Design and Implementation of Fuzzy Controller for Traffic Signals to Optimize Traffic Congestion Delays

A

Dissertation

Submitted in partial fulfillment of the requirement
For the award of degree of

MASTER OF ENGINEERING
(Computer Technology and Applications)

Submitted by

PARVEEN JAIN

(University Roll No. 8561)
(College Roll No. 10/CTA/07)

Under the Expert Guidance of
Mr. MANOJ SETHI

DEPARTMENT OF COMPUTER ENGINEERING
DELHI COLLEGE OF ENGINEERING
BAWANA ROAD, DELHI
(DELHI UNIVERSITY)

CERTIFICATE

DELHI COLLEGE OF ENGINEERING
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI - 110042

Date:___________

This is to certify that Dissertation entitled **"Design and implementation of Fuzzy Controller for Traffic Signals to Optimize Traffic Congestion Delays"** has been completed by **Mr. Parveen Jain under roll no. 8561** in partial fulfillment of the requirement for the award of degree of **Master of Engineering** in **Computer Technology & Applications**. This work has not been submitted elsewhere for a degree.

This is a record of work carried out by him under my supervision and support. This is a beneficial work in field of Fuzzy Control System for developing traffic light control system to improve the traffic condition on roads by using the currently available technologies and resources.

(Mr. MANOJ SETHI)
PROJECT GUIDE
(Dept. of Computer Engineering)
DELHI COLLEGE OF ENGINEERING
(Now Delhi Technological University)
BAWANA ROAD, DELHI - 110042

ACKNOWLEDGEMENT

This work is not a solo endeavor but rather the amalgamate consequence of contribution from various people and sources. Therefore it would be discourteous to present it without acknowledging their valuable guidance.

It is distinct pleasure to express my deep gratitude and indebtedness to my learned supervisor Mr. Manoj Sethi for his valuable guidance, encouragement and patient reviews. His continuous inspiration only has made me complete this dissertation. I am also thankful to Mr. Ravin Ahuja for his timely guidance and giving me time from his busy schedule. All of them kept me boosting time to time and for putting an extra ounce of effort to realize this work.

I would also like to take this opportunity to present my sincere regards to my teachers Dr. Daya Gupta, Dr. Rajni Jindal and Dr. S. K. Saxena for their support, encouragement and unstinted co-operation. I am also thankful to my classmates for their unconditional support and motivation during work.

Finally, I thank my parents and almighty God for making me capable to purse degree in such a reputed college and to pursue work in dedicated wide scoped research field.

(PARVEEN JAIN)
Roll no. 10/CTA/07
Univ. Roll no: 8561
Master of Engineering
(Computer Technology & Application)
Dept. of Computer Engineering
DELHI COLLEGE OF ENGINEERING
BAWANA ROAD, DELHI - 110042

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Transportation research has the goal to optimize transportation flow of people and goods. As the number of road users constantly increases, and resources provided by current infrastructures are limited, intelligent control of traffic is now a very important issue. Therefore the need arises for simulating and optimizing traffic control algorithms to better accommodate this increasing demand. However, some limitations to the usage of intelligent traffic control exist. Avoiding traffic jams for example is thought to be beneficial to both environment and economy like a decrease in each of delay, number of stops, fuel consumption, pollutant emissions, noise, vehicle operating costs, queue length and personal time, but improved traffic-flow may also lead to an increase in demand Optimization of traffic light switching increases road capacity and traffic flow, and can prevent traffic congestions.

As basically, the time period of traffic light depends on the load and number of vehicle on the road. Here main focus is on the two subsystems namely the sensor mechanism and the controller unit. The fuzzy controller will regularly query the traffic conditions in order to decide whether to extend or terminate a current phase. Here I am presenting a systematic Fuzzy based traffic signal controller for controlling the traffic light.

The project work shows that traffic congestion delay can be considerably reduced by replacing the conventional traffic control system by fuzzy based traffic control system. Fuzzy traffic signal Controller can reduce the waiting time or congestion delays by significant amount of time.

1. INTRODUCTION

1.1 Beginning of Fuzzy

The real world is complex; complexity in the world generally arises from uncertainty in form of ambiguity. Problems featuring complexity and ambiguity have been addressed subconscious by humans since they could think; these ubiquitous features pervade most social, technical, and economic problems faced by human race. Why then are computers, which have been designed by human after all, not capable of addressing complex and ambiguous issues? How can humans reason about real, when the complete description of a real system often requires more detailed date than a human could ever hope to recognize simultaneously and assimilate with understanding? The answer is that human have the capacity to reason approximately, a capability that computers currently do not have. In reasoning about a complex system, Human reason approximately about its behavior, thereby maintaining only a generic understanding about the problem. Fortunately, this generality and ambiguity are sufficient for human comprehension of complex systems.

As we learn more and more about system, its complexity decreases and our understanding increases. As complexity decreases, the precision afforded by computational methods become more useful in modeling the system. Close form mathematical expressions provide precise descriptions of the systems. For system that are a little more complex, but for significant data exist, Model Free Methods such as artificial neural networks, provide a powerful and robust means to reduce some uncertainty through learning based on patterns in available data. Finally for the most complex system where few numerical data exist and where only ambiguous or imprecise information may be available, fuzzy reasoning provides a way to understand system behavior by allowing us to interpolate approximately between observed input and output situations. The imprecision in fuzzy models is therefore generally quite high. Fuzzy system can implement crisp input and output and in case produce a non linear functional mapping just as do algorithms.

All are mathematical abstractions of the real physical real world. In situations where precision is apparent, for example, fuzzy systems are less efficient than the more precise algorithm in providing us with the best understanding of the problem. On other hand, Fuzzy system can focus on modeling problems characterized by imprecise or ambiguous information.

1.2 Background Details

The real world is very complex; complexity in the world generally arises from uncertainty in form of ambiguity. In decade after Dr. Zadeh's seminal paper on fuzzy sets [Zadeh, 1965], Many theoretical developments in fuzzy logic took place in the United States, Europe, and Japan. However Japanese researchers have been a primary force in advancing the practical implementation of the theory. Much of the success of the new products associated with the fuzzy technology is due to fuzzy logic. And some is also due to advanced sensors used in these products.

Fuzzy logic affects many disciplines. In videography, for instance, Fisher, Sanyo, and other make fuzzy logic camcorders, which offer fuzzy focusing and image stabilization. Mitsubishi manufacturers a fuzzy air conditioner that controls temperature change according to human comfort indexes. Matsushita builds a fuzzy washing machine that combines smart sensors with fuzzy logic. The sensors detect the color and kind of clothes present and the quantity of grit, and a fuzzy microprocessor selects the most appropriate combination from 600 available combinations of water temperature, detergent amount and washes and spins cycle times. The Japanese city of Sendai has a 16 station subway system that is controlled by a fuzzy computer. The ride is so smooth; riders do not need to hold straps and the controller makes 70 percent fewer judgment errors in acceleration and breaking than human operators. Nissan introduced a fuzzy automatic transmission and fuzzy anti skid breaking system in one of their recent luxury cars. In Japan, there are fuzzy golf diagnostic system, fuzzy toasters, fuzzy rice cookers, fuzzy vacuum cleaners and many other industrial fuzzy control processes[1].

In fact, the number of fuzzy consumer products and fuzzy applications involving new patents is increasing so rapidly that in order to stay competitive; many companies are launching their own internal fuzzy projects. National Aeronautics and space administration has been involved for a number of years in use of fuzzy logic in space docking control. The term Fuzzy carries negative connotations in the English-speaking world that it does not in other, especially Asian cultures. Few Linguists accept a theory of strong linguistic determinism, which argues that fluency in different languages gibes the speaker special strong cognitive advantages and disadvantages.

1.3 Uncertainty and Imprecision

Fuzzy set theory provides a means for representing uncertainties. Historically, Probability theory has been the primary tool for representing the uncertainty in mathematic models. Because of this, all uncertainty was assumed to follow the characteristics of random uncertainty. A random process is one where the outcomes of any particular realization of the process are strictly a matter of chance; a prediction of sequence of events is not possible. What is possible for random sequence is a precise description of the statistics of the long run averages of the process. However, not all uncertainty is random. Some forms of uncertainty are non-random and hence not suited to treatment or modeling by probability theory. In fact, it could be argued that the overwhelming amount of uncertainty associated with complex system and issues, which humans address on a daily basis, is non random in nature. Fuzzy set theory is marvelous tool for modeling the kind of uncertainty associated with vagueness, with imprecision, and with a lack of information regarding a particular element of the problem at hand.

Many of us, especially those in positions to develop models of physical processes, understand that we lack complete information in solving problems. Some of the information we do have about a particular problem might be judgmental, perhaps an intuitive reaction on the part of modeler, rather than hard quantitative information. How then do we incorporate intuition into a problem.

One prevalent way to convey information is our own mean of communication: natural language. By its very nature, natural language is a vague or imprecise, yet is a most powerful form of communication and information exchange among humans. Despite the vagueness in natural language, Human has little trouble understanding one other's concept and ideas: this understanding is not possible in communications in computer which requires extreme precision in its instructions. For instance, what is the meaning of a "tall person ?". To individual a tall person may be anybody over 5'10". To an individual B a tall person is someone who is 6'2" or taller. What sort of meaning does the linguistic descriptor "tall" convey to either of these individuals [1].

It is very surprising that, despite the potential for misunderstanding, the term "Tall" conveys sufficiently similar information to the two individuals, even they are significantly different heights themselves and that understanding and correct communication are possible between them. Individuals A and B regardless of their heights; do not required identical information of term "Tall" to communicate effectively; again a computer would require a specific height to compare with a pre assigned value for "Tall".

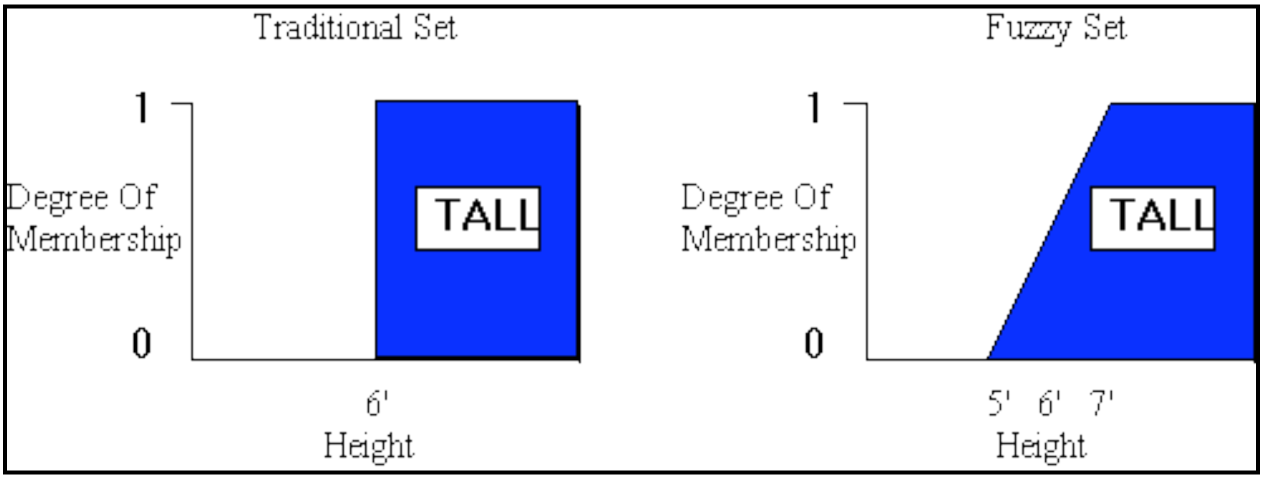


Figure 1.1. Uncertainty and Imprecision in Fuzzy Logic [1]

The power of fuzzy set is that it uses linguistic variables rather than quantitative variables., to represent imprecise concepts. The incorporation of fuzzy set theory and fuzzy logic into computer models has shown tremendous pay off in area where intuition and judgment still play major roles in model. Control applications such as temperature control, traffic control or process control are the most prevalent of current fuzzy logic applications. Fuzzy logic seems to be most successful in two kind of situation:

(1) Very complex models where understanding is strictly limited or in fact quite judgmental.
(2) Processes where human reasoning, human perception or human decisions making are inextricably involved.

