SCT UNIT-3 - SCT

COMPUTER SCIENCE ENGINEERING (Jawaharlal Nehru Technological University, Kakinada)

SOFT COMPUTING TECHNIQUES

UNIT-3

UNIT -III:

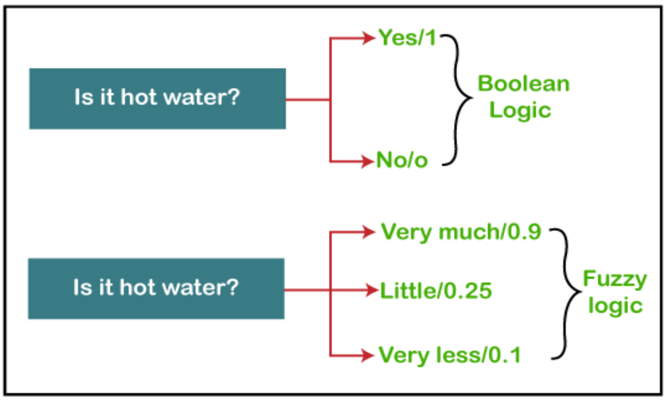
Fuzzy Logic System: Introduction to fuzzy logic, classical sets and fuzzy sets, fuzzy set operations, fuzzy relations, fuzzy composition, natural language and fuzzy interpretations, fuzzy inference system, fuzzy controllers

Introduction to fuzzy logic

What is Fuzzy Logic?

The 'Fuzzy' word means the things that are not clear or are vague. Sometimes, we cannot decide in real life that the given problem or statement is either true or false. At that time, this concept provides many values between the true and false and gives the flexibility to find the best solution to that problem.

Example of Fuzzy Logic as comparing to Boolean Logic



Fuzzy logic contains the multiple logical values and these values are the truth values of a variable or problem between 0 and 1. This concept was introduced by Lofti Zadeh in 1965 based on the Fuzzy Set Theory. This concept provides the possibilities which are not given by computers, but similar to the range of possibilities generated by humans.

In the Boolean system, only two possibilities (0 and 1) exist, where 1 denotes the absolute truth value and 0 denotes the absolute false value. But in the fuzzy system, there are multiple possibilities present between the 0 and 1, which are partially false and partially true.

The Fuzzy logic can be implemented in systems such as micro-controllers, workstation-based or large network-based systems for achieving the definite output. It can also be implemented in both hardware or software.

Characteristics of Fuzzy Logic

Following are the characteristics of fuzzy logic:

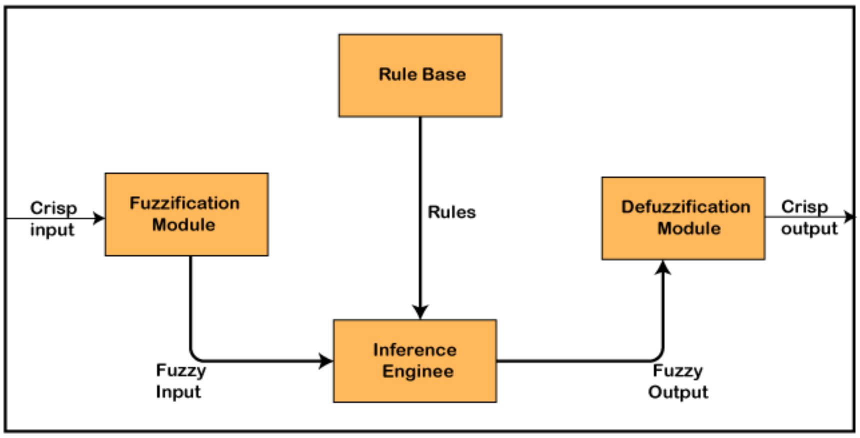
1. This concept is flexible and we can easily understand and implement it.
2. It is used for helping the minimization of the logics created by the human.
3. It is the best method for finding the solution of those problems which are suitable for approximate or uncertain reasoning.
4. It always offers two values, which denote the two possible solutions for a problem and statement.
5. It allows users to build or create the functions which are non-linear of arbitrary complexity.
6. In fuzzy logic, everything is a matter of degree.
7. In the Fuzzy logic, any system which is logical can be easily fuzzified.
8. It is based on natural language processing.
9. It is also used by the quantitative analysts for improving their algorithm's execution.
10. It also allows users to integrate with the programming.

