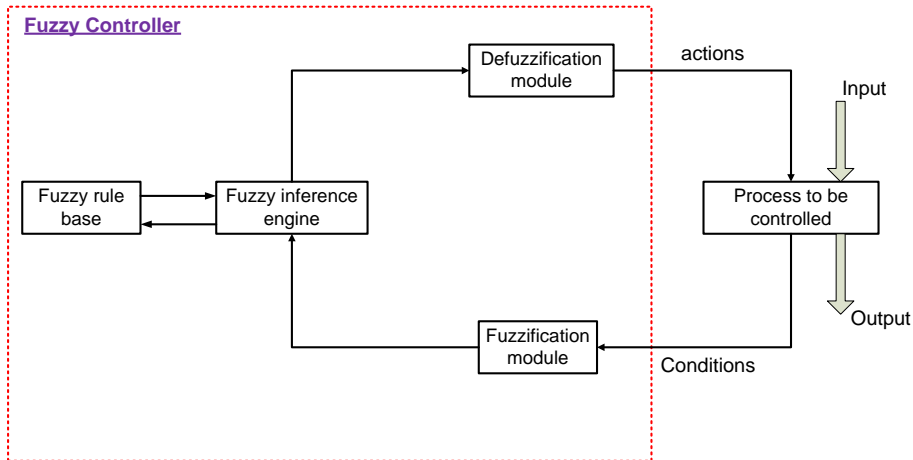# Fuzzy Logic Controller

# Applications of Fuzzy Logic

## Fuzzy Systems : Fuzzy Logic Controller

- Concept of fuzzy theory can be applied in many applications, such as fuzzy reasoning, fuzzy clustering, fuzzy programming etc.

- Out of all these applications, fuzzy reasoning, also called "fuzzy logic controller (FLC)" is an important application.

- Fuzzy logic controllers are special expert systems. In general, a FLC employs a knowledge base expressed in terms of a fuzzy inference rules and a fuzzy inference engine to solve a problem.

- We use FLC where an exact mathematical formulation of the problem is not possible or very difficult.

- These difficulties are due to non-linearities, time-varying nature of the process, large unpredictable environment disturbances etc.

# Fuzzy Systems : Fuzzy Logic Controller

A general scheme of a fuzzy controller is shown in the following figure.

# Fuzzy Systems : Fuzzy Logic Controller

A general fuzzy controller consists of four modules:

1. a fuzzy rule base,

2. a fuzzy inference engine,

3. a fuzzification module, and

4. a defuzzification module.

## Fuzzy Systems : Fuzzy Logic Controller

As shown in Figure 1, a fuzzy controller operates by repeating a cycle of the following four steps :

1. Measurements (inputs) are taken of all variables that represent relevant condition of controller process.

2. These measurements are converted into appropriate fuzzy sets to express measurements uncertainties. This step is called fuzzification.

3. The fuzzified measurements are then used by the inference engine to evaluate the control rules stroed in the fuzzy rule base. The result of this evaluation is a fuzzy set (or several fuzzy sets) defined on the universe of possible actions.

4. This output fuzzy set is then converted into a single (crisp) value (or a vector of values). This is the final step called defuzzification. The defuzzified values represent actions to be taken by the fuzzy contoller.

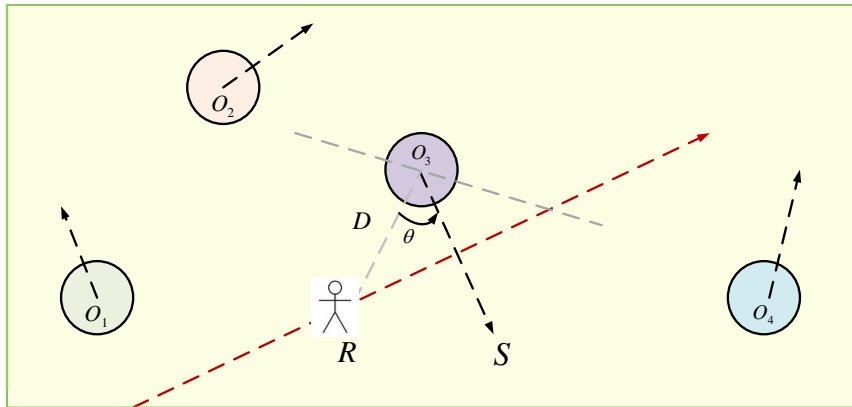# Fuzzy Systems : Fuzzy Logic Controller

There are two approaches of FLC known.

