1.1 INTRODUCTION

In fuzzy logic, the truth of any statement is a matter of degree. Any statement can be fuzzy. A membership function is the curve that defines how true a given statement is for a given input value. It defines how each point in the input space is mapped to a membership value (or degree of membership) from 0 and 1.

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth-truth values between "completely true" and "completely false". Fuzzy logic extends Boolean logic to handle the expression of vague concepts, and to express imprecision in a quantitative fashion. It introduces a set of membership functions that maps elements to real values between ‘0’ (Zero) and ‘1’ (One), the value indicates the "degree" to which an element belongs to a set. A membership value of zero indicates that the element is entirely outside the set, whereas a one indicates that the element lies entirely inside a given set. Any value between the two extremes indicates a degree of partial membership to the set. Hence fuzzy logic allows imprecise and qualitative information to be presented in quantitative way. It offers a rigorous and practical technique for manipulating qualitative information originally expressed in linguistic form.

This chapter gives an overview on origin and evolution of fuzzy theory, introduction to fuzzy set theory, and some basic control strategies (actions) including proportional plus integral plus derivative (PID) controller, fuzzy logic controller and integrated fuzzy logic controllers. It also discusses the significance and role of computers in the field of process instrumentation.

1.2 HISTORY AND EVOLUTION OF FUZZY THEORY

1.2.1 The Birth of Fuzzy Logic and Theory

Hitherto mid twentieth century, the fuzzy set theory was not known to the world. The idea of fuzzy sets was born in July 1964. Lotfi A. Zadeh, a well-respected professor in the department of electrical engineering and computer science at University of California, Berkeley. In the fifties, he believed that all real-world problems could be solved with
efficient, analytical methods and/or fast (and big) electronic computers. In this direction, he has made significant contributions in the development of system theory (e.g., the state variable approach to the solution of simultaneous differential equations) and computer science. In early 1960s, however, he began to feel that traditional system analysis techniques were too precise for many complex real-world problems. In a paper written in 1961, he mentioned that a different kind of mathematics was needed:

*We need a radically different kind of mathematics, the mathematics of fuzzy or cloudy quantities, which are not described in terms of probability distributions. Indeed, the need for such mathematics is becoming increasingly apparent, for in most practical cases the priori data as well as the criteria by which the performance of a man-made system is judged are far from being precisely specified or having accurately known probability distributions.*

1.2.2 A Decade of Theory Development (1965-1975)

Even though there was strong resistance to fuzzy logic, many researchers around the world became Zadeh’s followers. While Zadeh continued to broaden the foundation of fuzzy set theory, scholars and scientists in a wide variety of fields – ranging from psychology, sociology, philosophy, and economics to natural sciences and engineering - were exploring this new paradigm during the first decade after the birth of fuzzy set theory. Important concepts introduced by Zadeh during this period include fuzzy multistage decision-making, fuzzy similarity relations, fuzzy restrictions, and linguistic hedges. Other contributions include R. E. Bellman’s work (with Zadeh) on fuzzy multistage decision making [1], G. Lakoff’s work from a linguistic view [2], J. A. Goguen’s work on the category-theoretic approach to fuzzify mathematical structure [3,4], L. J. Kohout and B. R. Gains on the foundation of fuzzy logic [5, 6], the work on fuzzy measures by R. E. Smith and M. Sugeno [7-10], G. J. Klir, Sols and Meseguer’s work on fuzzified algebraic and topological systems [11,12], C. L. Chang’s work on fuzzy topology [13], Dunn and J. C. Bezdek’s work on fuzzy clustering [14], C. V. Negoita’s work on fuzzy information retrieval [15-17], the work by M. Mizumoto and K. Tanaka on fuzzy automata and fuzzy grammars [18-21], A. Kandel’s work on fuzzy switching function [22,23], and H. J. Zimmermann’s work on fuzzy optimization [24].
During the first decade, many mathematical structures were fuzzified by generalizing the underlying sets to be fuzzy. These structures include logics, relations, functions, graphs, groups, automata, grammars, languages, algorithms, and programs. One of the two early fuzzy logic journals in the world is actually a Chinese journal on fuzzy mathematics.

In the late 1970s, a few small university research groups on fuzzy logic were established in Japan. Professor T. Terano and Professor H. Shibata from Tokyo University led one such group in Tokyo. A second research group in the Kansai area was led by Professor K. Tanaka from Osaka University and Professor K. Asai from the University of Osaka Prefecture. These researchers encountered an “anti-fuzzy” atmosphere in Japan during those early days. However their persistence and hard work proved to be worthwhile a decade later. These Japanese researchers, their students, and the students of their students made many important contributions to the theory as well as to the applications of fuzzy logic.

An important milestone in the history of fuzzy logic control was established by Assilian and E. Mamdani in the United Kingdom in 1974. They developed the first fuzzy logic controller, which was used for controlling a steam generator. They were initially comparing learning algorithms for adaptive control of a nonlinear, multidimensional plant for a physical steam engine but found that many learning schemes failed to even begin to converge on a reasonable time scale (running out of steam!). A fuzzy linguistic method was developed to prime the learning controller with an initial policy to speed the adaptation - the verbal statements of engineers were transcribed as fuzzy rules and used under fuzzy logic to form a control policy. The performance of these fuzzy linguistic controllers was so good in their own right, however, that they became central to a range of studies that subsequently took place. In the earlier 1975’s, E. Mamdani and Baaklini showed that fuzzy control rules may be tuned automatically by fuzzy linguistic adaptive strategies. Automatic learning and tuning of fuzzy rules proved to be a very important area in the next two decades.

Pioneering efforts to use fuzzy logic applications in civil engineering were made by C. B. Brown, D. Blockley, D. Dubois. In April 1971, C. Brown and R. Leonard [25] introduced and discussed the civil engineering applications of fuzzy sets during the ASCE Structural Engineering Meeting in Baltimore, Maryland. In 1975, D. Blockley [26] published a paper on the likelihood of structural accidents, which was followed by a continuous flow of stimulating papers [27,28] and a thought-provoking book [29]. In 1979, C. Brown [30] presented a fuzzy safety measure, with which more realistic failure rates were obtained by
utilizing both subjective information and objective calculations. Later, Brown treated entropy constructed probabilities [31].

In 1977, Dubois applied fuzzy sets in a comprehensive study of traffic conditions [32,33]. The general problem of uncertainty and fuzziness in engineering decision-making was discussed in a comprehensive manner by J. Munro [34]. J. Yao summarized the development of civil engineering applications of fuzzy sets during the seventies in a 1985 NSF workshop on civil engineering applications of fuzzy sets held at Purdue University.

One way to get an overall picture of the growth of fuzzy logic during the first decade is to study the number of papers published on the subject. Based on a survey conducted by B. Gaines, the number of papers increased by about 40 percent annually during the mid 1970s. Works accomplished during this period established the foundation of fuzzy logic technology and led to the development of application of the technology in the subsequent years.