Generally, Simple linear system or naturally automated processes have not been improved by the implementation of fuzzy logic. Fuzzy logic is not a panacea for all the problems. However, its value has been demonstrated numerous times in the two kinds of situations just mentioned and these comprise many of the processes or products of interest to a large number of consumers. Our understanding of physical resources is based largely on imprecise human reasoning. This imprecision's is nonetheless form of information that can be quite useful to humans. The ability to embed such reasoning in hitherto intractable and complex problems is the criteria by which the efficacy of fuzzy logic is judged. Undoubtedly this ability cannot solve problems that require Precision problems such as shooting precision laser beams over tens of kilometers in space; milling machine components to accuracies of parts per billion; or focusing a microscopic electron beam as a whole rather than as a collection of independent common a specimen the size of nanometer. The impact of fuzzy logic in these areas might be years away, if ever. But not many human problems require such precision problems, such as parking a car, navigating a car among others on a free way, washing clothes, controlling traffic at intersections, judging beauty contestants and so on.

Requiring precision in engineering models and product translates to requiring high cost and long lead times in production and development. For other than simple system, expense is proportional to precision. More precision entails higher cost. When considering the use of fuzzy logic for a given problem, an engineer or scientist should ponder the need of exploiting the tolerance for imprecision. Not only does high precision dictate high costs but it also entails low tractability problem.

1.4 Information Processing in Real World

Fuzzy logic has long excelled at delivering exact results from imprecise or ambiguous information, and its primary use has been in embedded controllers. Now fuzzy logic is entering the mainstream with a wide range of desktop applications. In the next few years, fuzzy logic will enter domains such as computer chips computer graphics, software development, financial planning, information processing, sales analysis, speech recognition, machine vision, and character recognition. There is uncertainty that arises from ignorance from chance from various classes of randomness from, imprecision, form the inability to perform adequate from lack of Knowledge or from vagueness like fuzziness inherent in our natural language. Cox cites an example of fuzzy-logic application that was developed for a bank. The application runs on PCs with Windows 3.1 and Excel, and it analyzes and rates the complexity of a software development project. This program takes into consideration function point, code density, and the total operational interface. The application calculates complexity indicators in software, such as the number of IF. . . THEN. . . ELSE statements, nested IF. . . THEN. . . ELSE statements, GO TOs, and comments. You use these rough figures to calculate ratios and statistical measures and feed them into a fuzzy-evaluation model. Previous attempts to measure complexity relied on sharp boundaries between what is and what is not complex. The fuzzy approach more closely models the way that managers think in degrees such as somewhat, moderately, and highly complex.

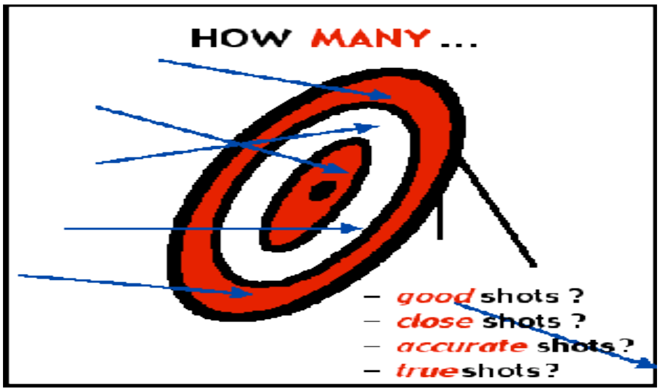


Figure 1.2. Fuzzy Logic in Real world [http://www.aai.org/aitopics/fuzzy.gif]

The employment of Fuzzy Control is no good idea if...

(1) Conventional control theory yields a satisfying result.

(2) An easily solvable and adequate mathematical model already exists.

(3) The problem is not solvable.

2.1. Requirements of Fuzzy Logic

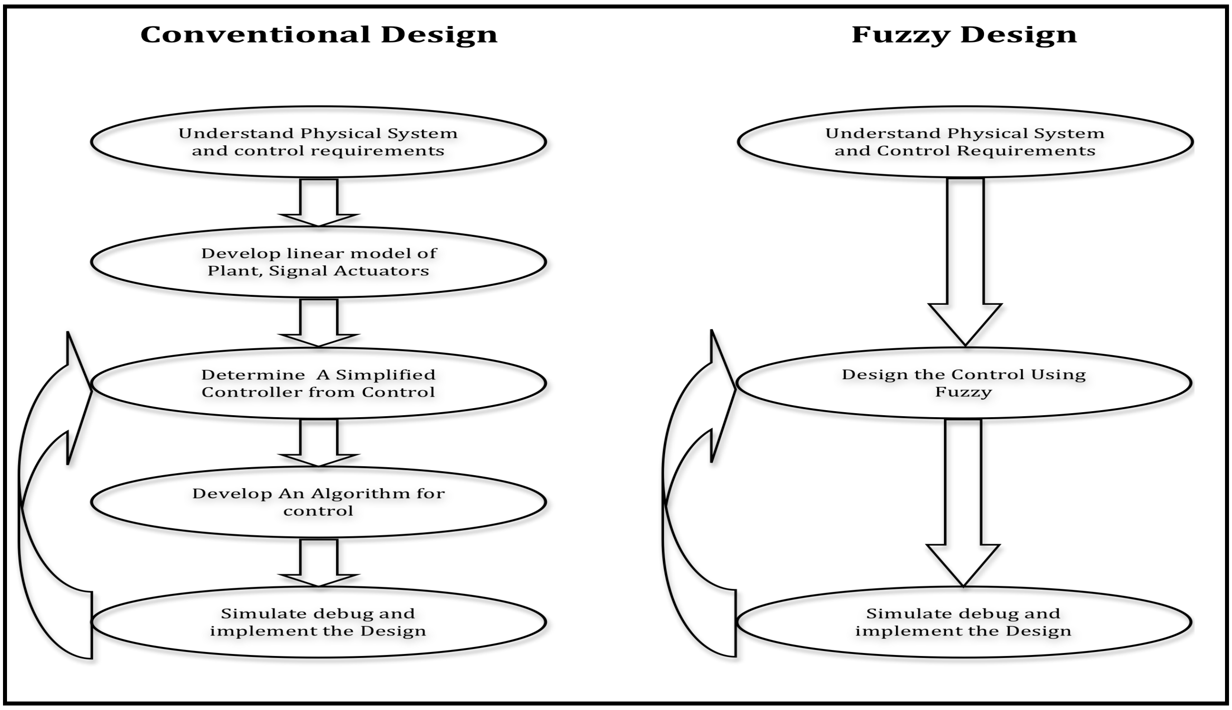
In order to appreciate why a fuzzy based design methodology is very attractive in embedded control applications let us examine a typical design flow. Figure 2.1 illustrates a sequence of design steps required to develop a controller using a conventional and a Fuzzy approach. Using the conventional approach, first step is to understand the physical system and its control requirements. Based on this understanding, second step is to develop a model which includes the plant, sensors and actuators. The third step is to use linear control theory in order to determine a simplified version of the controller, such as the parameters of a PID controller. The fourth step is to develop an algorithm for the simplified controller. The last step is to simulate the design including the effects of non-linearity, noise, and parameter variations. If the performance is not satisfactory we need to modify our system modeling, re-design the controller, re-write the algorithm and re-try. With Fuzzy Logic the first step is to understand and characterize the system behavior by using our knowledge and experience. The second step is to directly design the control algorithm using fuzzy rules, which describe the principles of the controller's regulation in terms of the relationship between its inputs and outputs. The last step is to simulate and debug the design. If the performance is not satisfactory we only need to modify some fuzzy rules and re-try.

Although the two design methodologies are similar, the fuzzy-based methodology substantially simplifies the design loop. This results in some significant benefits, such as reduced development time, simpler design and faster time to market [2]:

(1) Fuzzy Logic reduces the design development cycle: With a fuzzy logic design methodology some time consuming steps are eliminated. Moreover, during the debugging and tuning cycle you can change your system by simply modifying rules, instead of redesigning the controller. In addition, since fuzzy is rule based, you do not need to be an expert in a high or low-level language. This helps you focus more on your application instead of programming. As a result, Fuzzy Logic substantially reduces the overall development cycle.

(2) Fuzzy Logic improves time to market: Commercial applications in embedded control Require a significant development effort a majority of which is spent on the software portion of the project. Development time is a function of design complexity, and the number of iterations

required in a debugging and tuning cycle. As we explained above, a fuzzy based design methodology addresses both issues very effectively. Moreover, due to its simplicity the description of a fuzzy controller not only is transportable across design teams, but also provides a superior media to preserve, maintain, and upgrade intellectual property. As a result, Fuzzy Logic can dramatically improve time to market.



2.1. Conventional and Fuzzy Design

(3) Fuzzy Logic simplifies design complexity: Fuzzy logic lets you describe complex systems using your knowledge and experience in simple English-like rules. It does not require any system modeling or complex math equations governing the relationship between inputs and outputs. Fuzzy rules are very easy to learn and use, even by non-experts. It typically takes only a Few rules to describe systems that may require several of lines of conventional software**.**

(4) Fuzzy Logic reduces hardware costs: Fuzzy Logic enables you to use a simple rule based approach which offers significant cost savings, both in memory and processor class.

(5) Fuzzy Logic simplifies implementation: Most control applications have multiple inputs and require modeling and tuning of a large number of parameters which makes implementation very

tedious and time consuming. Fuzzy rules can help you simplify implementation by combining multiple inputs into single if-then statements while still handling non-linearity. Also it requires minimum mathematical calculations.

2.2 Fuzzy Logic

Fuzzy Logic was initiated in 1965 by Lotfi A. Zadeh , professor for computer science at the University of California in Berkeley. Conventional computer logic was incapable of manipulating data representing subjective or vague human ideas such as "an attractive person" or "pretty hot". Fuzzy logic hence was designed to allow computers to determine the distinctions among data with shades of gray, similar to the process of human reasoning. Zadeh published his seminal work "Fuzzy Sets" which described the mathematics of fuzzy set theory, and by extension fuzzy logic. This theory proposed making the membership function (or the values False and True) operate over the range of real numbers [0.0, 1.0]. Fuzzy logic was now introduced to the world.

2.2.1. Theory of Fuzzy Logic[3]

A logic based on the two truth values Truth and False is sometimes inadequate when describing human reasoning. Fuzzy logic uses the whole interval between 0 and 1 to describe human reasoning. Basically, Fuzzy Logic (FL) is a multi-valued logic that allows intermediate values to be defined between conventional evaluations like true/false, yes/no, high/low, etc. Notions like rather tall or very fast can be formulated mathematically and processed by computers, in order to apply a more human-like way of thinking in the programming of computers. Fuzzy system is an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy. Both degree of yes or no range between 0 and 1 or we can say. Both degrees of truth and probabilities range between 0 and 1 and hence may seem similar at first. However, they are distinct conceptually; truth represents membership in vaguely defined sets, not likelihood of some event or condition as in probability theory. For example, let a 100-ml glass contain 30 ml of water. Then we may consider two concepts: Empty and Full. The meaning of each of them can be represented by a certain fuzzy set. Then one might define the glass as being 0.7 empty and 0.3 full. Note that the concept of emptiness would be subjective and thus would depend on the observer or designer. Another designer might equally well design a set membership function where the glass would be considered full for all values down to 50 ml. It is essential to realize that fuzzy logic uses truth degrees as a mathematical model of the vagueness phenomenon while probability is a mathematical model of randomness.