1. Mamdani approach

2. Takagi and sugeno's approach

- Mamdani approach follows linguistic fuzzy modeling and chacterized by its high interpretability and low accuracy.
- On the other hand, Takagi and Sugeno's approach follows precise fuzzy modeling and obtains high accuracy but at the cost of low interpretabily.
- We illustrate the above two approaches with two examples.

- Consider the control of navigation of a mobile robot in the pressure of a number of moving objects.

- To make the problem simple, consider only four moving objects, each of equal size and moving with the same speed.

- A typical scenario is shown in Figure 2.

## Mamdani approach : Mobile Robot

- We consider two parameters : $D$, the distance from the robot to an object and $\theta$ the angle of motion of an object with respect to the robot.

- The value of these parameters with respect to the most critical object will decide an output called deviation $(\delta)$.

- We assume the range of values of $D$ is [0.1, .... 2.2] in meter and $\theta$ is [-90, ... , 0, ... 90] in degree.

- After identifying the relevant input and output variables of the controller and their range of values, the Mamdani approach is to select some meaningful states called "linguistic states" for each variable and express them by appropriate fuzzy sets.

# Linguistic States

For the current example, we consider the following linguistic states for the three parameters.

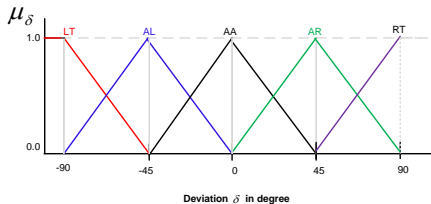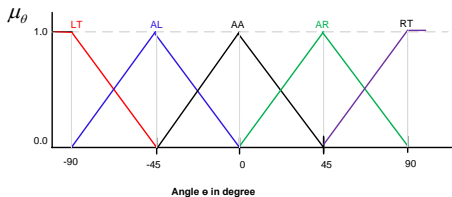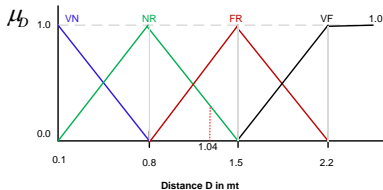**Distance** is represented using four linguistic states:

1. VN : Very Near
2. NR : Near
3. VF : Very Far
4. FR : Far

**Angle** (for both angular direction ($\theta$) and deviation ($\delta$)) are represented using five linguistic states:

1. LT : Left
2. AL : Ahead Left
3. AA: Ahead
4. AR : Ahead Right
5. RT : Right

# Linguistic States

Three different fuzzy sets for the three different parameters are given below (Figure 3).

## Fuzzy rule base

Once the fuzzy sets of all parameters are worked out, our next step in FLC design is to decide fuzzy rule base of the FLC.

The rule base for the FLC of mobile robot is shown in the form of a table below.

|    | LT | AL | AA | AR | RT |
|----|----|----|----|----|----|
| VN | AA | AR | AL | AL | AA |
| NR | AA | AA | RT | AA | AA |
| FR | AA | AA | AR | AA | AA |
| VF | AA | AA | AA | AA | AA |

## Fuzzy rule base for the mobile robot

Note that this rule base defines 20 rules for all possible instances.
These rules are simple rules and take in the following forms.

- Rule 1: If (distance is VN ) and (angle is LT) Then (deviation is AA)
  ⋮
  ⋮
- Rule 13: If (distance is FR ) and (angle is AA) Then (deviation is AR)
  ⋮
  ⋮
- Rule 20: Rule 1: If (distance is VF ) and (angle is RT) Then (deviation is AA)

## Fuzzification of inputs

- The next step is the fuzzification of inputs. Let us consider, at any instante, the object $O_3$ is critical to the Mobile Robot and distance D = 1.04 m and angle $\theta = 30^o$ (see Figure 2).

- For this input, we are to decide the deviation $\delta$ of the robot as output.

- From the given fuzzy sets and input parameters' values, we say that the distance D = 1.04m may be called as either NR (near) or FR (far).