1.2.3 Pioneers of Industrial Applications (1976 – 1987)

In 1976, the first industrial application of fuzzy logic was developed by Blue Circle Cement and SIRA in Denmark. The system is a cement kiln controller that incorporates the “know-how” of experienced operators to enhance the efficiency of a clinker through smoother grinding. The system went to operation in 1982.

After eight years of persistent research, development, and deployment efforts, Seiji Yasunobu and his colleagues at Hitachi put a fuzzy logic-based automatic train operation control system into operation in Sendai city’s subway system in 1987. Another early successful industrial application of fuzzy logic is a water-treatment system developed by Fuji Electric. We should make a few important points about these applications. First, after a successful demonstration of these approaches, it took years for both projects to be deployed in real world operation due to various concerns about this new technology from government officials. The roads traveled by these engineers were not easy at all. Second, these two applications became the major “success stories” of fuzzy logic technology in Japan. Consequently, many more Japanese engineers and companies started to investigate fuzzy logic applications. Third, both applications made significant contributions to the technology of fuzzy logic. The development of the Sundai subway system introduced an interesting
architecture for using fuzzy logic for predictive control. It also used fuzzy logic together with mathematical modeling. The former for recommending control options and for evaluating control options, and the latter for simulating control options to predict their effects. The development of water treatment system enabled Fuji Electric to introduce the first Japanese general-purpose fuzzy logic controller (named FRUITAX) into the market in 1985.

1.2.4 The Fuzzy Boom (1987- Present)

The fuzzy boom in Japan was a result of the close collaboration and technology transfer between universities and industries. In 1988, the Japanese government launched a careful feasibility study about establishing national research projects on fuzzy logic involving both universities and industry. Two large-scale national research projects were established by two agencies – the Ministry of International Trade and Industry (MITI) and the Science and Technology Agency (STA). The project established by MITI was a consortium called the Laboratory for International Fuzzy Engineering Research (LIFE), which involved 50 companies with a six-year total budget of US $5 billions.

Matsushita Electric Industrial Co. (also known as Panasonic outside Japan) was the first to apply fuzzy logic to a consumer product, a shower head that controlled water temperature, in 1987. In late January 1990, Matsushita Electric Industrial Co. named their newly developed fuzzy controlled automatic washing machine “Asai-go Day Fuzzy” and launched a major commercial campaign for the fuzzy product. This campaign turned out to be a successful marketing effort not only for the product, but also for the fuzzy logic technology. A foreign word pronounced “fuzzy” was thus Japan introduced a new meaning - intelligence. Many other home electronic companies followed Panasonic’s approach and introduced fuzzy vacuum cleaners, fuzzy rice cookers, fuzzy refrigerators, and others. This resulted in a fuzzy vogue in Japan.

This fuzzy boom in Japan triggered a broad and serious interest in technology in Europe, and to a lesser extent, in the United States, where fuzzy logic was invented. Several major European companies formed fuzzy logic task forces within their corporate Research and Development (R&D) divisions. They include SGS-Thomson of Italy, Siemens, Diamler-Benz, and Klockner-Moeller in Germany. In the United States, the General Electric
Corporate Research Division and Rockwell International Science Center have both developed advanced fuzzy logic technology as well as their industrial applications.

1.2.5 Tools for Fuzzy Control Applications and more Principled Design

Various tools have been used for implementing the fuzzy logic in many applications viz. microprocessors, microcontrollers, VLSI chips and computers. The first VLSI chip for performing fuzzy logic inferences developed by M. Togai and H. Watanabe in 1986 [35]. MATLAB is a high performance language of technical computing from Math Works Inc. The fuzzy logic toolbox for MATLAB was introduced as an add-on component to MATLAB in the year 1994.

The fuzzy systems in the early days required the manual tuning of the system parameters based on observing the system performance. This draw back has become one of the major criticisms of the fuzzy logic. Even though Mamdani and Baaklini introduced self-adaptive fuzzy logic control as early as 1975, the most common citation of the first work in this area is a paper by T. J. Proyak and E. H. Mamdani published in 1979. This was followed by Japanese researchers in the 1980s. T. Takagi and his advisor M. Sugeno together took an important step by developing the first approach for constructing (not tuning) fuzzy rules using training data for controlling a toy vehicle by observing how a human operator controlled the vehicle. This work laid the foundation for a popular subarea in fuzzy logic, which is now referred to as fuzzy model identification in the 1990s.

Another trend that contributed to research in the fuzzy model identification is the increasing visibility of neural network research in the late 1980s. Because of certain similarities between neural networks and fuzzy logic, researchers began to combine the two technologies. A system built in this way is called a neuro-fuzzy system. Bart Kosko, known for his contribution to neuro-fuzzy systems and he is the author of books (his was the first book on this topic) on neural networks [36]. The 1990s saw an era of new computational paradigms. In addition to fuzzy logic and neural networks, a third non-conventional computational paradigm has also become popular – evolutionary computing, which includes genetic algorithms, evolutionary strategies and programming. The various combinations of neural networks, genetic algorithms, and fuzzy logic help the people to view them as complementary.
1.3 FUZZY SET THEORY

1.3.1 Fuzzy Sets and Membership

The concept of fuzzy set and fuzzy logic were introduced by Professor Lotfi A. Zadeh, University of California, Berkeley. Zadeh's seminal work "Fuzzy Sets", which described the mathematics of fuzzy set theory, and by extension fuzzy logic [37,38]. His intention in introducing this fuzzy set theory was to deal with problems involving knowledge expressed in vague and linguistic terms.

Classically, a set is defined by its members. An object may be either a member or a non-member, which is the characteristic of traditional (crisp) set. The connected logical proposition may also be true or false. This concept of crisp set may be extended to fuzzy set with the introduction of the idea of partial truth. Any object may be a member of a set 'to some degree'; and a logical proposition may hold true 'to some degree'. Often, we communicate with other people by making qualitative statements, some of which are vague because we simply do not have the precise datum at our disposal e.g., a person is tall (we have no exact numerical value at that moment) or because the datum is not measurable in any scale e.g. a beautiful girl (for beautiful, no metric exists). Here, tall and beautiful are fuzzy sets. So, fuzzy concepts are one of the important channels by which we mediate and exchange information, ideas and understanding among ourselves. Fuzzy set theory offers a precise mathematical form to describe such fuzzy terms in the form of fuzzy sets of a linguistic variable.

Bivalent (Boolean) logic does not provide the means to identify an intermediate value. Fuzzy logic extends Boolean logic to handle the expression of vague concepts and to express imprecision in a quantitative fashion, it introduces a set of membership function that maps elements to real values between zero and one; the value indicates the "degree" to which an element belongs to a set. A membership value of zero indicates that the element is entirely outside the set, whereas a one indicates that the element lies entirely inside a given set. Any value between the two extremes indicates a degree of partial membership to the set. Consider an example that 100°F is a "hot" room temperature and 25°F is a "cold" room temperature. If bivalent logic is used to represent hotness of the room, 100°F would have a membership value of one and 25°F would have a membership value of 0. On the other hand, 75°F becomes much harder to classify the temperature as "hot" or "cold". If fuzzy logic is
used for the same, 100°F would have a membership value of one and 25°F would have a membership value of 0. On the other hand, 75°F would have a membership value between zero and one.