A probabilistic setting would first define a scalar variable for the fullness of the glass, and second, conditional distributions describing the probability that someone would call the glass full given a specific fullness level. This model, however, has no sense without accepting occurrence of some event e.g. that after a few minutes, the glass will be half empty. Note that the conditioning can be achieved by having a specific observer that randomly selects the label for the glass, a distribution over deterministic observers, or both. Consequently, probability has nothing in common with fuzziness; these are simply different concepts, which superficially seem similar because of using the same interval of real numbers [0, 1]. Still, since theorems such as De Morgan's have dual applicability and properties of random variables are analogous to properties of binary logic states, one can see where the confusion might arise.

A fuzzy controller, in a cement plant for example, aims to mimic the operator's terms by means of fuzzy logic. To illustrate, consider the tank in Fig. 1, which is for feeding a cement mill such that the feed f low is more or less constant. The simplified design in the figure 2.2 consists of a tank, two level sensors, and a magnetic valve. The objective is to control the valve V1, such that the tank is refilled when the level is as low as LL, and stop the refilling when the level is as high as LH. The sensor LL is 1 when the level is above the mark, and 0 when the level is below; likewise with the sensor LH. The valve opens when V1 is set to 1, and it closes when V1 is set to 0. In two-valued (Boolean) logic the controller can be described as:

V1= 1, If LLswitches from 1 to 0

V1= 0, If LH switches from 0 to 1.

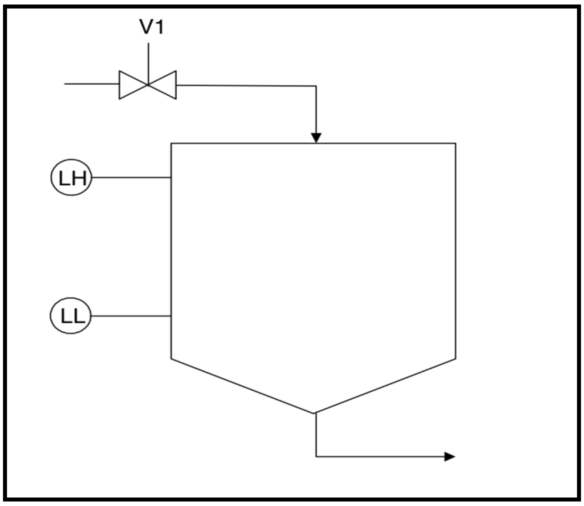


Figure 2.2. Process Diagram of Feed Tank [3]

2.2.2. Fuzzy Sets

Fuzzy sets are a further development of the mathematical concept of a set. Sets were first studied formally by the German mathematician Georg Cantor (1845-1918). His theory of Sets met much resistance during his lifetime, but now days most mathematicians believe it is possible to express most, if not all, of mathematics in the language of set theory. Many researchers are looking at the consequences of 'fuzzifying' set theory, and much mathematical literature is the result. For control engineers, fuzzy logic and fuzzy relations are the most important in order to understand how fuzzy rules work.

The very basic notion of fuzzy systems is a fuzzy (sub)set. In classical mathematics we are familiar with what we call crisp sets. For example, the possible interferometric coherence g values are the set X of all real numbers between 0 and 1. From this set X a subset A can be defined, (e.g. all values $0 \leq g \leq 0.2$). The characteristic function of A, (i.e. this function assigns a number 1 or 0 to each element in X , depending on whether the element is in the subset A or not) is shown in Fig.2.3.

The elements which have been assigned the number 1 can be interpreted as the elements that are in the set A and the elements which have assigned the number 0 as the elements that are not in the set . A. This concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis.

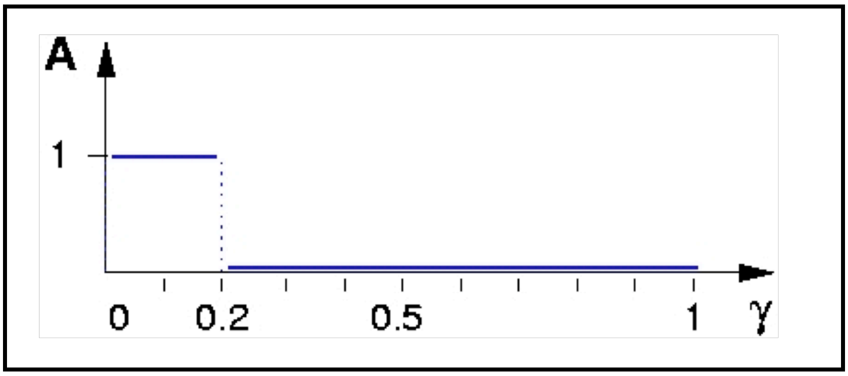


Figure 2.3. Characteristic Function of Crisp Set [3]

Since g starts at 0, the lower range of this set ought to be clear. The upper range, on the other hand, is rather hard to define. As a first attempt, we set the upper range to 0.2. Therefore we get B as a crisp interval B=[0,0.2]. But this means that a g value of 0.20 is low but a g value of 0.21 not. Obviously, this is a structural problem, for if we moved the upper boundary of the range from g =0.20 to an arbitrary point we can pose the same question. A more natural way to construct the set B would be to relax the strict separation between low and not low.

This can be done by allowing not only the (crisp) decision Yes/No, but more flexible rules like " fairly low". A fuzzy set allows us to define such a notion. The aim is to use fuzzy sets in order to make computers more 'intelligent', therefore, the idea above has to be coded more formally.

A straight way to generalize this concept, is to allow more values between 0 and 1. In fact, infinitely many alternatives can be allowed between the boundaries 0 and 1, namely the unit interval I = [0, 1]. The interpretation of the numbers, now assigned to all elements is much more difficult. Of course, again the number 1 assigned to an element means, that the element is in the set B and 0 means that the element is definitely not in the set B. All other values mean a gradual membership to the set B. This is shown in Fig. 2.4.

The membership function, operating in this case on the fuzzy set of interferometric coherence g, returns a value between 0.0 and 1.0. For example, an interferometric coherence g of 0.3 has a Membership of 0.5 to the set low coherence (see Fig. 2.4). It is important to point out the distinction between fuzzy logic and probability. Both operate over the same numeric range, and have similar values: 0.0 representing False (or non-membership), and 1.0 representing True (or full-membership). However, there is a distinction to be made between the two statements: The probabilistic approach yields the natural-language statement,

"There is a 50% chance that g is low," while the fuzzy terminology corresponds to "g's degree of membership within the set of low interferometric coherence is 0.50." The semantic difference is significant: the first view supposes that g is or is not low; it is just that we only have a 50% chance of knowing which set it is in. By contrast, fuzzy terminology supposes that g is "more or less" low, or in some other term corresponding to the value of 0.50.

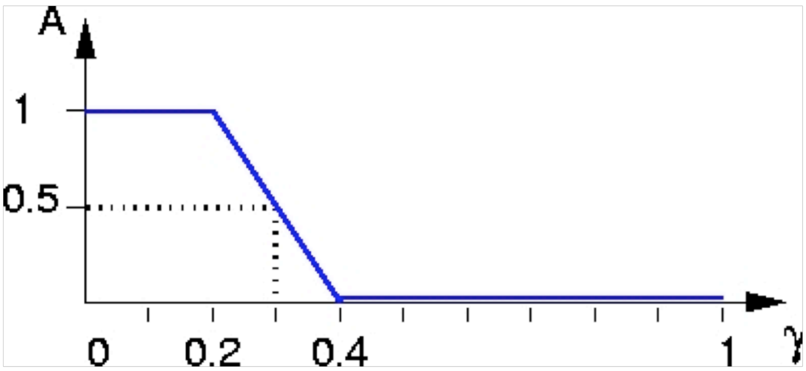


Figure 2.4. Characteristic function of Fuzzy Set [3]

2.2.3. Membership Functions

The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. There are different memberships functions associated with each input and output response. Some features to note are:

SHAPE - triangular is common, but bell, trapezoidal, sine and, exponential have been used. Complex functions are possible but require greater computing overhead to implement.. HEIGHT magnitude (usually normalized to 1) WIDTH (of the base of function), SHOULDERING (locks height at maximum if an outer function. Shouldered functions evaluate as 1.0 past their center) CENTER points (center of the member function shape) OVERLAP (N&Z, Z&P, typically about 50% of width but can be less). Figure 2.5 illustrates the features of the triangular membership function which is used in this example because of its mathematical simplicity. Other shapes can be used but the triangular shape lends itself to this illustration.
The degree of membership (DOM) is determined by plugging the selected input parameter (error or error-dot) into the horizontal axis and projecting vertically to the upper boundary of the membership function(s). In Figure2.6, consider an "error" of -1.0 and an "error-dot" of +2.5. [4]

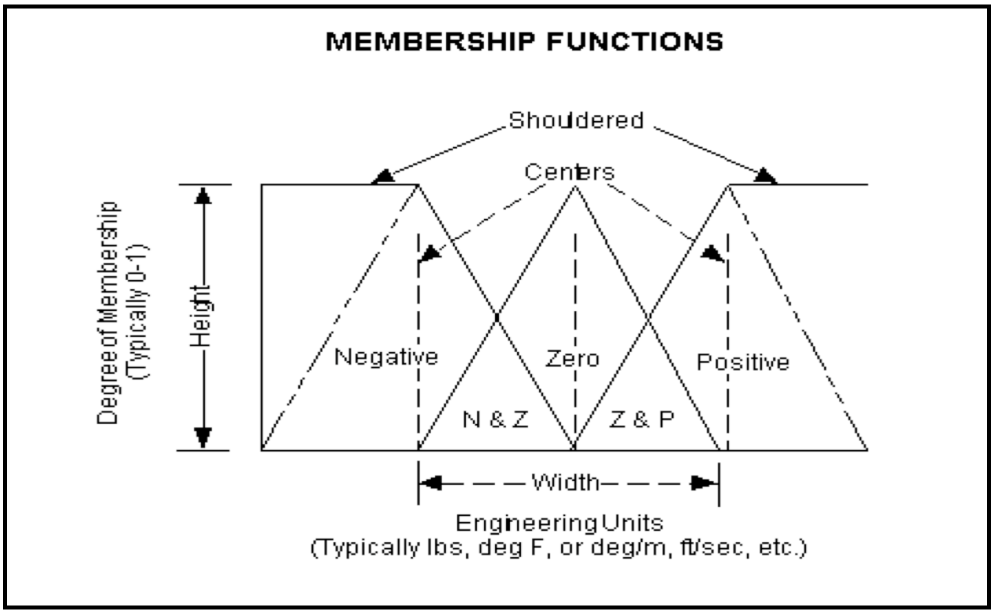


Figure 2.5. Membership Functions [4]

ERROR & ERROR-DOT FUNCTION MEMBERSHIP

The degree of membership for an "error" of -1.0 projects up to the middle of the overlapping part Of the "negative" and "zero" function so the result is "negative" membership = 0.5 and "zero" membership = 0.5. Only rules associated with "negative" & "zero" error will actually apply to the Output response. This selects only the left and middle columns of the rule matrix.
For an "error-dot" of +2.5, a "zero" and "positive" membership of 0.5 is indicated. This selects the middle and bottom rows of the rule matrix. By overlaying the two regions of the rule matrix, it can be seen that only the rules in the 2-by-2 square in the lower left corner (rules 4,5,7,8) of the rules matrix will generate non-zero output conclusions. The others have a zero weighting due to the logical AND in the rules.[4]