- Similarly, the input angle $\theta = 30^o$ can be declared as either AA (ahead) or AR(ahead right).

## Fuzzification of inputs

Hence, we are to determine the membership values corresponding to these values, which is as follows.
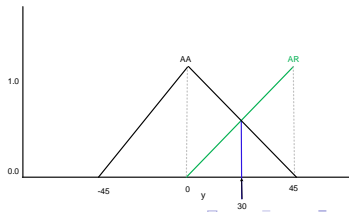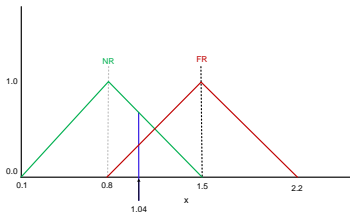
$$x = 1.04m$$
$$\mu_{NR}(x) = 0.6571$$
$$\mu_{FR}(x) = 0.3429$$
$$y = 30^o$$
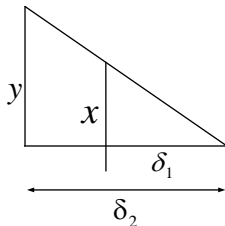$$\mu_{AA}(y) = 0.3333$$
$$\mu_{AR}(y) = 0.6667$$

## Fuzzification of inputs

**Hint** : Use the principle of similarity. $\frac{x}{y} = \frac{\delta_1}{\delta_2}$



Thus, $\frac{x}{1} = \frac{1.5-1.04}{1.5-0.8}$, that is, $x = 0.6571$

## Rule strength computation

There are many rules in the rule base and all rules may not be applicable.

For the given $x = 1.04$ and $\theta = 30^o$, only following four rules out of 20 rules are firable.

- $R1$: If (distance is NR) and (angle is AA) Then (deviation is RT)

- $R2$: If (distance is NR) and (angle is AR) Then (deviation is AA)

- $R3$: If (distance is FR) and (angle is AA) Then (deviation is AR)

- $R4$: If (distance is FR) and (angle is AR) Then (deviation is AA)

# Rule strength computation

The strength (also called $\alpha$ values) of the firable rules are calculated as follows.

- $\alpha(R1) = min(\mu_{NR}(x), \mu_{AA}(y)) = min(0.6571, 0.3333) = 0.3333$

- $\alpha(R2) = min(\mu_{NR}(x), \mu_{AR}(y)) = min(0.6571, 0.6667) = 0.6571$

- $\alpha(R3) = min(\mu_{FR}(x), \mu_{AA}(y)) = min(0.3429, 0.3333) = 0.3333$

- $\alpha(R4) = min(\mu_{FR}(x), \mu_{AR}(y)) = min(0.3429, 0.6667) = 0.3429$

In practice, all rules which are above certain threshold value of rule strngth are selected for the output computation.

## Fuzzy output

The next step is to determine the fuzzified outputs corresponding to each fired rules.

The working principle of doing this is first discussed and then we illustrate with the running example.

Suppose, only two fuzzy rules, $R1$ and $R2$, for which we are to decide fuzzy output.

- $R1$: IF ($s_1$ is $A_1$) AND ($s_2$ is $B_1$) THEN (f is $C_1$)

- $R2$: IF ($s_1$ is $A_2$) AND ($s_2$ is $B_2$) THEN (f is $C_2$)

Suppose, $s_1^*$ and $s_2^*$ are the inputs for fuzzy variables $s_1$ and $s_2$. $\mu_{A1}$, $\mu_{A2}$, $\mu_{B1}$, $\mu_{B2}$, $\mu_{C1}$ and $\mu_{C2}$ are the membership values for different fuzzy sets.