The crisp set is defined in such a way as to dichotomize the individuals in some given universe of discourse into members or non-members [39]. A sharp unambiguous distinction (boundary) exists between the members and non-members are as shown in Fig. 1.1 (a). From this figure it is clear that, point ‘a’ is obviously a member of crisp set A; point ‘b’ is unambiguously not a member of set A. A fuzzy set on the other hand, defined mathematically by assigning to each possible individual in the universe of discourse a value representing its grade of membership in the fuzzy set. The fuzzy set is prescribed by vague or ambiguous properties; hence their boundary is ambiguously specified, and is as shown in Fig. 1.1 (b) for a set Ā (Ā). In the central (inside the boundary) region of the fuzzy set, point ‘a’ is clearly a complete member of the set Ā. Outside the boundary region, point ‘b’ is clearly not a member of the fuzzy set. However, the membership of point ‘c’, which is on the boundary region, is ambiguous [40]. The membership of point ‘c’ must have some intermediate value on the interval zero and one. The membership of point ‘c’ approaches a value of 1 as it moves closer to the central region, and a value of 0 as it moves closer to ‘b’ leaving the boundary region of Ā is as shown in Figure 1.1 (b).

Hence the crisp sets have unambiguous boundaries and the transition for an element in the universe between memberships in a given set is abrupt and well defined [41]. The fuzzy set elements, satisfy imprecise properties of membership, and have varying degrees of membership in the set and these can be approximated to real values between one and zero.
The characteristic function of a crisp set is given as,

$$\mu_A(x) = \begin{cases} 
1 & \text{for } x \in A \\
0 & \text{for } x \notin A 
\end{cases}$$

where, $\mu_A(x)$ is the degree of membership.

The membership function of a fuzzy set is given as,

$$\mu_A(x) \rightarrow [0,1]$$

Classical sets contain the objects that satisfy precise properties of membership; fuzzy sets contain objects that satisfy imprecise properties of membership, i.e., membership of an object in a fuzzy set can be approximate. For example, the set of heights from 5 to 7 feet is crisp; the set of heights in the region around 6 feet is fuzzy. Fig 1.2 (a) and (b) show the membership function for the crisp and fuzzy sets.

Fuzzy sets can also be defined by assigning a continuous function to describe the membership either analytically or graphically. Various types of membership functions are illustrated in Fig. 1.3 and the most commonly used membership functions are triangular, bell-shaped and trapezoidal functions. The mathematical expressions for these membership functions are discussed in later sections [42].
Fig. 1.3 Different types of membership functions (i) Triangular (ii) Trapezoidal (iii) Gaussian (bell) (iv) T- function (v) L- function (vi) S- function and (vii) Singleton

1.3.2 Operations on Fuzzy Sets

Fuzzy sets are the tools, which convert the concept of fuzzy logic into algorithms leading to applications. Hence fuzzy operations are useful in expressing ideas of fuzzy logic leading to applications such as fuzzy controllers. The representation or form of a fuzzy set is,

\[ A = \{ x, \mu_A(x) | x \in X \} \]

where, \( X \) is the universe of discourse,
x represents elements in the universe X

$\mu_A(x)$ is the grade of membership in the range from 0 to 1.

The standard fuzzy set operations [43,44], most commonly used in engineering applications are

**a. Fuzzy Union (OR)**

The union of two fuzzy sets $A$ and $B$ in the universe $X$ is a fuzzy set, whose membership function is defined as

$$\mu_{A \cup B}(x) = \mu_A(x) \cup \mu_B(x)$$

(or)

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$$

Example,

Let $A = \{0.7/1, 0.5/2, 0.3/3, 0.0/4, 0.8/5, 1.0/6\}$

$B = \{0.3/1, 0.6/2, 0.2/3, 0.9/4, 0.1/5, 0.4/6\}$

Then, $A \cup B = \{0.7/1, 0.6/2, 0.3/3, 0.9/4, 0.8/5, 1.0/6\}$

**b. Fuzzy Intersection (AND)**

The intersection of two fuzzy sets $A$ and $B$ in the universe $X$ is a fuzzy set, whose membership function is defined as

$$\mu_{A \cap B}(x) = \mu_A(x) \cap \mu_B(x)$$

(or)

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$$

Example,

Let $A = \{0.7/1, 0.5/2, 0.3/3, 0.0/4, 0.8/5, 1.0/6\}$ and

$B = \{0.3/1, 0.6/2, 0.2/3, 0.9/4, 0.1/5, 0.4/6\}$

Then, $A \cap B = \{0.3/1, 0.5/2, 0.2/3, 0.0/4, 0.1/5, 0.4/6\}$

**c. Fuzzy Complement**

The absolute complement of a fuzzy set $A$ in the universe $X$ is denoted by $\overline{A}$ and its membership function is defined by

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \text{ for all } x \in X$$

Example,

Let $A = \{0.6/1, 0.3/2, 0.0/3, 0.1/4, 0.8/5, 1.0/6\}$

Then, $\overline{A} = \{0.4/1, 0.7/2, 1.0/3, 0.9/4, 0.2/5, 0.0/6\}$
Venn diagrams for these three operations in terms of membership function (triangular and trapezoidal), extended to consider fuzzy sets are shown in Fig. 1.4. Any fuzzy set \( \mathcal{A} \) is defined on a universe \( X \) is a subset of that universe

\[ \mathcal{A} \subseteq X \Rightarrow \mu_{\mathcal{A}}(x) \leq \mu_{X}(x) \]

For all \( x \in X, \mu_{\mathcal{A}}(x) = 0 \)

For all \( x \in X, \mu_{X}(x) = 1 \)
De Morgan’s laws hold good for fuzzy sets, as denoted by these expressions

\[
\overline{A \cup B} = \overline{A} \cap \overline{B} \\
\overline{A \cap B} = \overline{A} \cup \overline{B}
\]

Fig. 1.5 Extended graphical illustration of excluded middle laws for fuzzy sets
(i) \( A \cup \overline{A} \neq X \) (law of excluded middle law) (ii) \( A \cap \overline{A} \neq \emptyset \) (law of contradiction)

All other operations on classical sets also hold for fuzzy sets, except for the excluded middle laws. These two laws do not hold for fuzzy sets; since fuzzy sets can overlap, a set and its complement can also overlap. The excluded middle laws, extended for fuzzy sets are expressed by

\[
\begin{align*}
\text{Law of excluded middle,} & \quad \overline{A \cup \overline{A}} \neq X \\
\text{Law of contradiction,} & \quad \overline{A \cap \overline{A}} \neq \emptyset
\end{align*}
\]

Extended graphical illustrations comparing the excluded middle laws for fuzzy sets are shown in Fig. 1.5.

1.3.3 Fuzzy Arithmetic and Algebraic Operations

The heart of a fuzzy logic controller is its inference engine. The knowledge about fuzzy interval-valued arithmetic and algebraic operations on fuzzy sets is more useful to form the
fuzzy inference. Hence, some of arithmetic operations on intervals and algebraic operations on fuzzy sets are performed and discussed here with the suitable examples [39,43,45].

