Chapter 5

FUZZY LOGIC - AN OVERVIEW

5.1 WHAT IS FUZZY LOGIC?

Fuzzy Logic (FL) is built on the *Fuzzy Set Theory*, which was introduced to the world, for the first time, by Lotfi Zadeh in 1965 [68]. In its narrow sense, fuzzy logic is logic of approximate reasoning which may be viewed as generalization and extension of multi-valued logic. But in a broader and much more significant sense, fuzzy logic is coextensive with the theory of fuzzy sets, that is, classes of objects in which the transition from membership to non-membership is gradual rather than abrupt. Fuzzy logic is an extension of classical logic and uses fuzzy sets rather than classical sets. The invention, or proposition, of Fuzzy Sets was motivated by the need to capture and represent the real world with its fuzzy data due to uncertainty. Uncertainty can be caused by imprecision in measurement due to imprecision of tools or other factors. Uncertainty can also be caused by vagueness in the language. Lotfi Zadeh realized that the crisp set Theory is not capable of representing those descriptions and classifications in many cases. In fact, crisp sets do not provide adequate representation for most cases.

Fuzzy set theory provides a means for repesenting uncertainities. Historically, probability theory has been the primary tool for representing uncertainty in mathematical model. Because of this, all uncertainty was assumed to follow the characteristics of random uncertainty. However, not all uncertainty is random. Some forms of uncertainties such as uncertainty associated with complex systems and issues, which human address on daily basis are nonrandom in nature. Fuzzy set theory is marvellous tool for modeling such kind of uncertainty regarding the problem at hand. The incorporation of fuzzy set theory and fuzzy logic into computer models has shown tremendous payoff in areas where intuition and judgement still play important roles in the model. The nature of uncertainty in a problem is a very important point that engineers should ponder prior to their selection of an appropriate method to express the uncertainty. Fuzzy sets provide a mathematical way to represent vagueness in humanistic systems. S*et membership* is the key to decision making when faced with uncertainty.

In engineering design problems, design methods and constraint evaluation involves fuzziness and imprecision. Also during the design state, the decision making in design is mainly based on the conceptual understanding. Unfortunately such information is usually vaguely defined by the experts. Therefore consideration of imprecise and vague information becomes an important aspect in the structural optimization problems. Fuzzy Logic provides an easy way of dealing with complex problems containing vagueness and imprecision. In this chapter, general concepts and applications of fuzzy logic are described in detail.

5.2 FUZZY SET AND FUZZY SET OPERATIONS

Fuzzy sets are an extension of classical set theory and are used in fuzzy logic. In classical set theory the membership of elements in relation to a set is assessed in binary terms (i.e 0 or 1) according to a crisp condition - an element either belongs (i.e. 1) or does not belong (i.e. 0) to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in relation to a set; this is described with the aid of a membership function $\mu \rightarrow [0, 1]$. Figure 5.1 explains this difference very clearly. Fuzzy systems, including fuzzy logic and fuzzy set theory, provide a rich and meaningful addition to standard logic. The mathematics generated by these theories is consistent, and fuzzy logic may be a generalization of classic logic. The applications which may be generated from or adapted to fuzzy logic are wide-ranging, and provide the opportunity for modeling of conditions which are inherently imprecisely defined, despite the concerns of classical logicians. Many systems may be modeled, simulated, and even replicated with the help of fuzzy systems, not the least of which is human reasoning itself.

A Fuzzy Set is a collection of distinct elements with varying degree of relevance or inclusion. The characteristic function determines the degree of relevance or inclusion which is also known as Membership Function (MF), can take interval values between 1 and 0 and shown inside brackets [1, 0]. A membership function is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1 as shown in Fig. 5.1. The input space is sometimes referred to as the *universe of discourse.* Fuzzy set can be expressed as $A = \{(x, \mu_A(x))\}$, x belongs to X where μ denotes membership function and $(x, \mu_A(x))$ is a singleton. The membership function is represented by union of singletons.

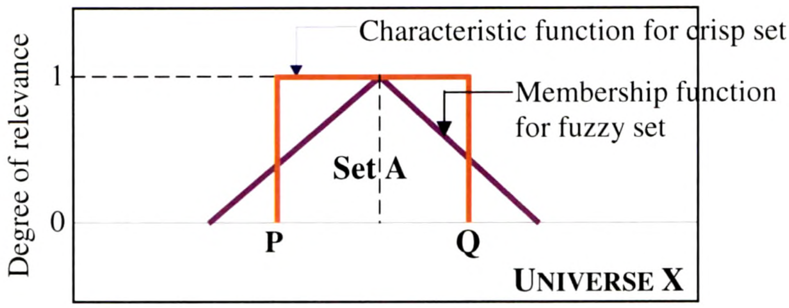


Fig. 5.1 Crisp Set and Fuzzy Set

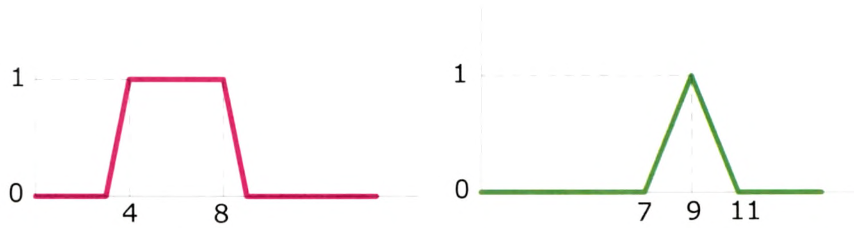
The standard definitions of fuzzy set operations in fuzzy logic as suggested by Lotfi are:

(i) Negate (negation criterion) : truth (not x) = 1.0 - truth (x)