# Fuzzy output

The fuzzy output computation is graphically shows in the following figure.
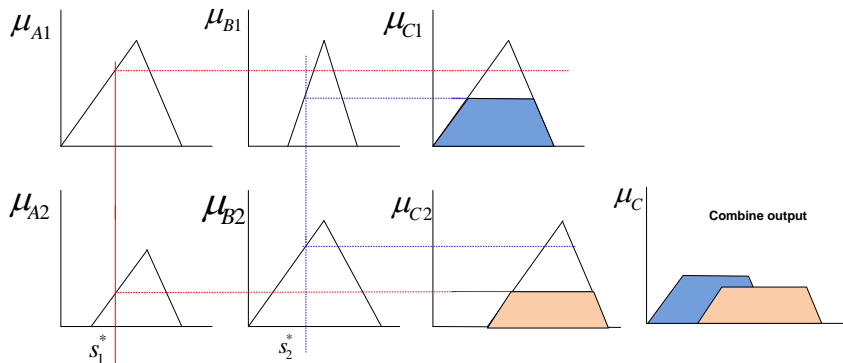
# Fuzzy output

**Note**:

- We take min of membership function values for each rule.

- Output membership function is obtained by aggregating the membership function of result of each rule.

- Fuzzy output is nothing but fuzzy OR of all output of rules.

# Illustration : Mobile Robot

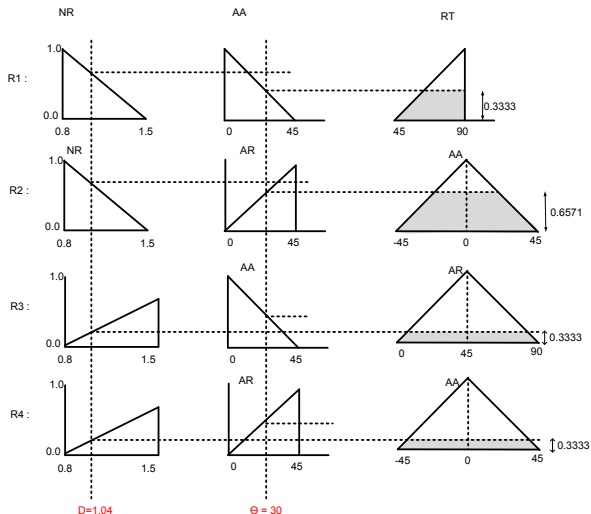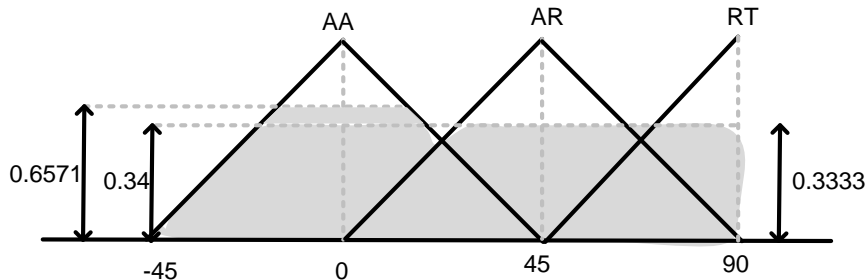For four rules, we find the following results.
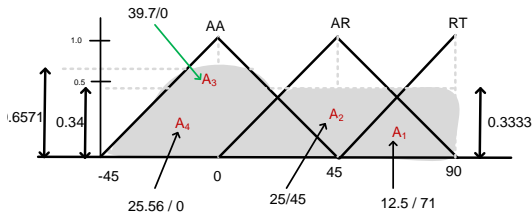
**Aggregation of all results**

# Defuzzification

The fuzzy output needs to be defuzzified and its crisp value has to be determined for the output to take decision.

## Illustration : Mobile Robot

From the combined fuzzified output for all four fired rules, we get the crisp value using Center of Sum method as follows.

$$v = \frac{12.5 \times 71 + 25 \times 45 + 25.56 \times 0 + 25.56 \times 0}{12.5 + 39.79 + 25 + 25.56} = 19.59$$



**Conclusion** : Therefore, the robot should deviate by 19.58089 degree towards the right with respect to the line joining to the move of direction to avoid collision with the obstacle $O_3$.

# Takagi and Sugeno's approach

- In this approach, a rule is composed of fuzzy antecedent and functional consequent parts.

- Thus, any i-th rule, in this approach is represented by
  If ($x_1$ is $A_1^i$) and ($x_2$ is $A_2^i$) ...... and ($x_n$ is $A_n^i$)

- Then, $y^i = a_0^i + a_1^i x_1 + a_2^i x_2 + ... + a_n^i x_n$
  where, $a_0$, $a_1$, $a_2$, ... $a_n$ are the co-efficients.