Architecture of a Fuzzy Logic System

In the architecture of the Fuzzy Logic system, each component plays an important role. The architecture consists of the different four components which are given below.

1. Rule Base
2. Fuzzification
3. Inference Engine
4. Defuzzification

Following diagram shows the architecture or process of a Fuzzy Logic system:



1. Rule Base

Rule Base is a component used for storing the set of rules and the If-Then conditions given by the experts are used for controlling the decision-making systems. There are so many updates that come in the Fuzzy theory recently, which offers effective methods for designing and tuning of fuzzy controllers. These updates or developments decreases the number of fuzzy set of rules.

2. Fuzzification

Fuzzification is a module or component for transforming the system inputs, i.e., it converts the crisp number into fuzzy steps. The crisp numbers are those inputs which are measured by the sensors and then fuzzification passed them into the control systems for further processing. This component divides the input signals into following five states in any Fuzzy Logic system:

- Large Positive (LP)
- Medium Positive (MP)
- Small (S)
- Medium Negative (MN)
- Large negative (LN)

3. Inference Engine

This component is a main component in any Fuzzy Logic system (FLS), because all the information is processed in the Inference Engine. It allows users to find the matching degree between the current fuzzy input and the rules. After the matching degree, this system determines which rule is to be added according to the given input field. When all rules are fired, then they are combined for developing the control actions.

4. Defuzzification

Defuzzification is a module or component, which takes the fuzzy set inputs generated by the Inference Engine, and then transforms them into a crisp value. It is the last step in the process of a fuzzy logic system. The crisp value is a type of value which is acceptable by the user. Various techniques are present to do this, but the user has to select the best one for reducing the errors.

Classical sets and Fuzzy sets

Classical set

1. Classical set is a collection of distinct objects. For example, a set of students passing grades.
2. Each individual entity in a set is called a member or an element of the set.
3. The classical set is defined in such a way that the universe of discourse is splitted into two groups members and non-members. Hence, In case classical sets, no partial membership exists.
4. Let A is a given set. The membership function can be use to define a set A is given by:

$$\mu A(x)=\{\ 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A\ \}$$

1. **Operations on classical sets:** For two sets A and B and Universe X:
    - **Union:**

    $$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$$

    - This operation is also called logical OR.
    - **Intersection:**

    $$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$

    - This operation is also called logical AND.

- **Complement:**

$$A' = \{x \mid x \notin A, x \in X\}$$

- **Difference:**

$$A|B = \{x \mid x \in A \text{ and } x \notin B\}$$

2. **Properties of classical sets: For two sets A and B and Universe X:**
    - **Commutativity:**

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

    - **Associativity:**

$$A \cup (B \cup C) = (A \cup B) \cup C$$

$$A \cap (B \cap C) = (A \cap B) \cap C$$

    - **Distributivity:**

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

    - **Idempotency:**

$$A \cup A = A$$

$$A \cap A = A$$

    - **Identity:**

$$A \cup \emptyset = A$$

$$A \cap X = A$$

$$A \cap \emptyset = \emptyset$$

$$A \cup X = X$$

    - **Transitivity:**

$$\text{If } A \subseteq B \text{ and } B \subseteq C, \text{ then } A \subseteq C$$

Fuzzy Set

1. Fuzzy set is a set having degrees of membership between 1 and 0. Fuzzy sets are represented with tilde character(~). For example, Number of cars following traffic signals at a particular time out of all cars present will have membership value between [0,1].
2. Partial membership exists when member of one fuzzy set can also be a part of other fuzzy sets in the same universe.