**Arithmetic Operations**

Let * denote any of the four arithmetic operations on closed intervals such as addition (+), subtraction (-), multiplication (.), and division (/). Then,

\[
[a, b] * [d, e] = \{ f * g | a \leq f \leq b, d \leq g \leq e \}
\]

is a general property of all arithmetic operations on closed intervals, except that \([a, b]/[d, e]\) is not defined when \(0 \in [d, e]\). That is, the result of an arithmetic operation on closed intervals is again a closed interval.

The four arithmetic operations on closed intervals are defined as follows,

**Addition** of two numbers is given as, \([a, b] + [d, e] = [a + d, b + e]\)

**Example:**

i) \([4, 5] + [3, 1] = [4 + 3, 5 + 2] = [7, 6]\)

ii) \([0, 5] + [-2, 3] = [0 + (-2), 5 + 3] = [-2, 8]\)

**Subtraction** of two numbers is given as, \([a, b] - [d, e] = [a - e, b - d]\)

**Example:**

i) \([4, 5] - [3, 7] = [4 - 7, 5 - 3] = [-3, 2]\)

ii) \([0, 5] - [-2, 3] = [0 - 3, 5 - (-2)] = [-3, 7]\)

**Multiplication** of two numbers is given as,

\([a, b] \cdot [d, e] = \{\min (ad, ae, bd, be), \max (ad, ae, bd, be)\}\)

**Example:**

i) \([2, 1] \cdot [-4, 3] = \{\min (-8, 4, -2, 6), \max (-8, 4, -2, 6)\} = [-8, 6]\)

ii) \([-1, 3] \cdot [-1, -2] = \{\min (1, 2, -3, -6), \max (1, 2, -3, -6)\} = [-6, 2]\)

**Division** of two numbers with the condition \(0 \notin [d, e]\) is given as,

\([a, b] / [d, e] = [a, b] \cdot [1/e, 1/d]\)

**Example:**

i) \([1, -2] / [4, 0.5] = \{\min (1/4, 0.5, -2/4, -2/0.5), \max (1/4, 0.5, -2/4, -2/0.5)\} = [-4, 2]\)

ii) \([-2, 6] / [1, 2] = \{\min (-2/1, -2/2, 6/1, 6/2), \max (-2/1, -2/2, 6/1, 6/2)\} = [-2, 6]\)
Algebraic Operations

The algebraic operations on fuzzy sets are given as,

a. The algebraic sum

The sum (addition) of two fuzzy sets $A$ and $B$ is a fuzzy set, $C = A + B$, with a membership function given by

$$
\mu_C(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \mu_B(x)
$$

where, the ‘+’ symbol here represents the algebraic summation.

Example:

Let $A = \{0.4/2, 0.3/4, 0.6/5\}$, and $B = \{0.1/2, 0.5/4, 0.7/8\}$

Then, $C = A + B$ would consist four members: 2, 4, 5, and 8 with the membership functions $\mu_C(x_1), \mu_C(x_2), \mu_C(x_3), \mu_C(x_4)$ respectively they can be determined as follows

$$
\begin{align*}
\mu_C(x_1) &= 0.4 + 0.1 - (0.4)(0.1) = 0.46 \\
\mu_C(x_2) &= 0.3 + 0.5 - (0.3)(0.5) = 0.65 \\
\mu_C(x_3) &= 0+0.6 - (0)(0.6) = 0.6 \\
\mu_C(x_4) &= 0+0.7 -(0)(0.7) = 0.7
\end{align*}
$$

Accordingly, $C = \{0.46/2, 0.65/4, 0.6/5, 0.7/8\}$

b. The bounded sum

The symbol $\oplus$ here represent the bounded sum of two fuzzy sets. The operation leads to a fuzzy set with a membership function defined by,

$$
\mu_A \oplus_B(x) = \min [1, (\mu_A(x) + \mu_B(x))]
$$

Example:

Let $A = \{0.4/2, 0.3/4, 0.6/5\}$, and $B = \{0.1/2, 0.5/4\}$

Then, $A \oplus B = \{0.5/2, 0.3/4, 1/5\}$

c. The bounded difference

The symbol $\ominus$ here represents the bounded difference of two fuzzy sets. The operation leads to a fuzzy set with a membership function defined by,

$$
\mu_A \ominus_B(x) = \min [1, (\mu_A(x) + \mu_B(x))]
$$

Example:

Let $A = \{0.5/2, 0.8/3, 0.6/7\}$, and
\[ \mathcal{B} = \{0.1/2, 0.5/3\} \]

Then, \( A \otimes \mathcal{B} = \{\mu(x_1)/2, \mu(x_2)/3, \mu(x_3)/7\} \)

where \( \mu(x_1) = \min [1, (0.5-0.1)] = 0.4, \)
\( \mu(x_2) = \min [1, (0.8-0.5)] = 0.3, \)
\( \mu(x_3) = \min [1, (0.6-0.0)] = 0.6. \)

hence, \( A \otimes \mathcal{B} = \{0.4/2, 0.3/3, 0.6/7\} \)

d. The algebraic multiplication

The algebraic product (multiplication) of two fuzzy sets \( A \) and \( \mathcal{B} \) leads to a fuzzy set \( \mathcal{C} \) such that
\[
\mathcal{C} = A \ast \mathcal{B} = \{\mu_A(a) \mu_B(b) / x \mid x \in A, x \in B\}
\]

Example:

Let \( A = \{0.4/2, 0.3/4, 0.6/5\} \), and
\( \mathcal{B} = \{0.1/2, 0.5/4, 0.7/8\} \)

Then, \( A \ast \mathcal{B} = \mathcal{C} = \{(0.4)(0.1)/2, (0.3)(0.5)/4, (0)(0.6)/5, (0)(0.7)/8\} \)
\( \mathcal{C} = \{0.04/2, 0.15/4\} \)

e. Cartesian product (Multiplication)

The Cartesian multiplication of two fuzzy sets \( A \) and \( \mathcal{B} \) leads to a fuzzy set \( \mathcal{C} \) such that
\[
\mathcal{C} = A \times \mathcal{B} = \{(\mu_{\mathcal{C}}(c)/(a, b) \mid a \in A, b \in B, \mu_{\mathcal{C}}(c) = \min[\mu_A(a), \mu_B(b)]\}
\]

Example:

Let \( A = \{0.4/3, 0.3/5, 1/6\} \), and
\( \mathcal{B} = \{0.1/2, 0.5/4\} \)

Then, \( A \times \mathcal{B} = \mathcal{C} = \{\min[0.4, 0.1]/(3, 2), \min[0.4, 0.5]/(3, 4), \min[0.3, 0.1]/(5, 2), \min[0.3, 0.5]/(5, 4), \min[0.6, 0.1]/(6, 2), \min[0.6, 0.5]/(6, 4)\} \)
\( \mathcal{C} = \{0.1/(3, 2), 0.4/(3, 4), 0.1/(5, 2), 0.3/(5, 4), 0.1/(6, 2), 0.5/(6, 4)\} \)

f. Exponent

Raising a set \( A \) to the power of \( \alpha \) is a special case of algebraic multiplication. It is defined by
\[
A^\alpha = \{((\mu_A(x))^{\alpha})/x \mid x \in A\}
\]

Example:

Let \( A = \{0.4/3, 0.3/5, 1/6\} \)

Then, \( A^2 = \{(0.4)^2/3, (0.3)^2/5, (1)^2/6\} \)
1.3.4 Properties of Fuzzy Sets

The properties of fuzzy sets are same as the properties of crisp sets [40]. Because of this fact and because the membership values of a crisp are a subset on the interval [0, 1], classical sets can be thought as a special case of fuzzy sets.