- The weight of i-th rule can be determined for a set of inputs $x_1$, $x_2$, .... $x_n$ as follows.

  $w^i = \mu_{A_1}^i(x_1) \times \mu_{A_2}^i(x_2) \times ..... \times \mu_{A_n}^i(x_n)$

  where $A_1$, $A_2$, .... , $A_n$ indicates membership function distributions of the linguistic hedges used to represent the input variables and $\mu$ denotes membership function value.

- The combined action then can be obtained as
  $y = \frac{\sum_i^k w^i y^i}{\sum_i^k w^i}$ ; where $k$ denotes the total number of rules.

## Illustration:

Consider two inputs $I_1$ and $I_2$. These two inputs have the following linguistic states :

$I_1$ : L(low), M(Medium), H(High)

$I_2$ : NR(Near), FR (Far), VF(Very Far)

**Note**:

The rule base of such as system is decided by a maximum of $3 \times 3 = 9$ feasible rules.

## Illustration:

The output of any i-th rule can be expressed by the following.

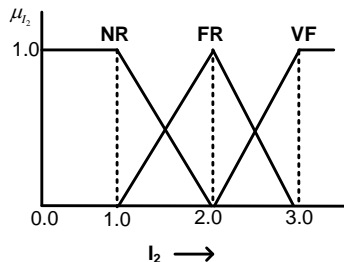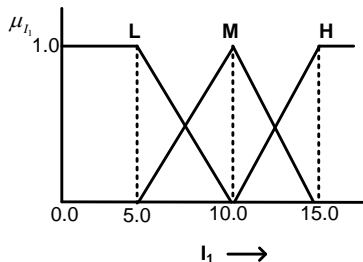$y^i = f(I_1, I_2) = a^i_j I_1 + b^i_k I_2$ ; where, j,k = 1,2,3.

**Suppose** : $a^i_1 = 1$, $a^i_2 = 2$, $a^i_3 = 3$ if $I_1 = L$, $M$ and $H$, respectively.

$b^i_1 = 1$, $b^i_2 = 2$, $b^i_3 = 3$ if $I_2 = NR$, $FR$, and $VF$, respectively.

We have to calculate the output of FLC for $I_1 = 6.0$ and $I_2 = 2.2$

## Illustration:

Given the distribution functions for $l_1$ and $l_2$ as below.

## Solution

a) The input $l_1 = 6.0$ can be called either L or M. Similarly, the input $l_2 = 2.2$ can be declared either FR or VF.

b) Using the principal of similarity of triangle, we have the following.
$\mu_L(l_1) = 0.8$
$\mu_M(l_1) = 0.2$
$\mu_{FR}(l_2) = 0.8$
$\mu_{VF}(l_2) = 0.2$

c) For the input set, following four rules can be fired out of all 9 rules.
R1: $l_1$ is L and $l_2$ is FR
R2: $l_1$ is L and $l_2$ is VF
R3: $l_1$ is M and $l_2$ is FR
R4: $l_1$ is M and $l_2$ is VF

## Solution

d) Now, the weights for each of the above rules can be determined as follows.

R1: $w^1 = \mu_L \times \mu_{FR} = 0.8 \times 0.8 = 0.6$

R2: $w^2 = \mu_L \times \mu_{VF} = 0.8 \times 0.2 = 0.16$

R3: $w^3 = \mu_M \times \mu_{FR} = 0.2 \times 0.8 = 0.16$

R4: $w^4 = \mu_M \times \mu_{VF} = 0.2 \times 0.2 = 0.6$

e) The functional consequent values for each rules can be calculated as below.

$y^1 = l_1 + 2l_2 = 6.0 + 2 \times 2.2 = 10.4$

$y^2 = l_1 + 3l_2 = 6.0 + 3 \times 2.2 = 12.6$

$y^3 = 2l_1 + 2l_2 = 2 \times 6.0 + 2 \times 2.2 = 16.4$

$y^4 = 2l_1 + 3l_2 = 2 \times 6.0 + 3 \times 2.2 = 18.6$

f) Therefore, the output $y$ of the controller can be determined as follows.

$y = \frac{w^1 y^1 + w^2 y^2 + w^3 y^3 + w^4 y^4}{w^1 + w^2 + w^3 + w^4} = 12.04$