3. The degree of membership or truth is not same as probability, fuzzy truth represents membership in vaguely defined sets.
4. A fuzzy set A~ in the universe of discourse, U, can be defined as a set of ordered pairs and it is given by

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\}$$

1. Common Operations on fuzzy sets: Given two Fuzzy sets A~ and B~
   - Union : Fuzzy set C~ is union of Fuzzy sets A~ and B~ :

$$\tilde{C} = \tilde{A} \cup \tilde{B},$$

$$\mu_{\tilde{C}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

   - Intersection: Fuzzy set D~ is intersection of Fuzzy sets A~ and B~ :

$$\tilde{D} = \tilde{A} \cap \tilde{B}$$

$$\mu_{\tilde{D}}(x) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$$

   - Complement: Fuzzy set E~ is complement of Fuzzy set A~ :

$$\tilde{E} = \mathbb{C}_{\tilde{A}} X$$

$$\mu_{\tilde{E}}(x) = 1 - \mu_{\tilde{A}}(x)$$

Fuzzy relations

Fuzzy relation defines the mapping of variables from one fuzzy set to another. Like crisp relation, we can also define the relation over fuzzy sets.

Let A be a fuzzy set on universe X and B be a fuzzy set on universe Y, then the Cartesian product between fuzzy sets A and B will result in a fuzzy relation R which is contained with the full Cartesian product space or it is a subset of the cartesian product of fuzzy subsets. Formally, we can define fuzzy relation as,

R = A x B

and

R ⊂ (X x Y)

where the relation R has a membership function,

μR(x, y) = μA x B(x, y) = min(μA(x), μB(y))

A binary fuzzy relation R(X, Y) is called a bipartite graph if X ≠ Y.

A binary fuzzy relation R(X, Y) is called directed graph or digraph if X = Y. , which is denoted as R(X, X) = R(X2)

Let A = {a1, a2, …, an} and B = {b1, b2, .., bm}, then the fuzzy relation between A and B is described by the fuzzy relation matrix as,

Fuzzy Relations

Let $\underset{\sim}{X} = \{x_1, x_2, \ldots, x_n\}$, $\underset{\sim}{Y} = \{y_1, y_2, \ldots, y_m\}$. The fuzzy relation $\underset{\sim}{R}(\underset{\sim}{X}, \underset{\sim}{Y})$ can be expressed by $n \times m$ matrix called *Fuzzy Matrix* denoted as:

$$\underset{\sim}{R}(\underset{\sim}{X}, \underset{\sim}{Y}) = \begin{bmatrix} \mu_{\underset{\sim}{R}}(x_1, y_1) & \mu_{\underset{\sim}{R}}(x_1, y_2) & \cdots & \mu_{\underset{\sim}{R}}(x_1, y_m) \\ \mu_{\underset{\sim}{R}}(x_2, y_1) & \mu_{\underset{\sim}{R}}(x_2, y_2) & \cdots & \mu_{\underset{\sim}{R}}(x_2, y_m) \\ \cdots & \cdots & \cdots & \cdots \\ \mu_{\underset{\sim}{R}}(x_n, y_1) & \mu_{\underset{\sim}{R}}(x_n, y_2) & \cdots & \mu_{\underset{\sim}{R}}(x_n, y_m) \end{bmatrix}$$

We can also consider fuzzy relation as a mapping from the cartesian space (X, Y) to the interval [0, 1]. The strength of this mapping is represented by the membership function of the relation for every tuple μR(x, y)

Example:

Given A = { (a1, 0.2), (a2, 0.7), (a3, 0.4) } and B = { (b1, 0.5), (b2, 0.6)}, find the relation over A x B

Fuzzy relations are very important because they can describe interactions between variables.