Let consider $A$, $B$, and $C$ are the three fuzzy sets. The properties of fuzzy sets operations are illustrated below:

a) Commutativity
$$A \cup B = B \cup A$$
$$A \cap B = B \cap A$$

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

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

e) Idempotency
$$A \cup A = A$$ and $$A \cap A = A$$

f) Identity
$$A \cup \phi = A$$ and $$A \cap X = A$$
$$A \cap \phi = \phi$$ and $$A \cup X = X$$

g) Transitivity
If $A \subseteq B \subseteq C$ then $A \subseteq C$

h) Involution
$$\overline{\overline{A}} = A$$

The forthcoming sections give an introduction to various controllers used in industries to regulate or synchronize the process, and control the process parameters by effective means.

1.4 PID CONTROLLER

The controllers are designed to eliminate the need for continuous operator attention. A controller is one, which compares the output value (process variable) with the desired value
(set point), determines the error and accordingly produces control action to minimize the error. Hence the controllers are used to automatically adjust some variable to hold the measurement (process variable) at the desired level [46,47]. Proportional plus Integral plus Derivative (PID) controllers are most commonly used to regulate the time-domain behavior of many different types of dynamic plants. These controllers are extremely popular because they can usually provide good closed-loop response characteristics, can be tuned using relatively simple design rules, and are easy to construct using either analog or digital components [48,49].

The ON/OFF controller is a simplest controller that turns the actuator either hard ON or fully OFF. To prevent excessive cycling or chattering, a dead band (hysteresis) is usually introduced to provide finer and smoother control. The proportional (P) controller is an improved controller over ON/OFF controller, wherein the dead band is replaced by proportional band. Over this operational band, the output of the proportional controller varies linearly with error around zero. Although capable of providing tight control than the ON/OFF controller, the proportional controller cannot fully eliminate the error to cause perfect steady state tracking between the set point and process variable. An integrator must be added to the proportional controller as it becomes proportional plus integral controller. This Proportional plus Integral (PI) controller will provide good steady state control, but responds sluggishly to transients. This deficiency can be overcome by the addition of a derivative element, which constitutes a complete Proportional plus Integral plus Derivative (PID) controller. This gives good transient as well as steady state control. It offers rapid proportional response to error, while having an automatic reset from the integral part to eliminate residual error. The derivative section stabilizes the controller and allows it to respond the rapid changes or transients in error [50].

As the PID controller composed of three components, it produces an output signal consisting of three terms-one is proportional to the error signal e(t), another one is proportional to integral of error signal e(t) and the third one is proportional to derivative of the error signal e(t) [51-53]. The equation of the PID controller is given as

\[ u(t) = K_p e(t) + K_p \int_0^t e(t) \, dt + Ki \frac{d}{dt} e(t) \]

where,

- \( K_p \) is the proportional gain
$T_i$ is the integral time
$T_d$ is the derivative time

The transfer function can be written as,

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i} s + T_d s\right)$$

The block diagram representation of the PID controller is shown in Fig. 1.6 and block diagram of the PID based control system is depicted in Fig. 1.7.

![Block diagram representation of the PID controller](image1)

![Block diagram of the PID based control system](image2)

To ensure the digital implementation of the PID control, the differential equation must be converted to a discrete difference equation as given bellow,

$$v_o = K_p (e) + K_i \int e(t)\,dt + K_d (de(t)/dt) \quad \text{...(1.1)}$$

Differentiating both sides of the above equation (1.1) with time $t'$, we get

$$dv_o/dt = K_p(de/dt) + K_i(e) + K_d (d(de/dt)/dt) \quad \text{...(1.2)}$$

Alternatively, the above equation can be written in difference form as

$$\Delta V_o/T = K_p(\Delta e/T) + K_i(e) + K_d (\Delta e/T)T \quad \text{...(1.3)}$$

where, $T = dt$ - the cycle time. Multiplying throughout by $T$ gives
\[ \Delta V_o = K_p \Delta e + K_i \Delta e T + K_d \Delta \Delta e / T \]  

where, \( \Delta V_0 = V_n - V_{n-1} \), \( \Delta e = e_n - e_{n-1} \). Rewriting equation (1.4) gives

\[ V_n - V_{n-1} = K_p (e_n - e_{n-1}) + K_i e_n T + K_d \Delta (e_n - e_{n-1}) / T \]

\[ V_n - V_{n-1} = K_p (e_n - e_{n-1}) + K_i e_n T + K_d \Delta (e_n - e_{n-1}) / T \]

\[ V_n - V_{n-1} = K_p (e_n - e_{n-1}) + K_i e_n T + K_d \left( (e_n - e_{n-1}) - (e_{n-1} - e_{n-2}) \right) / T \]

\[ V_n - V_{n-1} = K_p (e_n - e_{n-1}) + K_i e_n T + K_d \left( (e_n - 2e_{n-1} + e_{n-2}) \right) \]  

Finally, the current value of the output is given by

\[ V_n = V_{n-1} + K_p (e_n - e_{n-1}) + K_i e_n T + K_d \left( (e_n - 2e_{n-1} + e_{n-2}) \right) \]

At any instant of time the current value of the PID output \( V_n \) is calculated based on the previous value of the PID output \( V_{n-1} \), current error \( e_n \), previous error \( e_{n-1} \), previous to the previous error \( e_{n-2} \), the cycle time \( T \) and weighing constant \((K_p, K_i, K_d)\).

The graphical representation of the response of a model PID controller is shown in Fig. 1.8 from this figure it is observed that PID controller has better response over all other mentioned controllers in terms of settling time, rise time, under shoots and over shoots. Hence, the PID controllers are still widely used in many industrial systems, despite the significant developments of recent years in control theory and technology [54-58]. This is because they perform well for a wide class of processes. Also, they give robust performance for a wide range of operating conditions. The advances in control theory such as improved PID controllers; fuzzy logic controllers and fuzzy PID controllers lead to more precise control. The digital implementation of PID, fuzzy and fuzzy PID control algorithms using microprocessors, microcontrollers and personal computers lead to automation, easy analysis and further intelligent control of a process.
1.5 FUZZY LOGIC CONTROLLER

Fuzzy logic can be designed to emulate the way an experienced operator controls certain process, which is either ill-defined or too complex to model. Fuzzy logic control (FLC) is a practical alternative for a variety of challenging control applications [59-62]. Fuzzy logic controllers are special expert systems and these vary substantially according to the nature of the control problems they are supposed to solve. Control problems range from complex tasks, typical in robotics, which require a multitude of coordinated actions, to simple goals, such as maintaining a prescribed state of a single variable.

