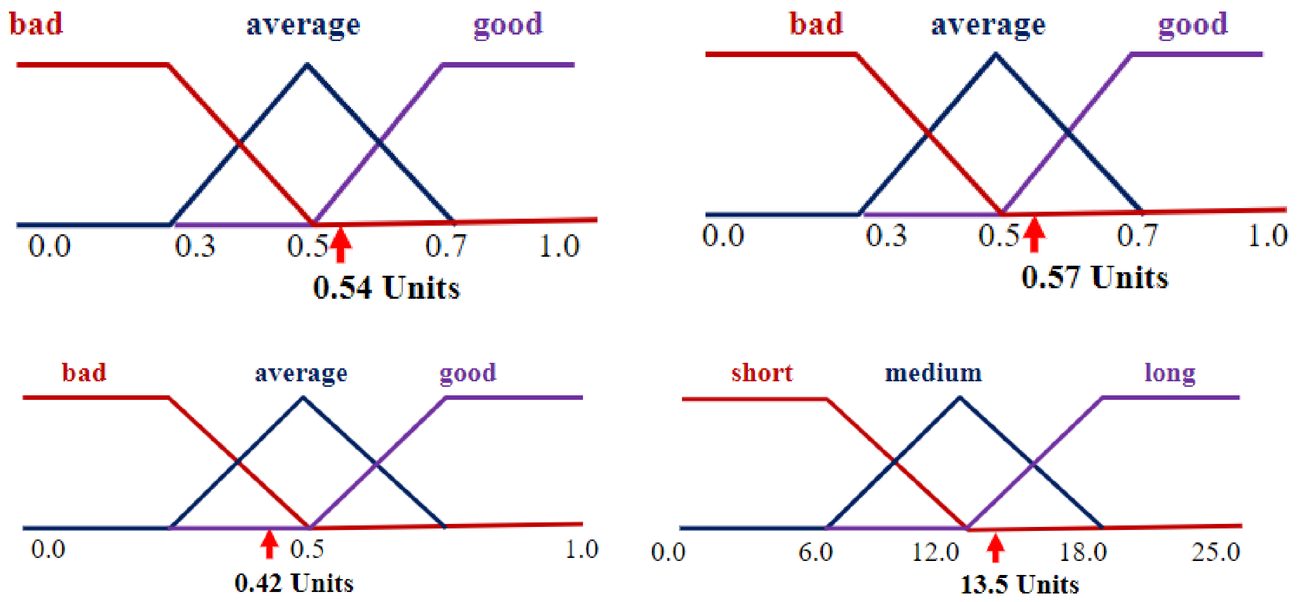


To obtain this output,
49 if-then rules were defined

M. A. Olivares-Mendez, I. Mellado, P. Campoy, I. Mondragon and C. Martinez, "A visual AGV-urban car using Fuzzy control," *The 5th International Conference on Automation, Robotics and Applications*, Wellington, **2011**, pp. 145-150,

Lecture- Part B: Fuzzy Logic FLC/ Applications

https://www.sciencedirect.com/science/article/pii/S1877050916326138



Lecture- Part B: Fuzzy Logic Summary

- Background
- Fuzzy Set Theory
- Fuzzy Logic Controller
- Applications

Lecture- Part A: Logical Agents Next Week:

Uncertainty