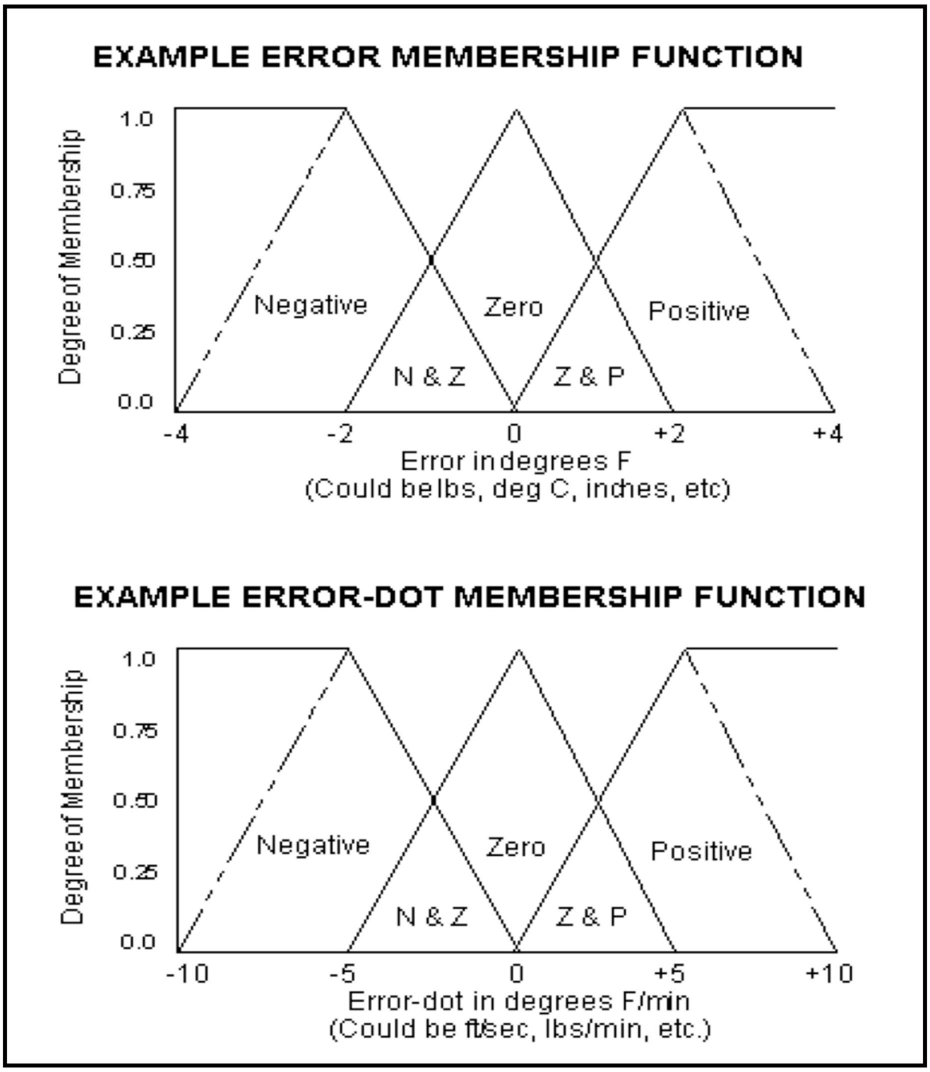


Figure 2.6 Membership functions for Error values [4]

2.2.4. Various Operations on Fuzzy Sets [4]

The standard operations perform precisely as the corresponding operations for the crisp sets when the range of membership grades is restricted to the set [0,1]. That is the standard fuzzy operations are generalizations of the corresponding classical set operations. It is now well understood, however, that they are not the only possible generalizations. For each of the three operations , there exists a broad class of functions whose members qualify as fuzzy generalizations of the classical operations as well. Since the fuzzy complement, intersection and union are not unique operations, contrary to their crisp counterparts, different functions may be appropriate to represent these operations in different contexts.

That is, not only membership functions of fuzzy sets but also operations on fuzzy sets are context dependent. The capability to determine appropriate membership functions and meaningful fuzzy operations in the context of each particular application is critical for making fuzzy set theory practically useful. Among the great variety of fuzzy complements, intersections and unions, the standard fuzzy operations possess certain properties that give them a special significance. The standard fuzzy intersection (min operator) produces for any given fuzzy sets the largest fuzzy set from among those produced by all possible fuzzy intersections. The standard fuzzy union (max operator) produces, on the contrary, the smallest fuzzy sets among the fuzzy sets produced by all possible fuzzy unions.

That is, the standard fuzzy operations occupy specific positions in the whole spectrum of fuzzy operations: the standard fuzzy intersection is the weakest intersection, while the standard fuzzy union is the strongest fuzzy union. A desirable feature of the standard fuzzy operations is their inherent prevention of the compounding of errors of the operands. If any error e is associated with the membership grades A(x) and B(x), then the maximum error associated with the membership grades of x in A' , A∩B and AUB remains e. Most of the alternative fuzzy set operations lack this characteristic. Fuzzy intersections and fuzzy unions do not cover all the operations by which fuzzy sets can be aggregated, but they cover all aggregating operations that are associative. Aggregation operations that, for any given membership grades $a_1, a_2, \ldots, a_n$, produce a membership grade that lies between min $(a_1, a_2, \ldots, a_n)$ and max$(a_1, a_2 \ldots, a_n)$ are called averaging operations. For any given Fuzzy sets, each of the averaging operations produces a fuzzy sets that is larger than any fuzzy intersection and smaller than fuzzy union (refer to Appendix B).

2.3. Various Applications of Fuzzy Logic

There are countless applications for fuzzy logic. In fact, some claim that fuzzy logic is the encompassing theory over all types of logic. The items in this list[7] are more common applications that one may encounter in everyday life.

1) Bus Time Table : How accurately do the schedules predict the actual travel time on the bus? Bus schedules are formulated on information that does not remain constant. They use fuzzy logic because it is impossible to give an exact answer to when the bus will be at a certain stop. Many unforeseen incidents can occur. There can be accidents, abnormal traffic backups, or the bus could break down. An observant scheduler would take all these possibilities into account, and include them in a formula for figuring out the approximate schedule. It is that formula which imposes the fuzziness.
2) Temperature Control: The trick in temperature control is to keep the room at the same temperature consistently. Well, that seems pretty easy, right? But how much does a room have to cool off before the heat kicks in again? There must be some standard, so the heat (or air conditioning) isn't in a constant state of turning on and off. Therein lies the fuzzy logic. The set is determined by what the temperature is actually set to. Membership in that set weakens as the room temperature varies from the set temperature. Once membership weakens to a certain point, temperature control kicks in to get the room back to the temperature it should be.
3) Auto Focus on Camera : How Does the camera even know what to focus on? Auto-focus cameras are a great revolution for those who spent years struggling with "old-fashioned" cameras. These cameras somehow figure out, based on multitudes of inputs, what is meant to be the main object of the photo. It takes fuzzy logic to make these assumptions. Perhaps the standard is to focus on the object closest to the center of the viewer. Maybe it focuses on the object closest to the camera. It is not a precise science, and cameras err periodically. This margin of error is acceptable for the average camera owner, whose main usage is for snapshots. However, most professional photographers prefer the “old-fashioned” manual focus cameras. For any errors in those photos cannot be attributed to a mechanical glitch. The decision-making in focusing a manual

camera is fuzzy as well, but a machine does not control it.

4) Medical Diagnosis: How many of and what kinds of symptoms will yield a diagnosis? How often are doctors in error? Surely everyone has seen those lists of symptoms for a horrible disease that say "if you have at least 5 of these symptoms, you are at risk". It is a hypochondriac's haven. The question is, how do doctors go from that list of symptoms to a diagnosis? Fuzzy logic. There is no guaranteed system to reach a diagnosis. If there were, we wouldn't hear about cases of medical misdiagnosis. The diagnosis can only be some degree within the fuzzy set.

5) Predicting Travel Time: This is especially difficult for driving, since there are plenty of traffic situations that can occur to slow down travel. As with bus timetabling, predicting ETA's is a great exercise in fuzzy logic. That's why it is called an estimated time of arrival. A major player in predicting travel time is previous experience. It took me six hours to drive to Philadelphia last time, so it should take me about that amount of time when I make the trip again. Unfortunately, other factors are not typically considered. Weather, traffic, construction, accidents should all be added into the fuzzy equation to deliver a true estimate.

6) Anti Lock Breaking System: It's probably something you hardly think about when you're slamming on the brakes in your car the point of an ABS is to monitor the braking system on the vehicle and release the brakes just before the wheels lock. A computer is involved in determining when the best time to do this is. Two main factors that go into determining this are the speed of the car when the brakes are applied, and how fast the brakes are depressed. Usually, the times you want the ABS to really work are when you're driving fast and slam on the brakes. There is, of course, a margin for error. It is the job of the ABS to be “smart” enough to never allow the error goes past the point when the wheels will lock. (In other words, it doesn't allow the membership in the set to become too weak.)

There are many other applications of fuzzy logic like automatic washing machine etc. All engineering disciplines have already been affected, to various degree, by the new methodological possibilities opened by fuzzy sets and fuzzy measures, by developing fuzzy controllers, which are currently the most significant systems based on fuzzy theory, electrical engineering was the first engineering discipline with in which the utility of fuzzy sets and fuzzy logic was recognized. One important category of problems in civil engineering for which fuzzy set theory has already proven useful consists of problems of assessing or evaluating existing constructions. Typical examples of these problems are the assessment of fatigue in metal structure, the assessment of quality of highway pavements and the assessment of damage in buildings, after as earthquake[7].

2.4. Neuro Fuzzy System [15]

A Neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples. Although there are a lot of different approaches, but usually use the term neuro--fuzzy system for approaches which display the following properties:

1) A neuro-fuzzy system is based on a fuzzy system which is trained by a learning algorithm derived from neural network theory. The (heuristical) learning procedure operates on local information, and causes only local modifications in the underlying fuzzy system.
2) A neuro-fuzzy system can be viewed as a 3-layer feed forward neural network. The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. Fuzzy sets are encoded as (fuzzy) connection weights. It is not necessary to represent a fuzzy system like this to apply a learning algorithm to it. However, it can be convenient, because it represents the data flow of input processing and learning within the model.
3) Neuro-fuzzy system can be always (i.e.\ before, during and after learning) interpreted As a system of fuzzy rules. It is also possible to create the system out of training data from scratch, as it is possible to initialize it by prior knowledge in form of fuzzy rules.
4) The learning procedure of a neuro-fuzzy system takes the semantically properties of the Underlying fuzzy system into account. This results in constraints on the possible modifications applicable to the system parameters.
5) A neuro-fuzzy system approximates an n-dimensional (unknown) function that is partially defined by the training data. The fuzzy rules encoded within the system represent vague samples, and can be viewed as prototypes of the training data. A

neuro-fuzzy system should not be seen as a kind of (fuzzy) expert system, and it has nothing to do with fuzzy logic in the narrow sense.

In the simplest way, a cooperative model can be considered as a preprocessor wherein ANN learning mechanism determines the FIS(Fuzzy Inference System) membership functions or fuzzy rules from the training data. Once the FIS parameters are determined, ANN(Artificial Neural networks) goes to the background. The rule based is usually determined by a clustering approach (self organizing maps) or fuzzy clustering algorithms. Membership functions are usually approximated by neural network from the training data.