(ii) Intersection (minimum criterion): truth (x and y) = minimum (truth(x), truth(y))

(iii)Union (maximum criterion): truth (x or y) = maximum (truth(x), truth(y))

This can be clarified through the example of two fuzzy sets $\tilde{A}$ and $\tilde{B}$ as shown in Figs. 5.2(a) and 5.2(b) respectively. Let $\tilde{A}$ be a fuzzy interval between 3 and 9 and $\tilde{B}$ be a fuzzy interval between 7 and 11 in the universe of discourse X. Fuzzy operations which are the basis of various fuzzy systems are (i) fuzzy intersection, (ii) fuzzy union and (iii) fuzzy complement and are shown below in mathematical and graphical form. Figures 5.3(a), (b) and (c) show membership functions of sets resulting from union and intersection of $\tilde{A}$ and $\tilde{B}$ and complement of $\tilde{A}$.



(a) Fuzzy Set $\tilde{A}$ **(b) Fuzzy Set $\tilde{B}$**

Fig. 5.2 Examples of Fuzzy Sets

(i) Fuzzy union (Fig. 5.3(a))

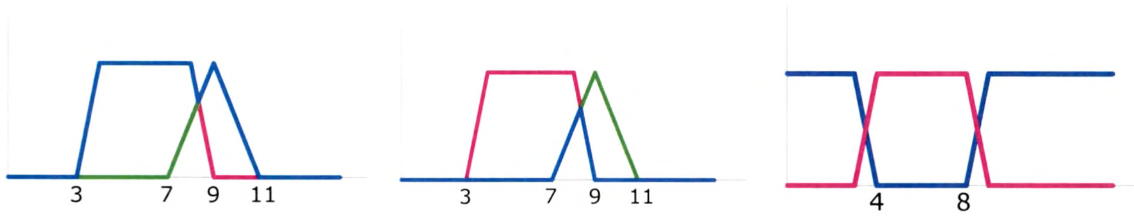
$$A \cup B \Leftrightarrow \mu_{A \cup B(x)} = max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \quad \ldots (5.1)$$

(ii) Fuzzy intersection (Fig. 5.3(b))

$$A \cap B \Leftrightarrow \mu_{A \cap B(x)} = min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \quad \ldots (5.2)$$

(iii) Fuzzy complement (Fig. 5.3(c))

$$\overline{A} = X - A \Leftrightarrow \mu_{\overline{A}}(x) = 1 - \mu_A(x) \qquad \ldots (5.3)$$



(a) Fuzzy union ($\tilde{A} \cup \tilde{B}$) **(b) Fuzzy intersection ($\tilde{A} \cap \tilde{B}$)** **(c) Fuzzy compliment ($\overline{A}$)**

Fig. 5.3 Fuzzy Operations

5.3 FUZZY MATHEMATICAL PROGRAMMING

Mathematical programming problems (MP) form a subclass of decision-making problems where preferences between alternatives are described by means of objective function(s) defined on the set of alternatives in such a way that greater values of the function(s) correspond to more preferable alternatives (if "higher" is "better"). The values of the objective function describe effects from choices of the alternatives. In structural engineering problems, for example, these values may reflect cost of the structure when using various shapes, size, topology and material of construction. The set of feasible alternatives in MP problems is described implicitly by means of constraints - equations or inequalities, or both - representing relevant relationships between alternatives. In any case the results of the analysis using given formulation of the MP problem depend largely upon how adequately various factors of the real system are reflected in the description of the objective function(s) and the constraint(s). Descriptions of the objective function and constraints in a MP problem usually include some parameters. For example, in problems of resources allocation such parameters may represent economic parameters like costs of various types of production, labor costs requirements, shipment costs, etc. Clearly, the values of such parameters depend on multiple factors not included into the formulation of the problem. More representative model includes the corresponding complex relations into it and the model becomes more cumbersome and analytically unsolvable. Moreover, it can happen that such attempts to increase "the precision" of the model will be of no practical value due to the impossibility of measuring the parameters accurately. On the other hand, the model with some fixed values of its parameters may be too crude, since these values are often chosen in a quite arbitrary way. An intermediate approach is based on the introduction into the model the means of a more adequate representation of experts understanding of the nature of the parameters in the form of fuzzy sets of their possible values. The resultant model, although not taking into account many details of the real system in question could be a more adequate representation of the reality than that with more or less arbitrarily fixed values of the parameters. This leads to a new type of MP problems containing fuzzy parameters. Treating such problems requires the application of fuzzy-set-theoretic tools in a logically consistent manner. Such treatment forms an essence of fuzzy mathematical programming (FMP).

FMP problems are based on the straightforward use of the intersection of fuzzy sets representing goals and constraints and on the subsequent maximization of the resultant membership function. This approach has been mentioned by Bellman and Zadeh [69] FMP problem is denoted by:

$$\widetilde{\text{Maximize}}\ \tilde{f}(x, \tilde{c}) \qquad \ldots (5.4)$$

$$\text{subject to } \tilde{g}_i(x, \tilde{a}_i)\, \tilde{R}_i\, \tilde{b}_i,\ i \in M = (1, \ldots\ldots m) \qquad \ldots (5.5)$$

where $\tilde{R}_i$, $i \in M$, are fuzzy relations on $\mathcal{F}(\mathbf{R})$, the set of all fuzzy subsets of R. Parameters c, a_i and b_i are fuzzy parameters and are denoted with upper wavelet. Here function $\tilde{f}$ and $\tilde{g}$ may be linear or nonlinear functions of x.

