FUZZY LOGIC TECHNIQUES

4.1: BASIC CONCEPT

Problems in the real world are quite often very complex due to the element of uncertainty. Although probability theory has been an age old and effective tool to handle uncertainty, it can be applied only to situations where the system characteristics are based on random processes. In such situations, fuzzy logic exhibits immense potential for effective solving of the uncertainty in the problem.

Fuzzy logic was develop by Lotfi A. Zadeh [89, 90] and represents a form of mathematical logic. Values between 0 and 1 represent uncertainty in decision-making. 0 indicates a false value while 1 indicates a true value. So within a fuzzy set a value x is not restricted by the values 0 or 1, but from the real interval \{0;1\}

Fuzzy logic is an extension of Boolean logic which handles the concept of partial truth, where the range of truth value is in between completely true and completely false [91] In classical logic concept we can express everything in the form of 1 or 0, true or false, or, white or black. But fuzzy logic replaces Boolean truth-values with some degree of truth. This degree of truth is used to capture the imprecise modes of reasoning that play an important role in the ability of human being to make decisions in an environment of uncertainty and imprecision. The process of formulating a mapping from a given input set to an output using fuzzy logic is known as fuzzy inference. The basic elements of fuzzy logic are fuzzy sets, linguistic variables and fuzzy rules. While variables in mathematics usually take numerical values, in fuzzy logic applications, the non-numeric linguistic
variables are often used to facilitate the expression of rules and facts [92]. The linguistic variables’ are words, specifically adjectives like “small,” “little,” “medium,” “high,” and so on. A fuzzy set is a collection of couples of elements.

In narrow sense, fuzzy logic can be defined as a logical system that is the extension of multivalued logic. In a wider sense fuzzy logic is almost similar to the theory of fuzzy sets, which talks about the relationship between classes of object with unsharp boundaries and their membership values.

4.2: FUZZY LOGIC CONTROLLER

To tackle the load balancing problem, conventional control theory can be applied to restore system equilibrium. Fuzzy logic control attempts to capture intuition in the form of IF-THEN rules, and conclusions are drawn from these rules [93]. Based on both intuitive and expert knowledge, system parameters can be modeled as linguistic variables and their corresponding membership functions can be designed. Thus, nonlinear system with great complexity and uncertainty can be effectively controlled based on fuzzy rules without dealing with complex, uncertain, and error-prone mathematical models.
The architecture of the fuzzy logic controller shown in figure 4.1 above includes four components: Fuzzifier, Rule Base, Fuzzy Inference Engine, and Defuzzifier.

**Fuzzifier:** The fuzzifier is the input interface which maps a numeric input to a fuzzy set so that it can be matched with the premises of the fuzzy rules defined in the application-specific rule base.

**Rule Base:** The rule base contains a set of fuzzy if-then rules which defines the actions of the controller in terms of linguistic variables and membership functions of linguistic terms.

**Fuzzy Inference Engine:** The fuzzy inference engine applies the inference mechanism to the set of rules in the fuzzy rule base to produce a fuzzy output set. This involves matching the input fuzzy set with the premises of the rules, activation of the rules to deduce the conclusion of each rule that is fired, and combination of all activated conclusions using fuzzy set union to generate fuzzy set output.

**Defuzzifier:** The defuzzifier is an output mapping which converts fuzzy set output to a crisp output. Based on the crisp output, the fuzzy logic controller can drive the system under control.

The fuzzy rule base contains a set of linguistic rules. These linguistic rules are expressed using linguistic values and linguistic variables. Different linguistic values can be assigned to a linguistic variable. These linguistic values are modeled as fuzzy sets. Based on the linguistic values, their corresponding membership functions can be expressed based on
application requirements. So, we can say that the job of a fuzzy logic controller is to carry out the following three steps.

1. To receive one or a large number, of measurement or other assessment of conditions existing in some system we wish to analyze or control.

2. Processing all these inputs according to human based, fuzzy "If-Then" rules, which can be expressed in plain language words.

3. Averaging and weighting the resulting outputs from all the individual rules into one single output decision or signal which decides what to do or tells a controlled system what to do. The output signal eventually arrived at is a precise appearing, defuzzified, "crisp" value.

4.3: FUZZY SET THEORY

A fuzzy set is a set which has no crisp, clearly defined boundary. It’s elements only have partial degree of membership.

4.3.1: Membership Functions A curve that tells how all of the input points are mapped to the membership value is called a membership function (MF). This membership value is also known as degree of membership and its range is between 0 and 1. The input space here is known as the universe of discourse and the output-axis has the membership value which is in between 0 and 1. The curve known as membership function is represented by the symbol $\mu$. 
A fuzzy set is nothing but an extension of a classical set. If the Universe of discourse is $X$ and its elements are represented by $x$, then a set of ordered pair

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

defines a fuzzy set where,

$\mu_A(x)$ is the membership function (or MF) of $x$ in $A$. This function maps each element of $X$ to a membership value between 0 and 1.

The only condition a membership function must satisfy is that it must vary between 0 and 1. There are 11 built-in membership functions in the fuzzy logic toolbox. Some of them are: triangular membership function, trapezoidal membership function etc. One example of a membership function for an input variable having three linguistic variables low, medium and high is shown in figure 4.2.