Example: A simple example of a binary fuzzy relation on X = {1, 2, 3}, called ”approximately equal” can be defined as

$$R(1, 1) = R(2, 2) = R(3, 3) = 1$$
$$R(1, 2) = R(2, 1) = R(2, 3) = R(3, 2) = 0.8$$
$$R(1, 3) = R(3, 1) = 0.3$$

The membership function and relation matrix of R are given by

$$\bar{R}(x, y) = \begin{cases} 1, & if x = y \\ 0.7, & if |x - y| = 1 \\ 0.3, & if |x - y| = 2 \end{cases}$$

$$\bar{R} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1.0 & 0.7 & 0.3 \\ 0.7 & 1.0 & 0.7 \\ 0.3 & 0.7 & 1.0 \end{bmatrix} \end{array}$$

Operations on fuzzy relation:

For our discussion, we will be using the following two relation matrices:

$$\bar{R} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 0.8 & 0.1 & 0.1 & 0.7 \\ 0.0 & 0.8 & 0.0 & 0.0 \\ 0.9 & 1.0 & 0.7 & 0.8 \end{bmatrix} \end{array}$$

$$\bar{S} = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 0.4 & 0.0 & 0.9 & 0.6 \\ 0.9 & 0.4 & 0.5 & 0.7 \\ 0.3 & 0.0 & 0.8 & 0.5 \end{bmatrix} \end{array}$$

Union:

R ∪ S = { (a, b), μA ∪ B(a, b) }

μR ∪ S(a, b) = max(μR(a, b), μS(a, b))

μR ∪ S(x1, y1) = max(μR(x1, y1), μS(x1, y1))

= max(0.8, 0.4) = 0.8

μR ∪ S(x1, y2) = max(μR(x1, y2), μS(x1, y2))

= max(0.1, 0.0) = 0.1

μR ∪ S(x1, y3) = max(μR(x1, y3), μS(x1, y3))

= max(0.1, 0.9) = 0.9

μR ∪ S(x1, y4) = max(μR(x1, y4), μS(x1, y4))

max(0.7, 0.6) = 0.7

.

.

.

μR ∪ S(x3, y4) = max(μR(x3, y4), μS(x3, y4))

= max(0.8, 0.5) = 0.8

Thus, the final matrix for union operation would be,

$$\bar{R} \cup \bar{S} = \begin{array}{c c} & \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \begin{bmatrix} 0.8 & 0.1 & 0.9 & 0.7 \\ 0.9 & 0.8 & 0.5 & 0.7 \\ 0.9 & 1.0 & 0.8 & 0.8 \end{bmatrix} \end{array}$$

Union of fuzzy relations

Intersection:

R ∩ S = { (a, b), μA ∩ B(a, b) }

μR ∩ S(a, b) = min(μR(a, b), μS(a, b))

μR ∩ S(x1, y1) = min(μR(x1, y1), μS(x1, y1))

= min(0.8, 0.4) = 0.4

μR ∩ S(x1, y2) = min(μR(x1, y2), μS(x1, y2))

= max(0.1, 0.0) = 0.0

μR ∩ S(x1, y3) = min(μR(x1, y3), μS(x1, y3))

= max(0.1, 0.9) = 0.1

μR ∩ S(x1, y4) = min(μR(x1, y4), μS(x1, y4))

max(0.7, 0.6) = 0.6

.

.

.

μR ∩ S(x3, y4) = min(μR(x3, y4), μS(x3, y4))

= max(0.8, 0.5) = 0.5

$$\bar{R} \cap \bar{S} = \begin{array}{c c} & \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \begin{bmatrix} 0.4 & 0.0 & 0.1 & 0.6 \\ 0.0 & 0.4 & 0.0 & 0.0 \\ 0.3 & 0.0 & 0.7 & 0.5 \end{bmatrix} \end{array}$$

Intersection of relation

Complement:

Rc = { (a, b), µRc(a, b) }

µRc(a, b) = 1 – µR(a, b)

µRc(x1, y1) = 1 – µR(x1, y1) = 1 – 0.8 = 0.2

µRc(x1, y2) = 1 – µR(x1, y2) = 1 – 0.1 = 0.9

µRc(x1, y3) = 1 – µR(x1, y3) = 1 – 0.1 = 0.9

.

.