Various methods have been proposed by researchers to solve FMP problems. One of these methods which is known as Multiple Fuzzy Reasoning (MFR) scheme is as follows. The solution process by MFR scheme consists of two steps [70]: first, for every decision variable $x \in R^n$ the value of the objective function, *MAX*(*x*), via sup-min convolution of the antecedents/constraints and the fact/objective is computed, then an optimal solution to the FMP problem is any point which produces a maximal element of the set $\{MAX(x) \mid x \in R^n\}$. This in a mathematical form is as follows:

Antecedent 1	Constraint$_1$(x) := $\tilde{g}_1(x, \tilde{a}_1)\, \tilde{R}_1\, \tilde{b}_1$	
...	... ...	
...	... ...	
Antecedent m	Constraint$_m$(x) := $\tilde{g}_m(x, \tilde{a}_m)\, \tilde{R}_m\, \tilde{b}_m$	... (5.6)
Fact	Goal (x) := $\tilde{f}(x, \tilde{c})$	... (5.7)
Consequence	*MAX*(*x*)	... (5.8)

where $x \in R^n$ and the consequence *MAX(x)* which is the value of the objective function subject to constraints at *x* is computed as follows.

$$MAX(x) = \text{Goal}(x) \cdot \bigcap_{i=1}^{m} \text{constraint}_i(x) \qquad \text{... (5.9)}$$

$$\mu_{MAX(x)}(v) = \sup \min \{\mu_{Goal(x)}(u), \mu_{Constraint_1(x)}(u,v), \ldots, \mu_{Constraint_m(x)}(u,v)\} \qquad \text{... (5.10)}$$

Then an optimal value of the objective function of the above optimization problem, denoted by M, is defined as

$$M = \sup \{MAX(x) \mid x \in R^n\}. \qquad \text{... (5.11)}$$

The most important mathematical programming problems are those where the functions f and gi are linear. The traditional Linear Programming (LP) is an algebraic method used to solve sets of linear equations. The purpose of linear programming is to find optimal solutions for systems which are modeled by linear equations. In LP, sharp constraints combine to limit the space of feasible solutions to the linear problem being posed. The variable dimensions of the system being modeled assume the form of a vector. The objectives of a problem are also modeled with linear equations. The linearity of the constraints and the objectives enables straight-forward solution methods. Vertices of the solution space correspond to optimizing vectors. The vectors are optimizing in the sense that non-zero linear equations of the system variables, representing the objectives, achieve maximal values at the vertices of the feasible solution space.

The classical model of linear programming can be stated as:

$$\begin{aligned} &\text{maximize } f(x) = c^T x \\ &\text{such that } \; Ax \le b \\ &\qquad\qquad x \ge 0 \\ &\text{with } c, x \in R^n,\, b \in R^m,\, A \in R^{m \times n} \end{aligned} \qquad \text{... (5.12)}$$

Where all coefficients of *A, b,* and *c* are crisp numbers, $\le$ is meant in a crisp sense and "maximize" is a strict imperative. Sometimes the LP-decision has to be made in fuzzy environments as discussed below, quite a number of possible modifications of (5.12) exist.

Essentially, traditional optimal concepts such as LP model are based on unique solutions and existence of complete information. Fuzzy optimization's main aim is to find the "best" solution under incomplete information, i.e. imprecise information and/or vague resources limits. The decision maker may not actually want to maximize or minimize the objective function. Rather he might want to reach some aspiration levels which might not even be definable crisply. Thus he might want to "improve the present cost situation considerably", and so on. In classical linear programming where the violation of any single constraint by any amount renders the solution infeasible. The constraints might be vague in one of the following ways. The ≤ sign might not be meant in the strict mathematical sense, but smaller violations might well be acceptable. This can happen if the constraints represent aspiration levels as mentioned above, or if for instance, the constraints represent sensory requirements which cannot adequately be approximated by a crisp constraint. Of course, the coefficients of the vectors b or c or of the matrix A itself can have a fuzzy character, either because they are fuzzy in nature or because perception of them is fuzzy. Fuzzy linear programming offers a number of ways for all those types of vagueness. Fuzzy linear programming can be regarded as a *shade sensitive* version of linear programming and may be expressed as:

$$\text{maximize } c^T x \gtrsim z$$
$$\text{such that } Ax \leq b \text{ and}$$
$$x \geq 0 \quad \ldots (5.13)$$

where $\gtrsim$ denotes the fuzzified version of ≥ and has the linguistic interpretation "essentially greater than". It is fully symmetric with respect to the objective function and constraints.

The concepts of fuzzy goal (objective) and fuzzy constraints were first introduced by Bellman and Zadeh [69]. According to them a fuzzy decision can be viewed as the intersection of fuzzy goal(s) and the problem constraints - since they are all defined as fuzzy sets in the space of alternatives. The "optimal" decision is the point at which the intersection of fuzzy goal(s) and constraints take the maximum membership value. This method is usually called the max-min approach.

In general, the fuzzification of the linear programming model includes following six forms of imprecision:

(i) Maximization or minimization of the objective function subject to fuzzy constraints.

(ii) Consideration of fuzzy objective function with crisp constraint.

(iii) Consideration of fuzzy goal and fuzzy constraints, represented by the membership values of the goal and constraints inequalities satisfaction.

(iv) The parameters (coefficients) of the variables of the constraints are not known precisely. i.e. the coefficients are fuzzy numbers.

(v) The coefficients of the variables in the objective function are not known precisely i.e. coefficients are fuzzy numbers.

(vi) All possible combinations of uncertainty.