In a concurrent model, ANN assists the FIS continuously to determine the required parameters especially if the input variables of the controller cannot be measured directly. In some cases the FIS outputs might not be directly applicable to the process. In that case ANN can act as a postprocessor of FIS outputs.

In a fused NF architecture, ANN learning algorithms are used to determine the parameters of FIS. Fused NF systems share data structures and knowledge representations. A common way to apply a learning algorithm to a fuzzy system is to represent it in a special ANN like architecture. However the conventional ANN learning algorithms (gradient descent) cannot be applied directly to such a system as the functions used in the inference process are usually non differentiable. This problem can be tackled by using differentiable functions in the inference system or by not using the standard neural learning algorithm.

2.5. Fuzzy Pattern Recognition

The desire to fill the gap between traditional pattern recognition methods and human behavior has lead to the development of fuzzy set theory. The fundamental role of fuzzy sets in pattern recognition, as it was stated by is to make the opaque classification schemes, usually used by a human, transparent by developing a formal, computer-realizable framework. In other words, fuzzy sets help to transfer a qualitative knowledge regarding a classification task into the relevant algorithmic structure. As a basic tool used for this interface serves a membership function. Its meaning can be interpreted differently depending on the application area of fuzzy sets. Three semantics of a membership grade can be generalised, in terms of similarity, uncertainty, or preference, respectively. A view as a degree of uncertainty is usually used in expert systems and artificial intelligence, and interpretation as a degree of preference is concerned with fuzzy optimization and decision analysis. Pattern recognition works with the first semantic, which can be formulated as follows:

Consider a fuzzy set A’, defined on the universe of discourse X, and the degree of membership u(x) of an element x in the fuzzy set A’. Then u(x) is the degree of proximity of x to prototype elements of A’ and is interpreted as a degree of similarity.

This view, besides a meaning of the semantic, shows distinctions between a membership grade and different interpretations of a probability value. Consider a pattern x and a class A. On observing x the prior probability $P(x_A) = 0.95$, expressing that the pattern x belongs to the class A, becomes a posterior probability: either $P(x_A \mid x) = 1$ or $P(x_A \mid x) = 0$. However, a degree u(x) = 0.95 to which the pattern x is similar to patterns of the class A remains unchanged after observation.

Fuzzy set theory provides a suitable framework for pattern recognition due to its ability to deal with uncertainties of the non-probabilistic type. In pattern recognition uncertainty may arise from a lack of information, imprecise measurements, random occurrences, vague descriptions, or conflicting or ambiguous information and can appear in different circumstances, for instance, in definitions of features and, accordingly, objects, or in definitions of classes. Different methods process uncertainty in various ways.

Statistical methods based on probability theory assume features of objects to be random variables and require numerical information. Feature vectors having imprecise, or incomplete, representation are usually ignored or discarded from the classification process. In contrast, fuzzy set theory can be applied for handling non-statistical uncertainty, or fuzziness, at various levels.

It can be used to represent fuzzy objects and fuzzy classes. Objects are considered to be fuzzy if at least one feature is described fuzzily, i.e. feature values are imprecise or represented as linguistic information. Classes are considered to be fuzzy, if their decision boundaries are fuzzy with gradual class membership. The combination of two representation forms of information such as crisp and fuzzy with two basic elements of pattern recognition such as object and class induces four categories of problems in pattern recognition :

- Crisp objects and crisp classes;
- Crisp objects and fuzzy classes;
- Fuzzy objects and crisp classes;
- Fuzzy objects and fuzzy classes.

The first category involves the problem of classical pattern recognition, whereas the latter three categories are concerned with fuzzy pattern recognition. It is obvious that the concept of fuzzy sets enriches the basic ideas of pattern recognition and gives rise to completely new concepts. The main reasons for the application of fuzzy set theory in pattern recognition can be summarized in the following way.

1) Fuzzy sets design an interface between linguistically formulated features and quantitative measurements. Features are represented as an array of membership values denoting the degree of possession of certain properties. Classifiers designed in such a way are often logic-oriented and reflect the conceptual layout of classification problems.
2) Class memberships of an object take their values in the interval [0, 1] and can be regarded as a fuzzy set defined on a set of classes. Thus, it is possible that an object belongs to more than one class, and a degree of membership of an object to a class expresses a similarity of this object with typical objects belonging to this specific class. Using a gradual degree of membership the most 'unclear' objects can be identified.
3) Membership functions provide an estimate of missing or incomplete knowledge.
4) A traditional distinction between supervised and unsupervised pattern recognition is enriched by admitting implicit rather than explicit object labeling or allowing for a portion of objects to be labeled. In the case of implicitly supervised classification objects are arranged in pairs according to their similarity levels.

Fuzzy set theory has given rise to a lot of new methods of pattern recognition, some of which are extensions of classical algorithms and others completely original techniques. The major groups of fuzzy methods are represented by fuzzy clustering, fuzzy rule-based, fuzzy pattern matching methods and methods based on fuzzy relations. Fuzzy techniques seem to be particularly suitable for dynamic pattern recognition when it is necessary to recognize gradual temporal changes in an object's states.

Considering the temporal development of objects, it is often difficult to assign objects to classes crisply and precisely. One can imagine, for instance, a system with two possible states as classes: proper operation and faulty operation. When the state of the system is changing measurement values express that the system operation is not more proper, but there is no error in operation yet.

This means that the observed objects do not belong to any of these classes, or belong to a small degree to both classes. The use of fuzzy set theory provides a possibility to produce fuzzy decision boundaries between classes and allows a gradual (temporally changing) membership of objects to classes. This primary advantage of fuzzy set theory is crucial for pattern recognition in general and for dynamic approach in particular, because of the possible temporal transition of objects between classes and changes of classes themselves. Therefore, the development of methods for dynamic pattern recognition in this thesis will be based on fuzzy techniques.

3.1. Control System

A control system is an arrangement of physical components designed to alter, to regulate, or to command, through a control action, another physical system so that it exhibits certain desired characteristics or behavior. Control systems are typically of two types[5]:

1) Open loop control system in which control action is independent of the physical system output.
2) Closed loop control (also known as feedback control systems) in which the control action depends on the physical system output.

Example of open loop control system are a toaster, in which the amount of heat is set by human and an automatic washing machine, in which the controls for water temperature, spin cycle time and so on are preset by human. In both these cases the control actions are not a function of the output of the toaster and washing machine.
Example of Closed loop control are room temperature thermostat, which senses the room temperature and activates a heating or cooling unit when a certain threshold temperature is reached, and an autopilot mechanism, which makes automatic courses corrections to an airplane when heading or altitude deviations from certain preset values are sensed by instruments in the planes cockpit.
In order to control any physical variable, we must first measure it. The system for measurement of the controlled signal is called a sensor. The physical system under control is called a plane. In a closed loop control system, certain forcing signals of the system are determined by the responses of the system. To obtain satisfactory responses and characteristics for the closed loop control system, it is necessary to connect an additional system, known as compensator or controller, into the loop. Control systems are sometimes divided into two classes. If the object of the control system is to maintain a physical variable at some constant value in the presence of disturbances, the system is Called a regulatory type of control or regulator. The room

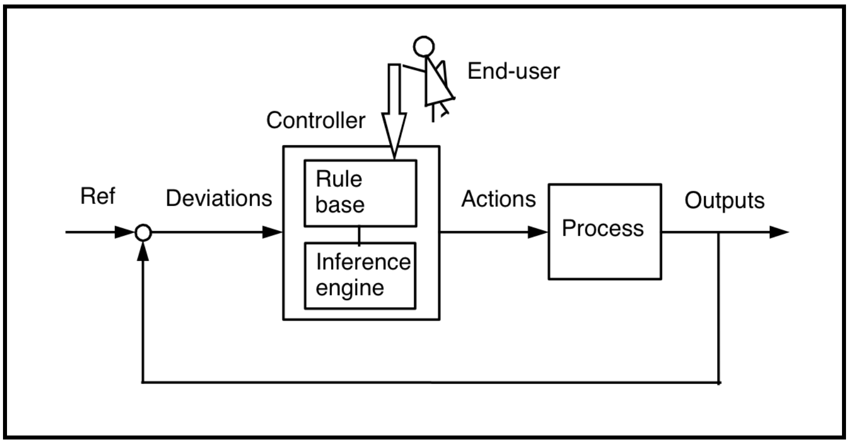


Figure 3.1. Example of Direct Control System [5]

temperature control and an autopilot are examples of regulatory controllers. The second class of control system is tracking controllers. In this second class of controllers, a physical variable is required to follow or tack some desired time function. An example of this type of systems is an automatic aircraft landing system, in which the aircraft follows a ramp to the desired touchdown point.

The control system is stated as follows[5]: the output or response of the physical system under control is adjusted as required by error signal. The error signal is the difference

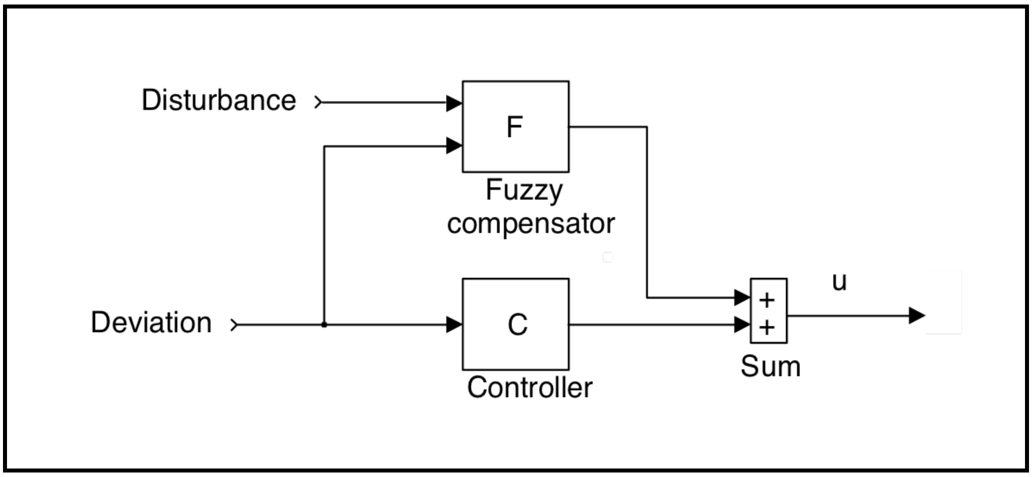


Figure 3.2. Example of Feed Forward Control System [5]

between the actual response of the plant, as measured by the sensor system and the desired response as specified by a reference input.

Proportional control :When controlling the temperature of an industrial furnace, it is usually better to control the opening of the fuel valve in proportion to the current needs of the furnace. This helps avoid thermal shocks and applies heat more effectively.

Proportional negative-feedback systems are based on the difference between the required set point (SP) and measured value (MV) of the controlled variable. This difference is called the error. Power is applied in direct proportion to the current measured error, in the correct sense so as to tend to reduce the error (and so avoid positive feedback). The amount of corrective action that is applied for a given error is set by the gain or sensitivity of the control system.

At low gains, only a small corrective action is applied when errors are detected: the system may be safe and stable, but may be sluggish in response to changing conditions; errors will remain uncorrected for relatively long periods of time: it is over-damped. If the proportional gain is increased, such systems become more responsive and errors are dealt with more quickly. There is an optimal value for the gain setting when the overall system is said to be critically damped. Increases in loop gain beyond this point will lead to oscillations in the MV; such a system is under-damped.