µRc(x3, y4) = 1 – µR(x3, y4) = 1 – 0.8 = 0.2

The complement of relation R would be,

$$\bar{R}^c = \begin{array}{c c} & \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} & \begin{bmatrix} 0.2 & 0.9 & 0.9 & 0.3 \\ 1.0 & 0.2 & 1.0 & 1.0 \\ 0.1 & 0.0 & 0.3 & 0.2 \end{bmatrix} \end{array}$$

Complement of fuzzy relation

Fuzzy composition

Fuzzy composition can be defined just as it is for crisp (binary) relations. Suppose R is a fuzzy relation on X × Y, S is a fuzzy relation on Y × Z, and T is a fuzzy relation on X × Z; then,

Fuzzy Max–Min composition is defined as:

$$\underline{T} = \underline{R} \circ \underline{S} = \mu_{\bar{T}}(x, z) = \bigvee_{y \in Y} (\mu_{\bar{R}}(x, y) \wedge \mu_{\bar{S}}(y, z))$$

$$= max_{y \in Y}\{min(\mu_{\bar{R}}(x, y), \mu_{\bar{S}}(y, z))\}$$

Fuzzy Max–Product composition is defined as:

$$\underline{T} = \underline{R} \circ \underline{S} = \mu_{\bar{T}}(x,z) = \bigvee_{y \in Y} (\mu_{\bar{R}}(x,y) . \mu_{\bar{S}}(y,z))$$

$$= max_{y \in Y}\{(\mu_{\bar{R}}(x,y) \times \mu_{\bar{S}}(y,z))\}$$

Let us try to understand it with the help of an example. The fuzzy max-min approach is identical to that of the crisp max-min composition.

Example:

X = {x1, x2}, Y = {y1, y2}, and Z = {z1, z2, z3}. Consider the following fuzzy relations:

$$\bar{R} = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.7 & 0.6 \\ 0.8 & 0.3 \end{bmatrix} \end{array}$$

$$\bar{S} = \begin{array}{c} \\ y_1 \\ y_2 \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.8 & 0.5 & 0.4 \\ 0.1 & 0.6 & 0.7 \end{bmatrix} \end{array}$$

Find the resulting relation, T which relates elements of universe X to elements of universe Z, i.e., defined on Cartesian space X × Z

- Using Max-Min composition and
- Using Max-Product composition

Solution:

So ultimately, we have to find the elements of the matrix,

$$\bar{T} = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} \ & \ & \ \\ \ & \ & \ \end{bmatrix} \end{array}$$

Composition of relation R and S

Max-Min Composition:

Max-min composition is defined as,

$$\underline{T} = \underline{R} \circ \underline{S} = \mu_{\bar{T}}(x,z) = \bigvee_{y \in Y} (\mu_{\bar{R}}(x,y) \wedge \mu_{\bar{S}}(y,z))$$

$$= max_{y \in Y}\{min(\mu_{\bar{R}}(x,y), \mu_{\bar{S}}(y,z))\}$$

From the given relations R and S,

μT(x1, z1) = max (min(μR(x1, y1), μS(y1, z1)), min(μR(x1, y2), μS(y2, z1)))

= max(min(0.7, 0.8), min(0.6, 0.1)) = max(0.7, 0.1) = 0.7

μT(x1, z2) = max (min(μR(x1, y1), μS(y1, z2)), min(μR(x1, y2), μS(y2, z2)))

= max(min(0.7, 0.5), min(0.6, 0.6)) = max(0.5, 0.6) = 0.6
µT(x1, z3) = max (min(µR(x1, y1), µS(y1, z3)), min(µR(x1, y2), µS(y2, z3)))
= max(min(0.7, 0.4), min(0.6, 0.7)) = max(0.4, 0.6) = 0.6
µT(x2, z1) = max (min(µR(x2, y1), µS(y1, z1)), min(µR(x2, y2), µS(y2, z1)))
= max(min(0.8, 0.8), min(0.3, 0.1)) = max(0.8, 0.1) = 0.8
µT(x2, z2) = max (min(µR(x2, y1), µS(y1, z2)), min(µR(x2, y2), µS(y2, z2)))
= max(min(0.8, 0.5), min(0.3, 0.6)) = max(0.5, 0.3) = 0.5
µT(x2, z3) = max (min(µR(x2, y1), µS(y1, z3)), min(µR(x2, y2), µS(y2, z3)))
= max(min(0.8, 0.4), min(0.3, 0.7)) = max(0.4, 0.3) = 0.4

$$\bar{T} = \begin{array}{c c} & \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{bmatrix} 0.7 & 0.6 & 0.6 \\ 0.8 & 0.5 & 0.4 \end{bmatrix} \end{array}$$