When fuzziness is considered in objective function and constraints, the design variable *x* can be obtained from a fuzzy domain *D*, such that the membership function μ_D for the fuzzy domain *D* can be obtained from the intersection of the fuzzy membership functions for objective function and constraints as follows:

$$\mu_D = \mu_W(x) \cap \left\{ \bigcap_{i=1,2,\ldots,m} \mu_{g_i}(x) \right\} \qquad \ldots (5.14)$$

where $\mu_W(x)$ and $\mu_{g_i}(x)$ are membership functions for objective function and the *i*th inequality design constraint, respectively.

From this fuzzy domain, the optimum solution x^* for the design variable *x* can be obtained by using the following max-min procedure (Bellman and Zadeh 1970 [69]):

$$\mu_D(x^*) = \text{maximize } \mu_D(x) \qquad \ldots (5.15)$$

where $\mu_D(x) = \min \left\{ \mu_W(x), \min_{i=1,2,\ldots,m} \mu_{g_i}(x) \right\}$

This max-min procedure can be solved by maximizing a scalar parameter λ, known as the overall satisfaction parameter in the following way (Zimmermann 1978 [71]) to find the vector of design variable *x* and the parameter λ such that:

maximize λ

subject to the constraints

$\lambda \le \mu_W(x)$

$\lambda \le \mu^u_{g_i}(x), i = 1, 2, \ldots, m$

$\lambda \le \mu^l_{g_i}(x), i = 1, 2, \ldots, m$

$$0 \le \lambda \le 1 \qquad \ldots (5.16)$$

where $\mu^{u}_{g_i}(x)$ and $\mu^{l}_{g_i}(x)$ are membership functions for the upper and lower bounds of the inequality constraints $g_i(x)$ respectively. Above fuzzy optimization problem can be solved by any of the available soft computing tools.

5.4 α – CUT METHOD FOR OPTIMIZATION

α – cut method for optimization consists of development of input fuzzy set, induced fuzzy set and performance fuzzy set and superimposition of induced and performance fuzzy sets to arrive at an optimal solution. To illustrate the steps of getting optimum design based on fuzzy logic's α – cut method, design of a simply supported rectangular beam with following data is considered here.

Span	= 5 m	Grade of concrete	= M20
Super-imposed load	= 40 kN/m	Grade of steel	= Fe415

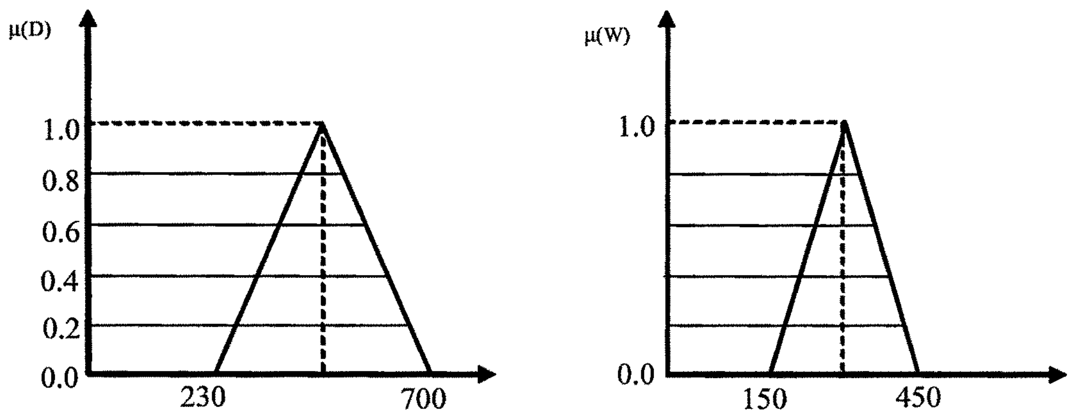
Step 1: Decide design parameters

In optimum design based on fuzzy logic, the first step is to decide design parameters such as pre-assigned, input, induced fuzzy set and performance parameters. The parameters such as the span of beam, load data, and material properties i.e. grade of concrete and grade of steel, whose values are provided before starting the design process, are called as pre-assigned parameters. Here optimum design of beam means optimization of cross-section dimensions i.e. width and depth of the beam which are considered as input parameters. Further in design of beam, compressive stress in concrete should not be more than the permissible compressive stress in concrete. Thus, the actual compressive stress in concrete and permissible compressive stress in concrete is considered as induced and performance parameter respectively.

Step 2: Fuzzify input parameters

After deciding the design parameters, the next step is to express input parameters (width and depth of beam) by appropriate fuzzy sets. For this, the minimum, maximum and accepted values of input parameters are required. The approximate values of depth and width required can be decided as follows. Generally, the depth of beam is assumed as 1/12 to 1/10 of span, i.e., 415 to 500. Thus the approximate value of depth required can be considered as 450 mm. The user may consider the region of search between 230 to 700 mm. These values will take membership grade value equal to zero. For simplicity, the average of these values is considered as accepted value i.e., 465 mm and assigned membership grade equal to one. Figure 5.4(a) shows the input fuzzy set for depth of beam depending upon selected values.

Usually the width of beam = 0.5 to 1.0 times the depth of the beam. The width of beam is 230 mm in most of the cases. Thus, this value can be considered as acceptable value of width. Here one can consider the region for input fuzzy set between 150 to 450 mm. These values will take the membership grade value equal to zero. The average value (i.e. 300 mm) is considered as accepted value of width that will take membership grade value equal to one. Figure 5.4(b) shows the input fuzzy set for width of beam.