![Figure 4.2: Membership function for an input variable with three linguistic variables low, medium and high.](image)

4.3.2: Logical Operations

The fuzzy logical reasoning is a superset of standard Boolean logic. If in fuzzy logic we keep the membership values at the two extremes of 0 (completely false) and 1
(completely true) it becomes the same as Boolean logic. The statement $A \text{ AND } B$ is resolved using the min function, where $A$ and $B$ are limited to the range (0, 1). Similarly the OR operation is replaced by the $max$ function, such that $A \text{ OR } B$ is equivalent to $max (A, B)$. Similarly $\neg A$ is same as the operation $1 - A$.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Min(A,B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 4.3: Truth table for AND operator**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Max(A,B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 4.4: Truth table for OR operator**
### 4.3.3: If-Then Rules

We have the concept of a subject and a verb in fuzzy logic and if-then rule statements are used to make conditional statements that are the building blocks of fuzzy logic. The form of a single fuzzy if-then rule is

\[ \text{if } x \text{ is } A \text{ then } y \text{ is } B \]

where \(A\) and \(B\) are the values of the linguistic variables \(X\) and \(Y\), respectively. \textit{Antecedent} or premise is the if-part of the rule “\(x\) is A” while \textit{consequent} or conclusion is the then-part of the rule “\(y\) is \(B\)”.  

**Interpretation of If-Then Rules**

The interpretation of the if-then rule is done in three parts:

1. **Fuzzify inputs**: All the fuzzy statements in the antecedent are resolved to a degree of membership between 0 and 1. If the antecedent has only one part then this is the degree of support for the rule.
2 **Apply fuzzy operator to multiple part antecedents**: If the antecedent has more than one part then we apply fuzzy logic operators to resolve the antecedent to a single number between 0 and 1. This becomes the degree of support for the rule.

3 **Apply implication method**: Now the degree of support for the entire rule is used to shape the output fuzzy set. An entire fuzzy set is assigned to the output by the consequent. This fuzzy set is represented by a membership function indicates the qualities of the consequent. The fuzzy set is truncated according to the implication method if the antecedent is only partially true, (i.e., is assigned a value less than 1).

4.4: **FUZZY INFERENCE SYSTEM**

The process of creating a mapping between input and output using fuzzy logic is known as fuzzy inference. The mapping is the base from which decisions can be made, or patterns discerned. Two types of fuzzy inference systems can be implemented in the toolbox: Mamdani-type and Sugeno-type. The description of these two methods is given in [94, 95]. The most commonly used method is the Mamdani’s fuzzy inference system. This was one of the first control systems built using fuzzy set theory proposed by Ebrahim Mamdani [96] in 1975. It was developed in an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Lotfi Zadeh’s 1973 paper on fuzzy algorithms for complex systems and decision processes [97] was the driving force behind this work of Mamdani.

After the aggregation process, we get a fuzzy set for each output variable which is defuzzified to get the crisp values.
4.5: FUZZY LOGIC APPLICATION IN LOAD DISTRIBUTION

There are numerous papers which talk about use of fuzzy logic in load balancing. These papers address the issue of uncertainty in the state of the distributed system and how it can be resolved using fuzzy logic. Stankovic [98] applied the Bayesian decision theory to job scheduling to manage uncertainty in the overall system state. A fuzzy expert system for load balancing has been described by Kumar et al [99]. Each node of a distributed computing system has an expert system that plays the role of a distributed decision maker. The fuzzy expert system reflects the impression in state information and makes scheduling decision based on a fuzzy logic. In [100] it has been assumed that the performance of any distributed computing system cannot be improved beyond a limit which is determined by the degree of uncertainty in global state. Here each distributed node dynamically adjusts its thresholds denoting the amount of consistency relaxation depending on the degree of uncertainty in the system state. A fuzzy based consistency model provides a mechanism that allows each node to make flexible scheduling and state update decisions based on its threshold.
Dierkes [101] has compared his own fuzzy based load balancing algorithm with two standard algorithms: Join Shortest Queue and prefer fastest. For his own algorithm also he has taken different scenarios with different number of rules and fuzzy sets. By using Mamdam’s fuzzy inference [102] in his decision algorithm, Dierkes was able to map attributes such as queue length onto a fuzzy set with three terms (small medium and large) over specific ranges. In another paper [103], Dierkes based load balancing decision making on a set of possible actions, a set of goals with priorities, and an effect matrix. The effect matrix showed the effect of the actions on the goals (whether the action either distracts or supports a goal). The application of fuzzy sets taken from the matrix significantly improved system performance, with a bias towards achieving the goals given weight in the effect matrix.

Three practical approaches for load balancing in distributed object computing systems have been studied in [104]. This paper considers JavaSpecs based, request redirection based and fuzzy decision based approaches and it has been found that fuzzy decision based algorithm outperforms the other two considerably. Another fuzzy logic based algorithm has been implemented in [105] which effectively reduces the amount of communication messages and also helps in giving high throughputs, low response time and short turnaround times. Fuzzy logic has been used for scheduling of periodic tasks in soft real time multiprocessor systems in [106]. Load balancing in Web server cluster has been proposed in [107] where the actual status of the real server has been modeled using fuzzy inference system and adaptive Neuro-fuzzy inference system. It has been shown that by using these approaches the scheduling overhead has been reduced and availability of these servers has been increased.
Center of gravity defuzzification method has been performed using the discretisation method, and two new methods namely slope based method and the modified transformation function method in [108]. It has been found that the new methods give very good accuracy at low computational cost as compared to the discretisation method.