PID control: Apart from sluggish performance to avoid oscillations, another problem with proportional-only control is that power application is always in direct proportion to the error. In the example above we assumed that the set temperature could be maintained with 50% power. What happens if the furnace is required in a different application where a higher set temperature will require 80% power to maintain it? If the gain was finally set to a 50° PB, then 80% power will not be applied unless the furnace is 15° below set point, so for this other application the operators will have to remember always to set the set point temperature 15° higher than actually needed. This 15° figure is not completely constant either: it will depend on the surrounding ambient temperature, as well as other factors that affect heat loss from or absorption within the furnace.

To resolve these two problems, many feedback control schemes include mathematical extensions to improve performance. The most common extensions lead to proportional-integral-derivative control, or PID control.

Derivative action: The derivative part is concerned with the rate-of-change of the error with time: If the measured variable approaches the set point rapidly, then the actuator is backed off early to allow it to coast to the required level; conversely if the measured value begins to move rapidly away from the set point, extra effort is applied in proportion to that rapidity to try to maintain it.

Derivative action makes a control system behave much more intelligently. On systems like the temperature of a furnace, or perhaps the motion-control of a heavy item like a gun or camera on a moving vehicle, the derivative action of a well-tuned PID controller can allow it to reach and maintain a set point better than most skilled human operators could. If derivative action is over-applied, it can lead to oscillations too. An example would be a temperature that increased rapidly towards SP, then halted early and seemed to "shy away" from the set point before rising towards it again.

Integral action: The integral term magnifies the effect of long-term steady-state errors, applying ever-increasing effort until they reduce to zero. In the example of the furnace above working at various temperatures, if the heat being applied does not bring the furnace up to set point, for whatever reason, integral action increasingly moves the proportional band relative to the set point until the time-integral of the MV error is reduced to zero and the set point is achieved.

3.2. Assumptions in a Fuzzy Control System

A number of assumptions are implicit in a fuzzy control system design. Six basic assumptions are commonly made whenever a fuzzy logic based control based control policy is selected.

1) The plant is observable and controllable. State, input and output variable are usually available for observations and measurements.
2) There exists a body of knowledge comprised of a set of expert production linguistic rules, engineering common sense, intuition, a set of input/output measurements date or an analytic model that can be fuzzified and from which rules can be extracted.
3) A solution exists.
4) The control engineer is looking for a good enough solution not necessarily the optimum one.
5) We will design a controller to the best of our knowledge and within acceptable range of precision.
6) The problem of stability and optimality are still open problems in fuzzy controller design.

Fuzzy rule based expert models can also be used to obtain acceptable approximations for the functions in case of a system identification problem.

A fuzzy production rule system consists of four structures.

1) A set of rules that represents the policies and heuristic strategies of the expert decision maker.
2) A set of input data accessed immediately prior to the actual decision.
3) A method for evaluating any proposed action in terms of its conformity to the expressed rules, given the available data.
4) A method of generating promising actions and for determining when to stop searching for betters ones.

The input data, rules and output action, or consequences are generally fuzzy sets expressed as membership functions defined on a proper space. The method used for the evaluation of rules is known as approximate reasoning or interpolative reasoning, and is commonly used by composition of fuzzy relations applied to a fuzzy relation equation.

The control surface, which relates the control action to the measured state or output variables, is obtained using these four structures. It is then sampled at a finite number of points, depending on the required resolutions, and a look up table is constructed. The look up table thus formed could be downloaded into read only memory chip and would constitute a fixed controller for the plant. Here is no design procedure in fuzzy control such as root-locus design, frequency response design, pole placement design, or stability margins, because the rules are often non-linear.

3.3. Fuzzy Logic Controller [5]

While it is relatively easy to design a PID controller, the inclusion of fuzzy rules creates many extra design problems, the approach here is based on a three step design procedure that builds on PID control:

1) Start with a PID Controller.
2) Insert an equivalent linear fuzzy controller.
3) Make it gradually non linear.

Fuzzy controllers are used to control consumer products, such as washing machines, video cameras, and rice cookers, as well as industrial processes, such as cement kilns, underground trains, and robots. Fuzzy control is a control method based on fuzzy logic. Just as fuzzy logic can be described simply as ''computing with words rather than numbers''; fuzzy control can be described simply as ''control with sentences rather than equations''. A fuzzy controller can include empirical rules, and that is especially useful in operator-controlled plants.

Take for instance a typical fuzzy controller.

1) If error is Neg and change in error is Neg then output is NB.
2) If error is Neg and change in error is Zero then output is NM.

The collection of rules is called a rule base. The rules are in the familiar if-then format, and formally the if-side is called the condition and the then-side is called the conclusion (more often, perhaps, the pair is called antecedent- consequent or premise- conclusion). The input value ''Neg'' is a Linguistic term short for the word Negative the output value ''NB'' stands for Negative Big and ''NM'' for Negative Medium. The computer is able to execute the rules and compute a control signal depending on the measured inputs error and change in error.

In a rule based controller the control strategy is stored in a more or less natural language. The control strategy is isolated in a rule base opposed to an equation based description. A rule based controller is easy to understand and easy to maintain for a non-specialist end-user.

Fuzzy rules are also used to correct tuning parameters in parameter adaptive control schemes. If a nonlinear plant changes operating point, it may be possible to change the parameters of the controller according to each operating point. This is called gain scheduling since it was originally used to change process gains. A gain scheduling controller contains a linear controller whose parameters are changed as a function of the operating point in a preprogrammed way. It requires thorough knowledge of the plant, but it is often a good way to compensate for

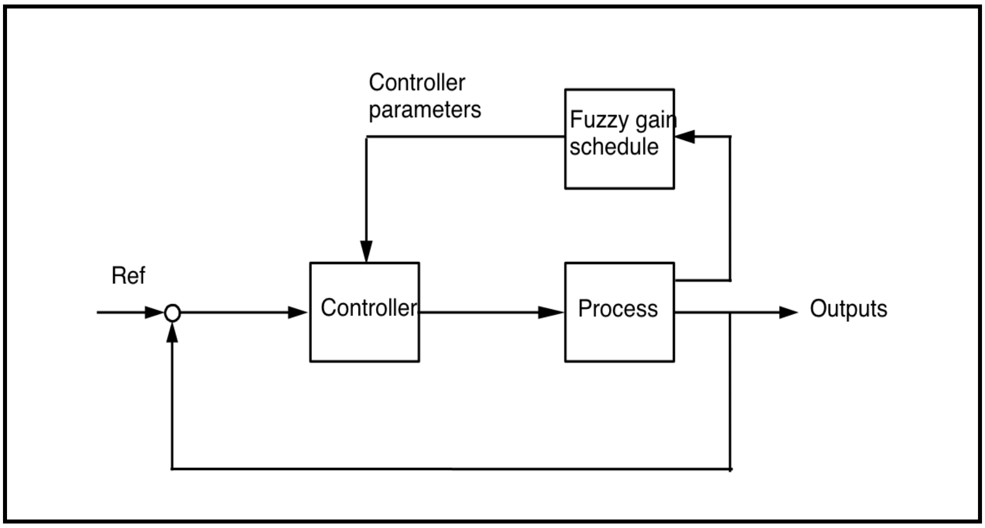


Figure 3.3. Fuzzy Parameter Adaptive Control System[5]

nonlinearities and parameter variations. Sensor measurements are used as Scheduling variables that govern the change of the controller parameters, often by means of a table look-up.

3.4. Structure of Fuzzy Controller [5]

There are specific components characteristic of a fuzzy controller to support a design procedure. In the block diagram in Fig. 3.4, the controller is between a preprocessing block and a post-processing block. The following explains the diagram block by block.

3.4.1. Preprocessing

The inputs are most often hard or crisp measurements from some measuring equipment, rather than linguistic. A preprocessor, the first block in Fig. 3.4, conditions the measurements before they enter the controller. Examples of preprocessing are:

1) Quantisation in connection with sampling or rounding to integers;
2) Normalization or scaling onto a particular, standard range;
3) Filtering in order to remove noise;
4) Averaging to obtain long term or short term tendencies;

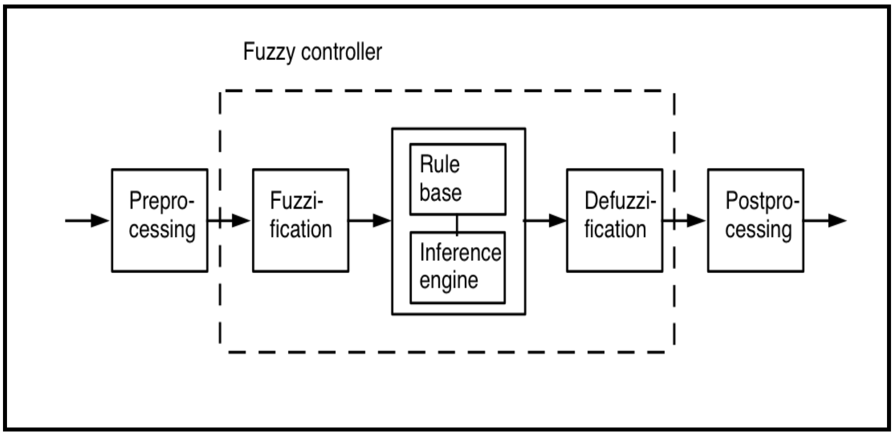


Figure 3.4. Block Diagram of Fuzzy Controller [5]

5) A combination of several measurements to obtain key indicators; and
6) Differentiation and integration or their discrete equivalences.

A quantiser is necessary to convert the incoming values in order to find the best level in a discrete universe. Assume, for instance, that the variable error has the value 4.5, but the universe is u= (-5, -4, . . . , 0, . . . , 4, 5). The quantiser rounds to 5 to fit it to the nearest level. Quantisation is a means to reduce data, but if the quantisation is too coarse the controller may oscillate around the reference or even become unstable.

Nonlinear Scaling is an option (Fig. 3.5). For a controller the operator is asked to enter three

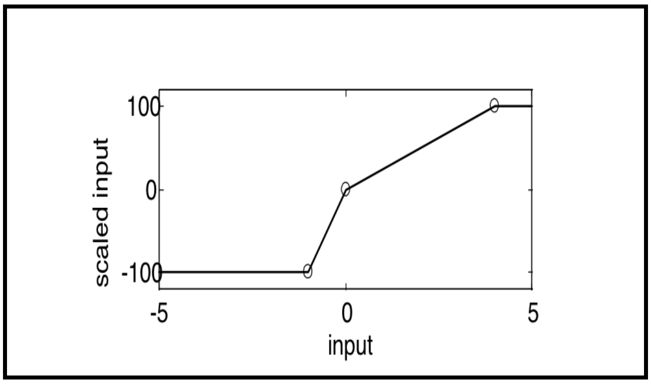


Figure 3.5. Non-Linear Scaling of an Input Measurement [5]

typical numbers for a small, medium and large measurement respectively. They become break-points on a curve that scales the incoming measurements (circled in the figure). The overall effect can be interpreted as a distortion of the primary fuzzy sets. It can be confusing with both scaling and gain factors in a controller, and it makes tuning difficult.

When the input to the controller is error, the control strategy is a static mapping between input and control signal. A dynamic controller would have additional inputs, for example derivatives, integrals, or previous values of measurements backwards in time. These are created in the preprocessor thus making the controller multi-dimensional, which requires many rules and makes it more difficult to design.

The preprocessor then passes the data on to the controller.

3.4.2. Fuzzification

The first block inside the controller is Fuzzification, which converts each piece of input data to degrees of membership by a look up in one or several membership functions. The fuzzification block thus matches the input data with the conditions of the rules to determine how well the condition of each rule matches that particular input instance. There is a degree of membership for each linguistic term that applies to that input variable.