(a) Depth of Beam (mm) **(b) Width of Beam (mm)**

Fig. 5.4 Input Fuzzy Sets

Table 5.1 Min and Max Values of Width and Depth

Membership Function	W1 (mm)	W2 (mm)	D1 (mm)	D2 (mm)
0.00	150. 0	450. 0	230. 0	700. 0
0.10	165. 0	435. 0	253. 5	676. 5
0.20	180. 0	420. 0	277. 0	653. 0
0.30	195. 0	405. 0	300. 5	629. 5
0.40	210. 0	390. 0	324. 0	606. 0
0.50	225. 0	375. 0	347. 5	582. 5
0.60	240. 0	360. 0	371. 0	559. 0
0.70	255. 0	345. 0	394. 5	535. 5
0.80	270. 0	330. 0	418. 0	512. 0
0.90	285. 0	315. 0	441. 5	488. 5
1.00	300. 0	300. 0	465. 0	465. 0

Once the fuzzy sets expressing preferences of the designer are determined for all input parameters, they are employed for calculating the associated induced fuzzy sets parameters. For this, one has to select appropriate values of α- cuts, preferably equally spaced, such as 0.05, 0.10, 0.15, 0.20,......, 1.0. Let the α- cut interval be 0.10 for this example. At each α-cut, the minimum and maximum values of width and depth are determined. These values are termed as crossover points, which are shown in Table 5.1.

Step 3: Develop performance fuzzy set

As mentioned earlier the permissible compressive stress in concrete is considered as performance parameter. Its value depends upon the grade of concrete. Here permissible compressive stress in concrete is taken as 7.5 N/mm^2 which will take membership grade equal to one. If tolerance in constraint violation is 15 %, then in no case the performance parameter will have value greater than 8.92 N/mm^2. Thus it will take membership grade value equal to zero. Figure 5.5 shows the variation of performance parameter with its membership grade whereas Table 5.2 shows the value of performance parameter at each α-cut.

Step 4: Develop induced fuzzy set

The actual compressive stress in concrete is considered as induced fuzzy set parameter. When input fuzzy sets are subjected to α-cuts, the four crossover points obtained are shown in Table 5.1. Four combinations are generated using these crossover point values i.e., (W1-D1),

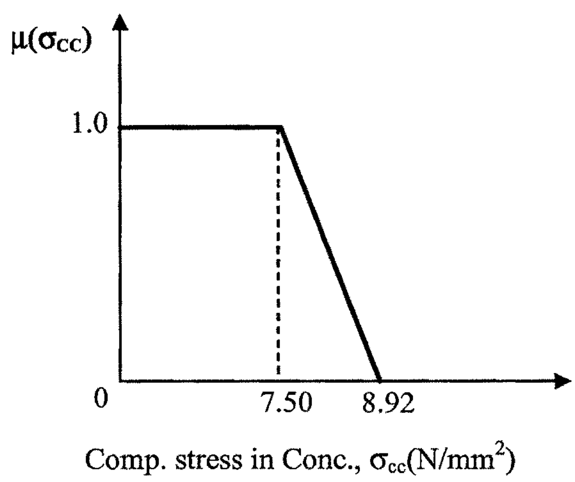


Fig. 5.5 Performance Fuzzy Set

(W1-D2), (W2-D1), and (W2-D2). For each combination, the actual value of compressive stress in concrete is computed and plotted against respective membership grade value. Table 5.2 shows stress value for each combination at α-cut interval of 0.1.

Table 5.2 Compressive Stress Values at Each α-Cut

M. F.	Induced Fuzzy Set Parameter for Different Combinations				Performance Parameter (Comp. Stress)
	(W1-D1)	(W1-D2)	(W2-D1)	(W2-D2)	
0. 00	73. 946	8. 658	26. 687	3. 555	8. 920
0. 10	55. 823	8. 488	22. 896	3. 865	8. 778
0. 20	43. 266	8. 405	19. 993	4. 218	8. 636
0. 30	34. 285	8. 396	17. 721	4. 622	8. 494
0. 40	27. 686	8. 454	15. 910	5. 088	8. 352
0. 50	22. 726	8. 575	14. 445	5. 628	8. 210
0. 60	18. 923	8. 758	13. 247	6. 258	8. 068
0. 70	15. 956	9. 005	12. 259	7. 000	7. 926
0. 80	13. 606	9. 321	11. 438	7. 876	7. 784
0. 90	11. 718	9. 710	10. 754	8. 922	7. 642
1. 00	10. 184	10. 184	10. 184	10. 184	7. 500

Step 5: Superimposition of two plots

After getting the performance and induced fuzzy sets, the next step is the superimposing of these plots to obtain the match (intersection) point. As shown in Fig.5.6 there are two match points. In real problem exact match point may not be always available. One way to do this is to find out difference of performance parameter value and compressive stress value at each α-cut. The point which gives minimum absolute difference and maximum performance can be considered as the match point. For the problem under consideration the point corresponding to α-cut value of 0.8 gives minimum difference for W2-D2 combination i.e. 7.876 - 7.784 = 0.092. Thus this point is the match point