Max-Min composition of Relations

Max-Product Composition:

$$\underline{T} = \underline{R} \circ \underline{S} = \mu_{\bar{T}}(x, z) = \underset{y \in Y}{\vee} (\mu_{\bar{R}}(x, y).\, \mu_{\bar{S}}(y, z))$$

$$= max_{y \in Y}\{(\mu_{\bar{R}}(x, y) \times \mu_{\bar{S}}(y, z))\}$$

µT(x1, z1) = max ((µR(x1, y1) × µS(y1, z1)), (µR(x1, y2) × µS(y2, z1)))
= max((0.7 × 0.8), (0.6 × 0.1)) = max(0.56, 0.06) = 0.56
µT(x1, z2) = max ((µR(x1, y1) × µS(y1, z2)), (µR(x1, y2) × µS(y2, z2)))
= max((0.7 × 0.5), (0.6 × 0.6)) = max(0.35, 0.36) = 0.36
µT(x1, z3) = max ((µR(x1, y1) × µS(y1, z3)), (µR(x1, y2) × µS(y2, z3)))
= max((0.7 × 0.4), (0.6 × 0.7)) = max(0.28, 0.42) = 0.42
µT(x2, z1) = max ((µR(x2, y1) × µS(y1, z1)), (µR(x2, y2) × µS(y2, z1)))
= max((0.8 × 0.8), min(0.3 × 0.1)) = max(0.64, 0.03) = 0.64
µT(x2, z2) = max ((µR(x2, y1) × µS(y1, z2)), (µR(x2, y2) × µS(y2, z2)))
= max((0.8 × 0.5), (0.3 × 0.6)) = max(0.4, 0.18) = 0.40
µT(x2, z3) = max ((µR(x2, y1) × µS(y1, z3)), (µR(x2, y2) × µS(y2, z3)))
= max((0.8 × 0.4), (0.3 × 0.7)) = max(0.32, 0.21) = 0.32

$$\bar{T} = \begin{array}{c c} & \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{array}{c} x_1 \\ x_2 \end{array} & \begin{bmatrix} 0.56 & 0.36 & 0.42 \\ 0.64 & 0.40 & 0.32 \end{bmatrix} \end{array}$$

Max-Product composition

Fuzzy Inference System

Fuzzy Inference System is the key unit of a fuzzy logic system having decision making as its primary work. It uses the “IF…THEN” rules along with connectors “OR” or “AND” for drawing essential decision rules.

Characteristics of Fuzzy Inference System

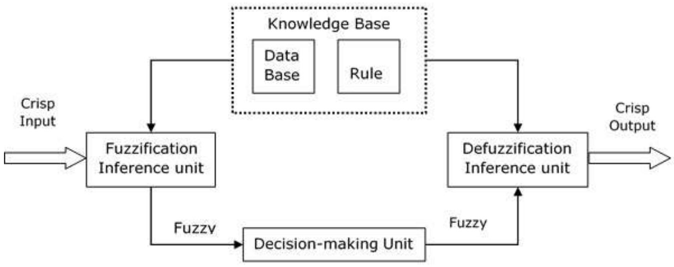
Following are some characteristics of FIS –

- The output from FIS is always a fuzzy set irrespective of its input which can be fuzzy or crisp.
- It is necessary to have fuzzy output when it is used as a controller.
- A defuzzification unit would be there with FIS to convert fuzzy variables into crisp variables.

Functional Blocks of FIS

The following five functional blocks will help you understand the construction of FIS –

- **Rule Base –** It contains fuzzy IF-THEN rules.
- **Database –** It defines the membership functions of fuzzy sets used in fuzzy rules.
- **Decision-making Unit –** It performs operation on rules.
- **Fuzzification Interface Unit –** It converts the crisp quantities into fuzzy quantities.
- **Defuzzification Interface Unit –** It converts the fuzzy quantities into crisp quantities. Following is a block diagram of fuzzy interference system.