3.4.3 Rule Base

The rules may use several variables both in the condition and the conclusion of the rules. The controller scan therefore is applied to both multi-input multi-output (MIMO) problems and single-input single-output (SISO) problems. The typical SISO problem is to regulate A control signal based on an error signal. The controller may actually need both the error, change in error, and accumulated error as inputs, but we will call it single-loop control, because in principle all three are formed from the error measurement. To simplify, here we assumes that the control objective is to regulate some process output around a prescribed set point or reference. The objective is thus limited to single-loop control.

Rule Formats: Basically a linguistic controller contains rules in the if- then format, but they can be presented in different formats. In many systems, the rules are presented to the end-user in a format similar to the one below,

1) If error is Neg and change in error is Neg then output is NB.
2) If error is Neg and change in error is Zero then output is NM.
3) If error is Neg and change in error is Pos then output is Zero.
4) If error is Zero and change in error is Neg then output is NM.
5) If error is Zero and change in error is Zero then output is Zero.
6) If error is Zero and change in error is Pos then output is PM.
7) If error is Pos and change in error is Neg then output is Zero.
8) If error is Pos and change in error is Zero then output is PM.
9) If error is Pos and change in error is Pos then output is PB.

The names = Zero, Pos, Neg are labels of fuzzy sets as well as NB, NM,PB AND PM (negative big, negative medium, positive big, and positive medium respectively). The same set of rules could be presented in a relational format, a more compact representation.

Error	Change in error	Output
Neg	Pos	Zero
Neg	Zero	NM
Neg	Neg	NB
Zero	Pos	PM
Zero	Zero	Zero
Zero	Neg	NM

Pos	Pos	PB
Pos	Zero	PM
Pos	Neg	Zero

The top row is the heading, with the names of the variables. It is understood that the two leftmost columns are inputs, the rightmost is the output, and each row represents a rule. This format is perhaps better suited for an experienced user who wants to get an overview of the rule base quickly. The relational format is certainly suited for storing in a relational database. It should be emphasized, though, that the relational format implicitly assumes that the connective between the inputs are always logical and logical or for that matter as long as it is the same operation for all rules and not a mixture of connectives. Incidentally, a fuzzy rule with an RU combination of terms Can be converted into an equivalent and combination of terms using laws of logic (DeMorgan's laws among others). A third format is the tabular linguistic format.

		Change in error		
		Neg	Zero	Pos
Error	Neg	NB	NM	Zero
	Zero	NM	Zero	PM
	Pos	Zero	PM	PB

This is even more compact. The input variables are laid out along the axes, and the output variable is inside the table. In case the table has an empty cell, it is an indication of a missing rule, and this format is useful for checking completeness. When the input variables are error and change in error, as they are here, that format is also called a Linguistic phase plane. In case there are n > 2 input variables involved, the table grows to an n-dimensional array; rather user-unfriendly.

To accommodate several outputs, a nested arrangement is conceivable. A rule with several outputs could also be broken down into several rules with one output. Lastly, a graphical format which shows the fuzzy membership curves is also possible. This graphical user-interface can display the inference process better than the other formats, but takes more space on a monitor.

Connectives: In mathematics, sentences are connected with the words and, or, if-then (or implies, and if and only if, or modifications with the word not. These five are called connectives. It also makes a difference how the connectives are implemented. The most prominent is probably multiplication for fuzzy AND instead of minimum. So far most of the examples have only contained AND operations, but a rule like ‘‘If error is very neg and not zero or change in error is zero then ...’’ is also possible.

The connectives And and OR are always defined in pairs, for example,

a AND b = min (a,b) minimum

a OR b = max(a,b) maximum

or

a AND b = a * b algebraic product

a or b= a + b – a * b algebraic or probabilistic sum

Modifiers: A linguistic modifier, is an operation that modifies the meaning of a term. For example, in the sentence ‘‘very close to 0’’, the word very modifies close to 0 which is a fuzzy set. A modifier is thus an operation on a fuzzy set. The modifier very can be defined as squaring the subsequent membership function, that is

$$\textbf{Very a} = \mathbf{a^2}$$

Some examples of other modifiers are

$$\textbf{Extremely a} = \mathbf{a^3}$$

$$\textbf{Slightly a} = \mathbf{a^{1/3}}$$

Somewhat a = moreorless a and not slightly a

A whole family of modifiers is generated by a^p where p is any power between zero and infinity. With $p = \infty$ the modifier could be named exactly, because it would suppress all memberships lower than 1.0.

Universes: Elements of a fuzzy set are taken from a universe of discourse or just universe. The universe contains all elements that can come into consideration. Before designing the membership functions it is necessary to consider the universes for the inputs and outputs. Take for example the rule

If error is Neg and change in error is Pos then output is 0

Naturally, the membership functions for Neg and Pos must be defined for all possible values of error and change in error and a standard universe may be convenient. Another consideration is whether the input membership functions should be continuous or discrete. A continuous membership function is defined on a continuous universe by means of parameters. A discrete membership function is defined in terms of a vector with a finite number of elements. In the latter case it is necessary to specify the range of the universe and the value at each point. The choice between fine and coarse resolution is a trade off between accuracy, speed and space demands. The quantiser takes time to execute, and if this time is too precious, continuous membership functions will make the quantiser obsolete.

Membership Functions: Every element in the universe of discourse is a member of a fuzzy set to some grade, maybe even zero. The grade of membership for all its members describes a fuzzy set, such as Neg. In fuzzy sets elements are assigned a grade of membership, such that the transition from membership to non-membership is gradual rather than abrupt. The set of elements that have a non-zero membership is called the support of the fuzzy set. The function that ties a number to each element x of the universe is called the membership function μ_x.

The designer is inevitably faced with the question of how to build the term sets. There are two specific questions to consider:

1) How does one determine the shape of the sets?
2) How many sets are necessary and sufficient?

For example, the error in the position controller uses the family of terms Neg, Zero, Pos. According to fuzzy set theory the choice of the shape and width is subjective, but a few rules of thumb apply.

1) A term set should be sufficiently wide to allow for noise in the measurement.
2) A certain amount of overlap is desirable; otherwise the controller may run into poorly defined states, where it does not return a well-defined output.

Preliminary answer to questions 1 and 2 is that the necessary and sufficient number of sets in a family depends on the width of the sets, and vice versa. A solution could be to ask the process operators to enter their personal preferences for the membership curves; but operators also find it difficult to settle on particular curves.

Triangular sets: All membership functions for a particular input or output should be symmetrical triangles of the same width. The leftmost and the rightmost should be shouldered ramps.

The Overlap should be over 50%: The widths should initially be chosen so that each value of the universe is a member of at least two sets, except possibly for elements at the extreme ends. If, on the other hand, there is a gap between two sets no rules fire for values in the gap. Consequently the controller function is not defined.

Membership functions can be f lat on the top, piece-wise linear and triangle shaped, rectangular, or ramps with horizontal shoulders.

Fig. 3.6 shows some typical shapes of membership functions. Strictly speaking, a fuzzy set D is a collection of ordered pairs

A={(x, u(x))}

Item x belongs to the universe and u(x) is its grade of membership in D. A single pair ={(x, u(x))}is a fuzzy Singleton ; singleton output means replacing the fuzzy sets in the conclusion by numbers (scalars). For example

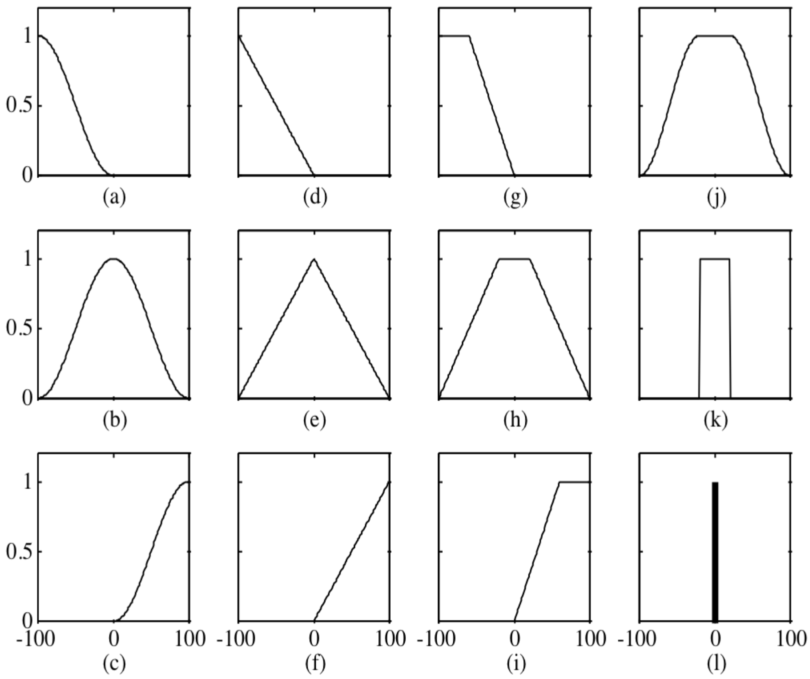


Figure 3.6. Examples of membership functions. Read from top to bottom, left to right: (a) s function, (b) Π function, (c) z function, (d-f) triangular versions, (g-i) trapezoidal versions, (j) flat Π function, (k) rectangle, (l) singleton. [5]

1) If error is Pos then output is 10 volts.
2) If error is Zero then output is 0 volts.
3) If error is Neg then output is -10 volts.

There are at least three advantages to this:

1) The computations are simpler;
2) It is possible to drive the control signal to its extreme values; and
3) It may actually be a more intuitive way to write rules.

The scalar can be a fuzzy set with the singleton placed in a proper position. For example - 10 Volts would be equivalent to the fuzzy set (0, 0, 0, 0, 1), defined on the universe (-10, -5, 0, 5 ,10) volts.

3.4.4. Inference Engine

For each rule, the inference engine looks up the membership values in the condition of the rule.

Aggregation: The Aggregation operation is used when calculating the degree of fulfillment or firing strength α_k of the condition of a rule k. A rule, say rule 1, will generate fuzzy membership value u_{e1} coming from the error and a membership value u_{ce1} coming from the change in error management. The aggregation is their combination,

$$\mathbf{u_{e1} \text{ AND } u_{ce1}}$$

Similarly for the other rules. Aggregation is equivalent to fuzzification, when there is only one input to the controller. Aggregation is sometimes also called Fulfillment of the rule or Firing strength.

Activation: The Activation of a rule is the deduction of the conclusion, possibly reduced by its firing strength. Thickened lines in the third column indicate the firing strength of each rule. Only the thickened part of the singletons are activated, and PLQ or product (*) is used as the activation operator. It makes no difference in this case, since the output membership functions are singletons, but in the general case of s, Π and z function, the multiplication scales the membership curves, thus preserving the initial shape, rather than clipping them as the PLQ operation does. Both methods work well in general, although the multiplication results in a slightly smoother control signal.

A rule n can be weighted a priori by a weighting factor W_n € [0,1] , which is its degree of confidence. In that case the firing strength is modified to

$$\alpha^*_k = w_k * \alpha_k$$

The degree of confidence is determined by the designer, or a learning program trying to adapt the rules to some input-output relationship.

Accumulation: All activated conclusions are accumulated, using the max operation. Alternatively, Sum accumulation counts overlapping areas more than once.
Singleton output and Sum accumulation results in the simple output.

$$\alpha_1 * s_1 + \alpha_2 * s_2 + \ldots + \alpha_n * s_n$$

The alpha's are the firing strengths from the n rules and $s_1 \ldots s_n$ are the output singletons. Since this can be computed as a vector product, this type of inference is relatively fast in a matrix oriented language.
There could actually have been several conclusion sets. An example of a one-input- two-outputs rule is ''If h_d is D then r4 is x and r5 is y. The inference engine can treat two (or several) columns on the conclusion side in parallel by applying the firing strength to both conclusion sets. In practice, one would often implement this situation as two rules rather than one, that is, ''if h_d is D then r4 is x ''If h_d is D then r5 is y.