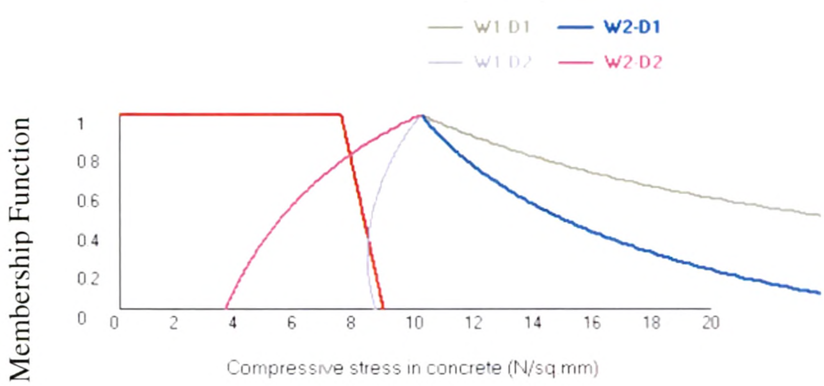


Fig 5.6 Superimposition of Performance and Induced Fuzzy Sets

5.5 FUZZY RULE BASED SYSTEM

Fuzzy logic with fuzzy rules has the potential to add human-like subjective reasoning capabilities to machine intelligences, which are usually based on bivalent Boolean logic. Rule base is a very important state in fuzzy logic. Fuzzy rules take into account essential qualitative information which is ignored in conventional control. Conventional rules contain conditions and control actions. Fuzzy rule base specifies the way Linguistic variables and Adjectives act between each other and later on the defuzzification will be computed. Conditional statements are used in order to make complete sentences. Fuzzy rules are linguistic IF-THEN- constructions and gives the ability to deal with two kinds of rules: Atomic rules and Compounded rules.

Atomic rule takes the shape of: If.... Then....., and

Syntax: *If (x is A) Then (y is B)*

where A and B are linguistic values defined by fuzzy sets on the ranges X and Y (universes of discourses). The 'If' part is called *antecedent* while 'Then' part is called *consequent* or conclusion.

Compounded Rule will look like: If.... (And/Or).... Then.... (And/Or)....

Both *antecedent* and *consequent* rules can have multiple parts.

Syntax: *"IF temperature is warm AND pressure is increasing THEN sky is grey AND wind is strong"*

In effect, the use of linguistic variables and fuzzy If-Then rules exploits the tolerance for imprecision and uncertainty. In this respect, fuzzy logic mimics the crucial ability of the human mind to summarize data and focus on decision relevant information. Several rules constitute a *fuzzy rule-based system*. Fuzzy rules can be designed manually by a user, it means that designer generates rules for all combinations of selected premise variable and a user fills consequent fuzzy terms. A typical fuzzy system is made up of three major components: fuzzifier, fuzzy inference engine (fuzzy rules) and defuzzifier.

5.5.1 Fuzzification of Input Data

Fuzzification [72] is the process of making the crisp quantity "fuzzy". It means spreading the information provided by a crisp number to its vicinity so that the close neighborhood of the crisp number can be recognized i.e. it allows addressing uncertainty in any parameter due to imprecision, ambiguity or vagueness. The concept of fuzzification enables to convert the linguistic variable into a crisp quantity. In general, for fuzzification of any linguistic variable, first of all parameters affecting it are required to be found out. Then importance factor is given to the each parameter i.e. scaling is required and fuzzy set is prepared for that particular variable considering suitable shape of MF and based on the scaled value of the linguistic variable the confidence interval is found out. This process is called as fuzzification.

5.5.2 Fuzzy Inference Engine

Fuzzy inference engine forms the mapping between the input membership functions and the output membership functions using fuzzy rules that can be obtained from expert knowledge of the relationships being modeled. The greater the input membership degree, the stronger the rule fires, thus the stronger the pull towards the output membership function. Since several different output membership functions can be contained in the consequents of rules triggered, a defuzzifier carries out the defuzzification process to combine the output into a single label or numerical value as required.

5.5.3 Defuzzification of Output Data

Defuzzification [10] is an operation reverse to fuzzification. The process of converting the fuzzy output into a crisp value is called as defuzzification. The set of rules is applied to the fuzzified input and the output of each rule is fuzzy. The output of the rule base can be the union of a two or more fuzzy membership functions defined at the universe of discourse of the output variable. These fuzzy outputs need to be converted into a scalar quantity so that the control action can be taken by the system. Defuzzification is especially necessary for hardware applications, because conventional systems operate based on crisp data exchange. To select a single value from the fuzzy set which controls the functioning of system is really a very difficult task.

Criteria for selection of defuzzification technique are,

- *Continuity*: A small change in input shall not produce large change in output.
- *Disambiguity*: Any defuzzification shall usually result in a unique value.
- *Plausibility*: It must have a high membership value and lie approximately in the middle.
- *Computational simplicity*: The longer it takes to compute, the less valuable is the method.

❖ *Weighting method*: The best method to weigh the different output fuzzy sets.

5.6 DEFUZZIFICATION TECHNIQUES

5.6.1 Center of Gravity Technique

This technique was developed by Sugeno in 1985 [73] and is also known as Centroid or Center of area defuzzification technique, which is analogous to finding the balance point by calculating the weighted mean of the fuzzy output. It is the most commonly used technique and is very accurate. The Centroid defuzzification technique can be expressed as,

$$z^* = \frac{\int \mu\tilde{A}(z) \,.\, z \, dz}{\int \mu\tilde{A}(z) \, dz} \qquad \ldots (5.17)$$

where z^* is the defuzzified output, $\tilde{A}$ is the fuzzy set, $\int$ denotes an algebraic integration, and z represents the membership strength at a particular point.

The only disadvantage of this method is that it is computationally difficult for complex membership functions.

5.6.2 Leftmost-Rightmost Maxima Technique

This is very simple method. Here, leftmost or rightmost maxima value is taken as the strength of membership function and as per that output is calculated. The comparison of Centroid and Leftmost-rightmost maxima method is shown in Fig. 5.7.