Working of FIS

The working of the FIS consists of the following steps –

- A fuzzification unit supports the application of numerous fuzzification methods, and converts the crisp input into fuzzy input.
- A knowledge base - collection of rule base and database is formed upon the conversion of crisp input into fuzzy input.
- The defuzzification unit fuzzy input is finally converted into crisp output.

Methods of FIS

Let us now discuss the different methods of FIS. Following are the two important methods of FIS, having different consequent of fuzzy rules –

- Mamdani Fuzzy Inference System
- Takagi-Sugeno Fuzzy Model (TS Method)

Mamdani Fuzzy Inference System

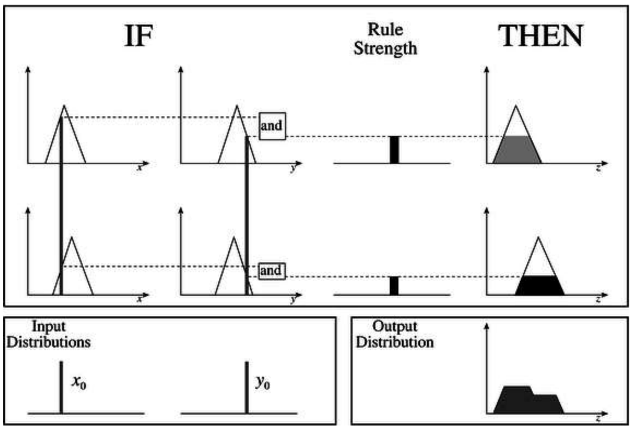
This system was proposed in 1975 by Ebhasim Mamdani. Basically, it was anticipated to control a steam engine and boiler combination by synthesizing a set of fuzzy rules obtained from people working on the system.

Steps for Computing the Output

Following steps need to be followed to compute the output from this FIS –

- Step 1 − Set of fuzzy rules need to be determined in this step.
- Step 2 − In this step, by using input membership function, the input would be made fuzzy.
- Step 3 − Now establish the rule strength by combining the fuzzified inputs according to fuzzy rules.
- Step 4 − In this step, determine the consequent of rule by combining the rule strength and the output membership function.
- Step 5 − For getting output distribution combine all the consequents.
- Step 6 − Finally, a defuzzified output distribution is obtained.

Following is a block diagram of Mamdani Fuzzy Interface System.



Takagi-Sugeno Fuzzy Model (TS Method)

This model was proposed by Takagi, Sugeno and Kang in 1985. Format of this rule is given as –

IF x is A and y is B THEN Z = f(x,y)

Here, AB are fuzzy sets in antecedents and z = f(x,y) is a crisp function in the consequent.

Fuzzy Inference Process

The fuzzy inference process under Takagi-Sugeno Fuzzy Model (TS Method) works in the following way –

- Step 1: Fuzzifying the inputs − Here, the inputs of the system are made fuzzy.
- Step 2: Applying the fuzzy operator − In this step, the fuzzy operators must be applied to get the output.

Rule Format of the Sugeno Form

The rule format of Sugeno form is given by −

if 7 = x and 9 = y then output is z = ax+by+c

Fuzzy controllers

Why Use Fuzzy Logic in Control Systems

A control system is an arrangement of physical components designed to alter another physical system so that this system exhibits certain desired characteristics. Following are some reasons of using Fuzzy Logic in Control Systems –

- While applying traditional control, one needs to know about the model and the objective function formulated in precise terms. This makes it very difficult to apply in many cases.
- By applying fuzzy logic for control we can utilize the human expertise and experience for designing a controller.
- The fuzzy control rules, basically the IF-THEN rules, can be best utilized in designing a controller.