Fuzzy logic controllers, contrary to conventional controllers, are capable of utilizing knowledge elicited from human operators [63,64]. This is crucial in control problems for which it is difficult or sometimes impossible to construct precise mathematical models, or for which the acquired models are difficult or expensive to use. These difficulties may result from inherent non-linearities, the time-varying nature of the process to be controlled, larger unpredictable environmental disturbances, degrading sensors or other difficulties in obtaining precise and reliable measurements, and a host of other factors. It has been observed that experienced human operators are generally able to perform well under these circumstances. Hence fuzzy control provides a convenient method for constructing nonlinear controllers via the use of heuristic information that may come from an operator who has acted as a controller for process. A fuzzy logic controller uses a set of control rules and an inference mechanism to determine the control action for a given process state.

The block diagram of the general fuzzy logic controller is depicted in Fig.1.9. It consists of the following major modules [65,66],

i) **Fuzzification interface:** Converts controller inputs (real data) into suitable linguistic values (fuzzy variables) which the inference engine can easily use to activate and apply rules.

ii) **Fuzzy inference engine:** Emulates the expert's decision making in interpreting and applying knowledge about how to do good control. It consists of

a) **Knowledge base:** Consists of a data base with necessary linguistic definitions (rule set)

b) **Decision making logic:** Is used to decide what control action should be taken.
iii) **Defuzzification interface**: Converts the conclusions (fuzzy control action) of the inference mechanism into actual inputs (non-fuzzy or crisp control action) for the process.

The fuzzy logic controller operates by repeating a cycle of four steps. First, the measurements are taken of all variables that represent relevant conditions of the controlled process. Next, these measurements are converted into appropriate fuzzy sets to express measurement uncertainties. This step is called fuzzification. The fuzzified measurements are then used by the knowledge base to evaluate the control rules stored in the fuzzy rule base. The result of this evaluation is a fuzzy set (or several fuzzy sets) defined in the universe of possible actions. This fuzzy set is then converted, in the final step of cycle, into a single (crisp) value (or a vector of values) that, in some sense, is the best representation of the fuzzy set (or fuzzy sets). This conversion is called a defuzzification. The defuzzified values represent actions taken by the fuzzy controller in individual control cycles [39].

The detailed discussion of these four major modules in the following sub-sections provides more knowledge about the design and working of a fuzzy logic controller [67].

![Fig. 1.9 Block diagram of the general fuzzy logic controller](image)

**1.5.1 Fuzzification**

Fuzzification is the process of making a crisp quantity into fuzzy. Many of the quantities that we consider to be crisp and deterministic are actually not deterministic at all; they carry considerable uncertainty. If the form of uncertainty happens to arise because of imprecision, ambiguity, or vagueness, then the variable is probably fuzzy and can be represented by
membership functions [68]. Fuzzification determines the degree of membership; the controller input data has to all appropriate fuzzy sets. Fuzzy sets are used to quantify the information in the rule-base, and the inference mechanism operates on fuzzy sets to produce fuzzy sets. More appropriate to say, fuzzification interface converts controller inputs into information that the inference mechanism can easily use to activate and apply rules.

a. Membership functions

Membership functions characterize the fuzziness in a fuzzy set—whether the elements in the set are discrete or continuous in a graphical form for eventual use in the mathematical formalisms of fuzzy set theory. The rules used to describe fuzziness graphically are also fuzzy. Just as there are an infinite number of ways to characterize fuzziness, there are an infinite number of ways to graphically depict the membership functions that describe fuzziness.

Since all information contained in a fuzzy set is described by its membership function, it is useful to develop a lexicon of terms to describe various special features of this function. Fig. 1.10 shows the parts of a membership function. The core of a membership function for a fuzzy set \( A \) is defined as that region of the universe that is characterized by complete and full membership in the set \( A \), such that \( \mu_A(x) = 1 \). The support of a membership function for the fuzzy set \( A \) is denoted as that region of universe that is characterized by nonzero membership in the set \( A \), such that \( \mu_A(x) > 0 \). The boundaries of a membership function for the fuzzy set \( A \) are defined as that region of the universe containing elements that have a nonzero membership but not complete membership, such that \( 0 < \mu_A(x) < 1 \).

![Fig. 1.10 Representation of core, support and boundaries of a fuzzy set](image-url)
The various membership functions used for fuzzification and their mathematical representations are given below. The most common membership functions are triangular, trapezoidal and bell-shaped (Gaussian) functions depicted already in Fig. 1.3 and their mathematical expressions are given below:

For triangular membership function, $\mu(x; \alpha, \beta, \gamma) = \begin{cases} 0 & \text{if } x \leq \alpha \\ \frac{(x - \alpha)}{(\beta - \alpha)} & \text{if } x \in [\alpha, \beta] \\ \frac{(\beta - x)}{(\gamma - \beta)} & \text{if } x \in [\beta, \gamma] \\ 0 & \text{if } x \geq \gamma \end{cases}$

For trapezoidal membership function, $\mu(x; \alpha, \beta, \gamma, \delta) = \begin{cases} 1 & \text{if } x \in [\beta, \gamma] \\ \frac{(\delta - x)}{(\delta - \gamma)} & \text{if } x \in [\gamma, \delta] \\ 0 & \text{if } x \leq \alpha \text{ or } x \geq \beta \end{cases}$

For bell-shaped membership function, $\mu(x; \alpha, \beta, \gamma) = \begin{cases} 0 & \text{if } x \leq \alpha \\ e^{-(x-\beta)} & \text{if } \alpha \leq x \leq \gamma \\ 0 & \text{if } x \geq \gamma \end{cases}$

For $\Gamma$- membership function, $\mu(x; \alpha, \beta) = \begin{cases} 0 & \text{if } x \leq \alpha \\ \frac{(x - \alpha)}{(\beta - \alpha)} & \text{if } x \in [\alpha, \beta] \\ 1 & \text{if } x \geq \beta \end{cases}$

For $L$- membership function, $\mu(x; \beta, \gamma) = \begin{cases} 1 & \text{if } x \leq \beta \\ \frac{(\beta - x)}{(\gamma - \beta)} & \text{if } x \in [\beta, \gamma] \\ 0 & \text{if } x \geq \gamma \end{cases}$

For $S$- membership function, $\mu(x; \alpha, \beta, \gamma) = \begin{cases} 0 & \text{if } x \leq \alpha \\ 2^* \frac{(x - \alpha)}{(\gamma - \alpha)} & \text{if } x \in [\alpha, \beta] \\ 1 - 2^* \frac{(x - \gamma)}{(\gamma - \beta)} & \text{if } x \in [\beta, \gamma] \\ 1 & \text{if } x \geq \gamma \end{cases}$
b. Fuzzification methods

The following list provides some of the methods described in the literature to assign membership values or functions to fuzzy variables. They are,

1. Intuition          6. Genetic algorithms
2. Inference          7. Inductive reasoning
3. Rank ordering      8. Soft partitioning

Intuition method is very simple and is derived from the capacity of humans to develop membership functions through their own innate intelligence and understanding [40]. Intuition involves contextual and semantic knowledge about an issue; it can also involve linguistic truth-values about this knowledge. The membership functions for the fuzzy variable temperature are shown in Fig. 1.11. It shows various curves on the universe of temperature as measured in units of degrees Celsius. Each curve is a membership function corresponding to various fuzzy variables, such as cold, cool, warm, and hot. Of course, these curves are a function of context and the analyst developing them. For example, if the temperatures are referred to the range of human comfort, we get one set of curves; and if they are referred to the range of safe operating temperatures for a steam turbine, we get another set. However, the important character of these curves for purposes of use in fuzzy operations is the fact that they overlap. The precise shapes of these curves are not so important in their utility. Rather, it is the approximate placement of the curves on the universe of discourse, the number of curves (partitions) used, and the overlapping character that are the most important ideas. This intuition method is often employed in our present study.