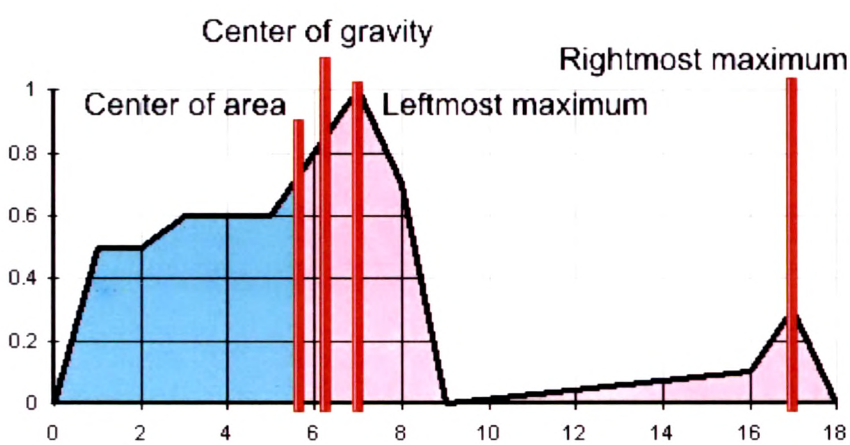


Fig. 5.7 Comparison of Centroid and Leftmost-Rightmost Technique

5.6.3 Weighted Average Defuzzification Technique

In this method the output is obtained by the weighted average of the each output of the set of rules stored in the knowledge base of the system. This method is computationally faster and easier and gives fairly accurate result. But it is only valid for symmetrical output membership functions as shown in Fig. 5.8. The Weighted average defuzzification technique can be expressed as,

$$z^* = \frac{\Sigma\ \mu\tilde{A}(z)\ .\ z}{\Sigma\ \mu\tilde{A}(z)} \qquad \ldots (5.18)$$

5.6.4 Maximum Membership Defuzzification Technique

This method gives the output with the highest membership function. It is very fast but only accurate for peaked output. The method is depicted in Fig. 5.9. This technique is given by algebraic expression as,

$$\mu\tilde{A}(z^*) \geq \mu\tilde{A}(z) \text{ for all } z \in Z. \qquad \ldots (5.19)$$

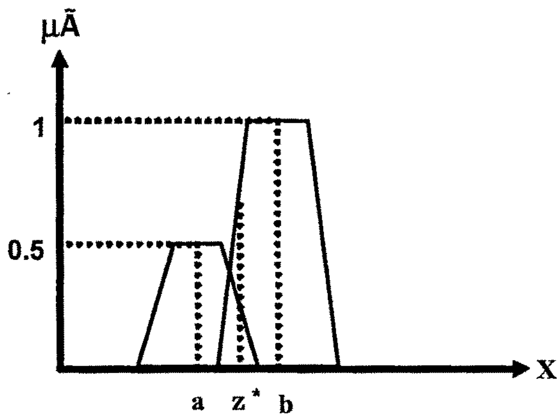


Fig. 5.8 Weighted Average Technique

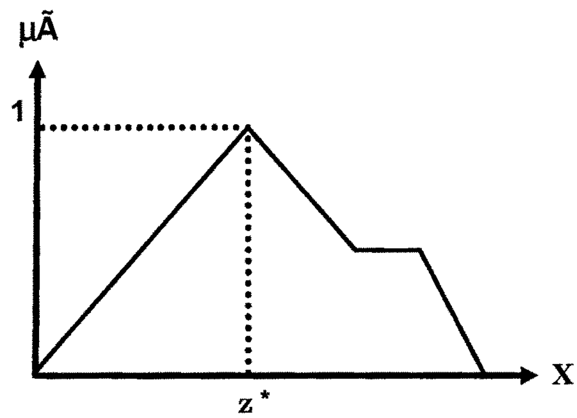


Fig. 5.9 Maximum Membership Technique

5.6.5 Mean-Max Membership Function Technique

This method is closely related to the maximum membership defuzzification technique, except that the locations of the maximum membership can be non-unique. The method is depicted in Fig. 5.10. This technique is given by algebraic expression as,

$$Z^* = (a + b) / 2 \qquad \ldots (5.20)$$

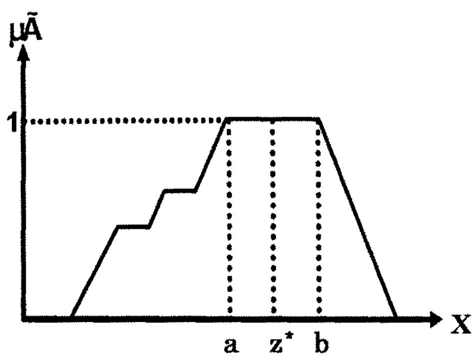


Fig. 5.10 Mean-Max Membership Function Technique

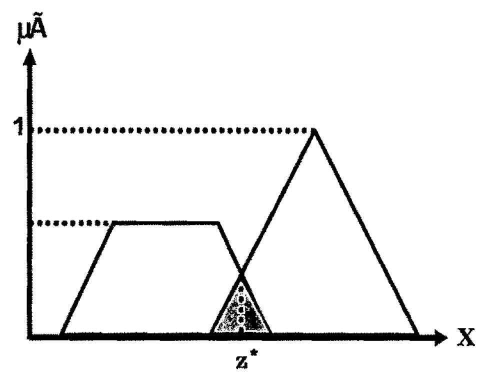


Fig. 5.11 Center of Sums Technique

5.6.6 Center of Sums Technique

This technique is faster than many defuzzification methods that are presently in use. This process involves the algebraic sum of individual output fuzzy sets, instead of their union. One drawback to this method is that the intersecting areas are added twice. Figure 5.11 is an illustration of the method. The defuzzified value z is given by the equations,

$$z^* = \frac{\int_z z \sum_{k=1}^{n} \mu\tilde{A}_k(z)\, dz}{\int_z \sum_{k=1}^{n} \mu\tilde{A}_k(z)\, dz} \qquad \ldots (5.21)$$

5.6.7 Center of Largest Area Technique

If the output fuzzy set has at least two convex sub regions, then the center of gravity of the convex fuzzy sub region with the largest area is used to obtain the defuzzified value z* of the output. This is shown graphically in Fig. 5.12 and given algebraically as,

$$z^* = \frac{\int \mu\tilde{A}m(z)\ z\, dz}{\int \mu\tilde{A}m\,(z)\, dz} \qquad \ldots (5.22)$$

where Ãm is the convex sub-region that has the largest area.