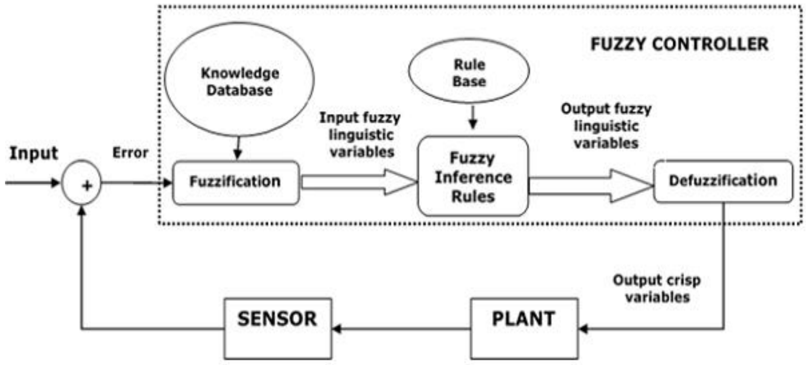
Assumptions in Fuzzy Logic Control (FLC) Design

While designing fuzzy control system, the following six basic assumptions should be made –

- The plant is observable and controllable − It must be assumed that the input, output as well as state variables are available for observation and controlling purpose.
- Existence of a knowledge body − It must be assumed that there exist a knowledge body having linguistic rules and a set of input-output data set from which rules can be extracted.
- Existence of solution − It must be assumed that there exists a solution.
- 'Good enough' solution is enough − The control engineering must look for 'good enough' solution rather than an optimum one.
- Range of precision − Fuzzy logic controller must be designed within an acceptable range of precision.
- Issues regarding stability and optimality − The issues of stability and optimality must be open in designing Fuzzy logic controller rather than addressed explicitly.

Architecture of Fuzzy Logic Control

The following diagram shows the architecture of Fuzzy Logic Control (FLC).



Major Components of FLC

Followings are the major components of the FLC as shown in the above figure –

- Fuzzifier − The role of fuzzifier is to convert the crisp input values into fuzzy values.
- Fuzzy Knowledge Base − It stores the knowledge about all the input-output fuzzy relationships. It also has the membership function which defines the input variables to the fuzzy rule base and the output variables to the plant under control.
- Fuzzy Rule Base − It stores the knowledge about the operation of the process of domain.
- Inference Engine − It acts as a kernel of any FLC. Basically it simulates human decisions by performing approximate reasoning.
- Defuzzifier − The role of defuzzifier is to convert the fuzzy values into crisp values getting from fuzzy inference engine.

Steps in Designing FLC

Following are the steps involved in designing FLC –

- Identification of variables − Here, the input, output and state variables must be identified of the plant which is under consideration.
- Fuzzy subset configuration − The universe of information is divided into number of fuzzy subsets and each subset is assigned a linguistic label. Always make sure that these fuzzy subsets include all the elements of universe.
- Obtaining membership function − Now obtain the membership function for each fuzzy subset that we get in the above step.
- Fuzzy rule base configuration − Now formulate the fuzzy rule base by assigning relationship between fuzzy input and output.
- Fuzzification − The fuzzification process is initiated in this step.
- Combining fuzzy outputs − By applying fuzzy approximate reasoning, locate the fuzzy output and merge them.
- Defuzzification − Finally, initiate defuzzification process to form a crisp output.

Advantages of Fuzzy Logic Control

Let us now discuss the advantages of Fuzzy Logic Control.

- Cheaper − Developing a FLC is comparatively cheaper than developing model based or other controller in terms of performance.
- Robust − FLCs are more robust than PID controllers because of their capability to cover a huge range of operating conditions.
- Customizable − FLCs are customizable.
- Emulate human deductive thinking − Basically FLC is designed to emulate human deductive thinking, the process people use to infer conclusion from what they know.
- Reliability − FLC is more reliable than conventional control system.
- Efficiency − Fuzzy logic provides more efficiency when applied in control system.

Disadvantages of Fuzzy Logic Control

- Requires lots of data − FLC needs lots of data to be applied.
- Useful in case of moderate historical data − FLC is not useful for programs much smaller or larger than historical data.
- Needs high human expertise − This is one drawback as the accuracy of the system depends on the knowledge and expertise of human beings.
- Needs regular updating of rules − The rules must be updated with time.