Example:

![Membership functions for the fuzzy variable “temperature”](image)

Fig. 1.11 Membership functions for the fuzzy variable “temperature”
1.5.2 Fuzzy Inference Engine

The heart of the fuzzy logic controller is its inference engine where the knowledge base and decision-making logic resides. The knowledge base of FLC consists of a database and a rule base. The basic function of the database is to provide the necessary information for the proper functioning of the fuzzification module, the rule base, and the defuzzification module. The rule base, its basic function is to represent, in a structured way, the control policy of an experienced process operator and/or control engineer in the form of a set of production rules such as IF (process state) and THEN (control output). The decision making logic, it makes a decision of control to the process using the expert rule base and certain rule implications.

The fuzzy inference engine emulates the expert's decision making in interpreting and applying knowledge about how to do good control [70]. The basic function of the inference engine is to compute the overall value of the control output variable based on the individual contributions of each rule in the rule base. Each such individual contribution represents the values of the control output variables as computed by a single rule. The output of the fuzzification module, representing the current, crisp value of the process state variables, is matched to each rule antecedent, and a degree of match for each rule is established. Based on this degree of match, the value of the control output variable in the rule-antecedent is modified. The inference process of the FLC relates the fuzzy state variables e(k) and ce(k) to the fuzzy control action u(k) via a set of linguistic rules. An example of a linguistic control rule base for the FLC is shown in Fig.1.12. The linguistic variables are given as NL- Negative Large, NM- Negative Medium, NS- Negative Small, NZ- Negative Zero, ZE- Zero Error, PZ- Positive Zero, PS- Positive Small, PM- Positive Medium, and PL- Positive Large. The process state variables (fuzzy state variables) e(k) and ce(k) are defined as,

\[ e(k) = E = \frac{\text{set-point} - \text{measured value}}{\text{set-point}}, \]
\[ ce(k) = CE = \text{present error} e(k) - \text{previous error} e(k-1), \text{where 'k' sampling time} \]
\[ u(k) = U = \text{fuzzy computed control action} \]

As an example, the following is the possible control rule for a FLC

**IF** e(k) is NL and ce(k) is PM **THEN** u(k) is NS. This control rule can be regarded as an implication \( E_n, CE_n \rightarrow U_s \). There are two standard implementations of the fuzzy inference mechanism that can be used efficiently in the inference mechanism.
First type, minimum operation rule $R_c$ is called the Mamdani’s fuzzy reasoning and second type, product operation rule $R_p$ is called Larsen’s fuzzy reasoning [69].

$$R_c : \bigcup_{i=1}^{n} a_i \cap \mu_{ui}$$

$$R_p : \bigcup_{i=1}^{n} a_i \cdot \mu_{ui}$$

where, the weighting factor the $a_i$ is the measure of the contribution of $i^{th}$ rule to the fuzzy control action. The weighting factor $a_i = \mu_{EI} \cap \mu_{CEi}$, ‘$\cap$’ and ‘$\cdot$’ are the rule implications. In
literature of fuzzy control, some common inference methods such as max-min, max-prod, sum-prod methods have been reported. As an example the graphical depiction of fuzzy decision-making using ‘min’ operation (implication) and centre of gravity technique for defuzzification is presented in Fig. 1.13.

Fig. 1.13 Graphical illustration of fuzzy decision-making

1.5.3 Defuzzification

Defuzzification operation is the final component of the fuzzy logic controller. Defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the “most certain” controller output. Some think of defuzzification as “decoding” the fuzzy set information produced by the inference process into numeric fuzzy controller outputs [71,72]. Or simply defuzzification is the process of
conversion of a fuzzy quantity to a precise quantity. In recent years as many as seven defuzzification methods have been proposed by investigators [40]. They are,

1. Max-Membership principle. 2. Centre of Gravity (COG) method
3. Weighted Average method 4. Mean of Maximum (MOM) method
5. Centre of Sums method 6. Centre of Largest Area method
7. Centre of Maximum (COM) method

1. **Max membership principle:** Also known as the height method, this scheme is limited to peaked output functions. This method is given by the algebraic expression,

\[ \mu_C(z^*) \geq \mu_C(z) \quad \text{for all} \quad z \in Z \quad \text{and is shown graphically in Fig.1.14 (a)}. \]

2. **Centre of gravity (COG) method:** This procedure (also called centre of area, centroid) is the most prevalent and physically appealing of all the defuzzification methods. It is given by the algebraic expression,

\[ z^* = \frac{\int \mu_C(z) \cdot z \, dz}{\int \mu_C(z) \, dz} \]

where, \( \int \) denotes an algebraic integration. This method is shown in Fig.1.14 (b).

3. **Weighted average method:** This method is only valid for symmetrical output membership functions. It is given by the algebraic expression,

\[ z^* = \frac{\sum \mu_C(z) \cdot z}{\sum \mu_C(z)} \]

where, \( \sum \) denotes an algebraic sum. This method is shown in Fig.1.14 (c). The weighted average method is formed by weighting each membership function in the output by its respective maximum membership value.

4. **Mean of maximum (MOM) method:** This method is given by the expression,

\[ z^* = (a+b)/2 \]

where, \( a \) and \( b \) are as defined in Figure 1.14 (d).

5. **Centre of sums:** This is faster than many defuzzification methods that are presently in use. The defuzzified value \( z^* \) is given by the following equation,

\[ z^* = \frac{\int z \sum_{k=1}^{n} \mu_C(k) \cdot z \, dz}{\int \sum_{k=1}^{n} \mu_C(k) \cdot z \, dz} \]

This method is similar to the weighted average method. Fig. 1.14 (e) is an illustration of the centre of sums method.
6. Centre of largest area method: If the output fuzzy set has at least two convex sub-regions, then the centre 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. 1.14 (f) and is expressed algebraically as,

$$z^* = \frac{\int \mu \mathcal{C}_m(z) z \, dz}{\int \mu \mathcal{C}_m(z) \, dz}$$
where, $\mathcal{C}_m$ is the convex sub-region that has the largest area making up $\mathcal{C}_k$.