5.6.8 First of Maxima Techniques

This method uses the overall output or union of all individual output fuzzy sets c to determine the smallest value of the domain with maximized membership degree in c. The method is graphically explained in Fig. 5.13 and the equations for z * are obtained as,

First, the largest height in the union [denoted $hgt(x_k)$] is determined,

$$hgt(x_k) = \sup_{z \in Z} \mu\tilde{A}_k(z) \qquad \text{... (5.23)}$$

Then the first of the maxima is found,

$$z^* = \inf_{z \in Z} \{ z \in Z \mid \mu\tilde{A}_k(z) = hgt(x_k) \} \qquad \text{... (5.24)}$$

An alternative to this method is called the last of maxima and it is given by,

$$z^* = \sup_{z \in Z} \{ z \in Z \mid \mu\tilde{A}_k(z) = hgt(x_k) \} \qquad \text{... (5.25)}$$

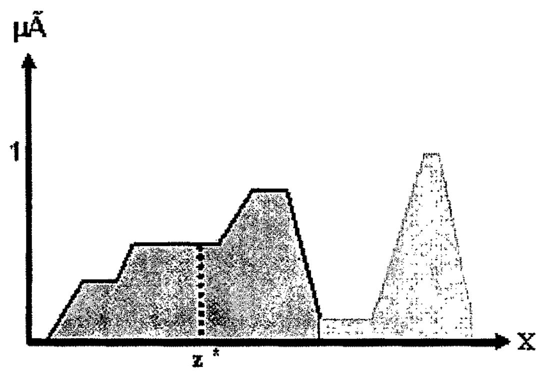


Fig. 5.12 Center of Largest Area Technique

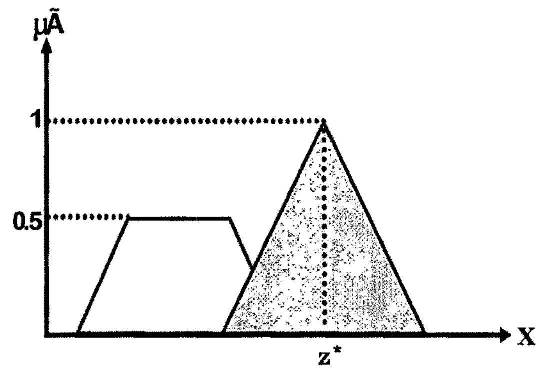


Fig. 5.13 First of Maxima Technique

5.7 FUZZY CONTROLLER

Control theory constitutes a classical topic in mathematics and engineering. The purpose of control is to influence the behavior of a system by changing an input or inputs to that system according to a rule or set of rules that model how the system operates. A typical Fuzzy controller consists of four modules (Fig. 5.14): the rule base, the inference engine, the fuzzification, and the defuzzification. A typical Fuzzy Control algorithm would proceed as follows:

1. ***Obtaining information***: Collect measurements of all relevant variables.
2. ***Fuzzification***: Convert the obtained measurements into appropriate fuzzy sets to capture the uncertainties in the measurements.
3. ***Running the Inference Engine***: Use the fuzzified measurements to evaluate the control rules in the rule base and select the set of possible actions.
4. ***Defuzzification***: Convert the set of possible actions into a single numerical value.
5. ***The Loop***: Go to step one.

Several defuzzification techniques have been devised. The most common defuzzification methods are: the center of gravity, the center of maxima and the mean of maxima.

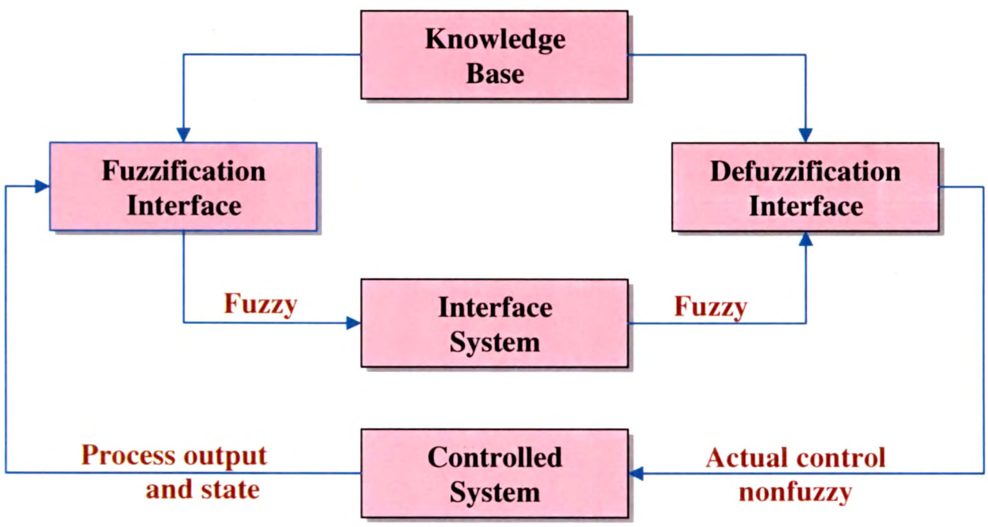


Fig. 5.14 Block Diagram of Fuzzy Controller