7. Centre of maxima (COM): This method uses the overall output or union of all individual output fuzzy sets $\mathcal{C}_k$ to determine the smallest value of the domain with maximized membership degree in $\mathcal{C}_k$. The equations for $z^*$ are as follows

First, the largest height in the union [denoted $\text{hgt}(\mathcal{C}_k)$] is determined,

$$\text{hgt}(\mathcal{C}_k) = \sup_{z \in \mathcal{Z}} \mu_{\mathcal{C}_k}(z)$$

Then the first of the maximum is found,

$$z^* = \inf_{z \in \mathcal{Z}} \{z \in \mathcal{Z} \mid \mu_{\mathcal{C}_k}(z) = \text{hgt}(\mathcal{C}_k)\}$$

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

$$z^* = \sup_{z \in \mathcal{Z}} \{z \in \mathcal{Z} \mid \mu_{\mathcal{C}_k}(z) = \text{hgt}(\mathcal{C}_k)\}$$

This method is shown graphically in Fig. 1.14 (g).

1.5.4 Digital Implementation of FLC

Since the mid-1980s, when the utility of fuzzy controller became increasingly visible, the need for computer hardware to implement the various operations involved in fuzzy logic and approximate reasoning has been recognized. The primary focus of fuzzy hardware developments has been on the implementation of fuzzy rules of inference and defuzzification procedures for fuzzy controllers. The principal reason for using specialized hardware for fuzzy computing is to increase operational speed via parallel processing. In principle, fuzzy computer hardware allows all inference rules of a complex fuzzy inference engine to be processed in parallel. This increases efficiency tremendously. As a consequence, it extends the scope of applicability of fuzzy controllers and potentiality, as compared to other expert systems. In general, computer hardware for fuzzy logic is implemented in either digital mode or analog mode. In digital mode, fuzzy sets are represented as vectors of numbers in $[0, 1]$. Operations on fuzzy sets are thus performed as operations on vectors. In analog mode, fuzzy sets are represented as continuous electric signals (electric currents or voltages) by appropriate electric circuits. These circuits play a role similar to function generators in classical analog computers. In the literature, there are many reports on the design of fuzzy hardware chips [73-76].
The digital implementation of a fuzzy logic controller is shown in Fig. 1.15. The two analog inputs to the fuzzy logic controller, error (reference - measurement) and change-in-error (present error - previous error) are digitized using A/D converter. The digitized inputs are passed to the fuzzy algorithm, which is implemented in a digital processor. The fuzzy logic controller processes the error and change-in-error and evaluates the control action. This fuzzy computed digital output is again converted into analog control action using D/A converter.

The block diagram representation of closed loop control system with the fuzzy logic controller is shown in Fig. 1.16.

The basic configuration of integrated fuzzy logic controller is shown in Fig. 1.17. It is basically a combination of fuzzy logic controller and PID controller. The IFLC is composed by cascading the fuzzy logic controller with the PID controller [77]. The fuzzy logic
controller works as an initial decision maker and the PID controller works as a final controller. The set-point to the PID controller \( c(k) \) is the sum of fuzzy computed output \( cu(k) \) and the actual set-point \( r(k) \). In the IFLC structure, the FLC has just a supplementary role to support the existed PID control system. The FLC will provide a supply of control to the PID control system; as a result the IFLC structure always acquires a satisfactory system response. Many researchers have reported that the IFLC structure possesses the good transient and steady state responses over simple PID and fuzzy control systems [78,79].

![Block diagram representation IFLC based control system](image)

**Fig. 1.17 Block diagram representation IFLC based control system**

### 1.7 ROLE OF COMPUTERS IN PROCESS INSTRUMENTATION

Instrumentation has its origin in the attempts to measure, monitor or record various physical, chemical, biological or engineering phenomena. Historically, instrumentation had its roots in mechanical and hydraulic systems, but rapidly moved to electrical and electronic devices.

The art of measurement is a wide discipline in both engineering and science, encompassing the areas of detection, acquisition, control and analysis of data. It involves the precise measurement and recording of physical, chemical, mechanical, optical and electrical parameters. Measurement plays a vital role in every branch of scientific research and industrial processes interacting basically with control systems, material sciences, and other branches of science and technology which resulted in the development of many sophisticated and high precision measuring devices and systems, catering to varied measurement problems in disciplines such as aeronautics, science and technology, space, medicine, oceanography, and industry in general.
The advent of the digital computer and its capability to perform complex mathematical calculations and making logical decisions proved a unique opportunity to improve the performance of any process through the application of supervisory control [80]. Personal computers (PCs) provide a powerful low-cost means for process control. PCs are not in direct competition with programmable logic controllers or distributed control systems, however, they are limited by both their architecture and programming. While PCs can provide direct digital control, this potential should only be utilized in special cases. PC offers the lowest hardware cost for implementing programmable control. The power of the PC and its high availability at very modest prices are its main attractions. However, the ever increasing speed and capacity and the ever shrinking cost of each new model of computer are reducing the need for sophisticated programming techniques and the extensive experience required to apply them [46].

Computer plays a vital role in both measurement and control instrumentation. In measurement of any physical parameter computer does the function of data acquisition, data processing, data storage, data presentation or display. On the other hand in control, it receives the reference command from the user, compares the acquired data with reference, performs the control algorithms and produces the control action. In both the cases the computer can further do the analysis on the acquired and stored data and in some cases even it can transmit the information over the long distance.

The speed and performance of today's computers, reaching that of any other dedicated processors/controllers, has opened up a new avenue in real time data acquisition, measurement and control. Today, as much of the attention is given for developing the sophisticated software packages to provide much easier and friendly interface to the user/operator with the computer, it is proving to be a great tool for any instrumentation/process designer, engineer, and a scientist.

The modern and expert controllers like FLC, Model Based Adaptive, Self-Tuning Control, ANN, and GA can be better implemented in computer as it provides the great deal of flexibility, power, and efficiency for developing and implementing the corresponding software and automation. PCs offer high performance and low cost, coupled with an ease of use that is unprecedented. Though, the distributed control system (DCS) has emerged as an important field for industrial control applications, another area that made a great impact, is the application of personal computers as a low cost tool [81,82].
There are many advantages of computer-based systems for measurement and control viz. CRT text and graphical read-out, observation of measured and controlled parameters, scale factors, general and scientific calculations, waveform scaling, time-frequency conversion, scale factor calculation, controlling and coordinating modules within an instrument etc. Another useful feature is auto or self-calibration. Self-calibration can also be extended to complete waveforms in waveform processing instruments. These instruments have the provision to store and print the data. Processing and analysis of data is another important feature of these instruments. There is also a provision for real-time measurements and control. Another important advantage in using computers is their self-test ability. A computer with appropriate software can easily locate the failure in the board level, or perhaps the section or chip level and also it helps in rectifying the faults. Automated instruments offer a major economic advantage because of their savings in labour costs.

In brief, the role of computer, in measurement and control point of view, is to acquire the data from sensor through analog to digital converter, do the processing on the acquired data and then generate control signals to the actuator, which in turn control the parameter being measured.
REFERENCES