3.4.5 Defuzzification

The resulting fuzzy set must be converted to a number that can be sent to the process as a control signal. This operation is called defuzzification and The resulting fuzzy set is thus defuzzified into a crisp control signal. There are several defuzzification methods.

Centre of Gravity(COG): The crisp output value x () is the abscissa under the centre of gravity of the fuzzy set,

$$u = \frac{\sum_i \mu(x_i) x_i}{\sum_i \mu(x_i)}$$

Here x_i is a running point in a discrete universe, and $\mu(x_i)$ is its membership value in the membership function. The expression can be interpreted as the weighted average of the elements in the support set. For the continuous case, replace the summations by integrals. It is a much used method although its computational complexity is relatively high. This method is also called Centriod of area.

Centre of Gravity Method for Singletons: If the membership functions of the conclusions are singletons , the output value is

$$u = \frac{\sum_i \mu(s_i) s_i}{\sum_i \mu(s_i)}$$

Here s_i is the position of singleton l in the universe, and $\mu(s_i)$, is equal to the firing strength. This method has a relatively good computational complexity, and μ is differentiable with respect to the singletons s_i , which is useful in neuro fuzzy systems.

Mean of Maxima: An intuitive approach is to choose the point with the strongest possibility, i.e. maximal membership. It may happen, though, that several such points exist, and a common practice is to take the Mean of Maxima (MOM). This method disregards the shape of the fuzzy set, but the computational complexity is relatively good.

Leftmost maximum(LM) and Rightmost maximum(RM): Another possibility is to choose the leftmost maximum (LM), or the rightmost maximum (RM). In the case of a robot, for instance, it must choose between left or right to avoid an obstacle in front of it. The Defuzzifier must then choose one or the other, not something in between. These methods are indifferent to the shape of the fuzzy set, but the computational complexity is relatively small.

3.4.6. Post processing

Output scaling is also relevant. In case the output is defined on a standard universe this must be scaled to engineering Units. for instance, volts, meters, or tons per hour. An example is the scaling from the standard universe [-1, 1] to the physical units [-10, 10] volts.

The post processing block often contains an output gain that can be tuned, and sometimes also an integrator.

3.5. Table Based Controller

If the universes are discrete, it is always possible to calculate all thinkable combinations of inputs before putting the controller into operation. In a table based controller, the relation between all input combinations and their corresponding outputs are arranged in a table. With two inputs and one output, the table is a two-dimensional look-up table. With three inputs the table becomes a three-dimensional array. The array implementation improves execution speed, as the run-time inference is reduced to a table look-up which is a lot faster, at least when the correct entry can be found without too much searching. Below is a small example of a look-up table corresponding to the rulebase (2) with the membership functions. A typical application area for the table based controller is where the inputs to the controller are the error and the change in error. The controller can be embedded in a larger system, a car for instance, where the table is downloaded to a table look-up mechanism.

		change in error				
		-100	-50	0	50	100
	-100	-200	-160	-100	-40	0
	-50	-160	-121	-61	0	40
error	0	-100	-61	0	61	100
	50	-40	0	61	121	160
	100	0	40	100	160	200

Table Regions: Referring to the look-up table , a negative value of error implies that the process output y is above the reference Ref , because the error is computed as error= Ref – y. A positive value implies a process output below the reference. A negative value of change in error means that the process output increases while a positive value means it decreases.

Certain regions in the table are especially interesting. The centre of the table corresponds to the case where the Ref is zero; the process is on the reference. Furthermore, the change in error is zero here, so the process stays on the reference. This position is the stable point where the process has settled on the reference. The anti-diagonal (orthogonal to the main diagonal) of the table is zero; those are all the pleasant states, where the process is either stable on the reference or approaching the reference. Should the process move away a little from the zero diagonal, due to noise or a disturbance, the controller will make small corrections to get it back. In case the process is far from the reference and also moving away from it, we are in the upper left and lower right corners.

The numerical values on the two sides of the zero diagonal do not have to be anti-symmetric; they can be any values, reflecting asymmetric control strategies. During a response with overshoot after a positive step in the reference, a plot of the point (error, change in error), will follow a trajectory in the table which spirals clockwise from the lower left corner of the table towards the centre. It is similar to a phase plane trajectory, where a variable is plotted against its derivative. A clever designer may adjust the numbers manually during a tuning session to obtain a particular response.

3.6. Input Output Mapping

Two inputs and one output results in a two dimensional table, which can be plotted as a surface for visual inspection. The relationship between one input and one output can be plotted as a graph. These plots are a design aid when selecting membership functions and constructing rules.

The shape of the surface can be controlled to a certain extent by manipulating the membership functions. In order to see this clearly, we will use the one-input-one-output case (without loss of generality). The fuzzy proportional rule base produced the different mappings.

1) If error is Neg then output is Neg.
2) If error is Zero then output is Zero
3) If error is Pos then output is Pos

The rightmost column is the input-output mapping, and each row is a different controller. The controllers have the input families in the if column and the output families in the then column. The results depend on the choice of design parameters, which in this case are the following: the * operation for activation. because it is continuous, the max operation for Accumulation since it corresponds to set union, and centre of gravity for defuzzification, since it is continuous, unambiguous, and it degenerates to COGS in the case of singleton output. If there had been two or more inputs, the * operation for AND would be chosen since it is continuous. The following comments relate to the figure, row by row:

1) Triangular sets in both condition and conclusion result in a winding input-output mapping. Compared to a linear controller (dotted line) the gain of the fuzzy controller varies. A slight problem with this controller is that it does not use the full output range; it is impossible to drive the output to 100%. Another problem is that the local gain is always equal or lower than the linear controller.
2) Singleton outputs eliminate the problem with the output range. The set Pos corresponds to 100, Zero to 0, and NEG to -100. The input terms are the same as before. Now the input-output mapping is linear.
3) Flat input sets produce flat plateaus and large gains far away from the reference. This is similar to a dead zone with saturation. Increasing the width of the middle term results in a wider plateau around the reference. Less overlap between neighboring
sets will result in steeper slopes.
4) If the sharp corners cause problems, they are removed by introducing nonlinear input sets. The input-output relationship is now smooth.
5) Adding more sets only makes the mapping more bumpy.
6) On the other hand with more sets it is easier to stretch the reference plateau by moving the singletons about.

The traffic signals affect the life of virtually everyone everyday. People accept and in some cases demand that traffic signals ensure safety and mobility. Traffic management is an integral part of urban management and transport planners have traditionally concentrated on the movement of vehicles **as** the major aim of this process. The general main goal is that the number of stops has to be minimized at the level of transportation system, while at the level of one intersection the delays have to be minimized. In traffic signal control several traffic flows compete from the same time and space, and different priorities are often set to different traffic flows or vehicle groups. Normally, the optimization includes several simultaneous criteria, like the average delays, maximum queue lengths and percentage of stopped vehicles.

Controlling timing of a traffic signal means making the following evaluation constantly, whether to terminate the current phase/signal-group and change to the next most appropriate phase/signal-group, or extend the current phase/signal-group. In other words, a controller continuously (or at regular intervals) gathers information and evaluates the status of each approach and takes the most appropriate option. So it is very likely that fuzzy control is very competitive in complicated real intersections where the use of traditional optimization methods is problematic.

4.1 Need of Traffic Signal Controller

As the number of vehicle in urban areas is ever increasing, it has been a major concern of city authorities to facilitate effective control of traffic flows in urban areas. Especially in rush hours, even a short period of poor control at traffic signals may result in a long time traffic jam causing a chain of delays in traffic flows. The total amount of accumulated delay time in a city due to waiting at signal stops is enormous if it is counted on an annual basis. To reduce the waiting time of vehicles at traffic signals is to reduce consumption of fuel and man-hours, thus it is significant to control the traffic signals in an effective manner.

As an effective control way, traffic signal controller is playing more and more important roles in modern management and control of urban traffic. Before installing a traffic signal at an intersection, established minimum criteria must be satisfied. A review includes: The amount of Vehicular and pedestrian traffic; the need to provide interruption to the major flow for side street vehicles and pedestrians, special conditions such as hills and curves; the accident history of the intersection and the proximity of schools. Signals offer maximum control at intersections. They relay messages of both what to do and what not to do. The primary function of any traffic signal is to assign right-of-way to conflicting movements of traffic at an intersection. This is done by permitting conflicting streams of traffic to share the same intersection by means of time separation.

4.1.1. Literature Survey

In order to design a Fuzzy controller, I have gone through various professional websites like IEEE, Springer link. I have studied various papers, journals to first get the knowledge of fuzzy logic then fuzzy controller after that what are the various researches going on in fuzzy logic. All the major Paper, journals studied for implementing Fuzzy controller is listed in references. Brief details of material studied is as below:

Using the conventional approach for controller, first step is to understand the physical system and its control requirements. Based on this understanding, second step is to develop a model which includes the plant, sensors and actuators. The third step is to use linear control theory in order to determine a simplified version of the controller, The fourth step is to develop an algorithm for the simplified controller. The last step is to simulate the design including the effects of non-linearity, noise, and parameter variations. If the performance is not satisfactory we need to modify our system modeling, re-design the controller, re-write the algorithm and re-try. With Fuzzy Logic the first step is to understand and characterize the system behavior by using our knowledge and experience. The second step is to directly design the control algorithm using fuzzy rules, which describe the principles of the controller's regulation in terms of the relationship between its inputs and outputs. The last step is to simulate and debug the design. If the performance is not satisfactory we only need to modify some fuzzy rules and re-try.

Fuzzy Logic reduces the design development cycle, Fuzzy Logic improves time to market, Fuzzy Logic simplifies design complexity, Fuzzy Logic reduces hardware costs, Fuzzy Logic simplifies implementation. [2]

Fuzzy logic is being applied in rule based automatic controllers, and Zadeh proposed a grade of membership, such that the transition from membership to non-membership is gradual rather than abrupt. Various set operations, modifiers, set relations, connectives, implication and inference rule are defined [3].

There is a unique membership function associated with each input parameter. The membership functions associate a weighting factor with values of each input and the effective rules. These weighting factors determine the degree of influence or degree of membership (DOM) each active rule has. By computing the logical product of the membership weights for each active rule, a set of fuzzy output response magnitudes are produced. All that remains is to combine and defuzzify these output responses [4].

In a fuzzy controller the data passes through a preprocessing block, a controller, and a post processing block. Preprocessing consists of a linear or non-linear scaling as well as a quantization in case the membership functions are discretised (vectors); if not, the membership of the input can just be looked up in an appropriate function. When designing the rule base, the designer needs to consider the number of term sets, their shape, and their overlap. The rules themselves must be determined by the designer, unless more advanced means like self-organization or neural networks are available. There is a choice between multiplication and minimum in the activation. There is also a choice regarding defuzzification; the post processing consists in a scaling of the output. In case the controller is incremental, post processing also includes an integration [5].

Various operations on fuzzy sets as well as techniques in detail are Fuzzy complements, Nested structure of the basic classes of fuzzy complements, Fuzzy intersections, Fuzzy unions, Duality of fuzzy set operations, Axiomatic requirements, Aggregation operations, Averaging operations, Norm operations, Ordered Weighted Averaging Operations – OWA, Decision as a fuzzy set.

1) Geometric mean gives the best prediction of the empirical data, among the operations tested.
2) Even if the criteria are independent, and the corresponding fuzzy sets do not "interact", the aggregation operation itself can "put them into interaction".
3) Every concrete decision set may require a specific aggregation operation.[6]

Various Applications of fuzzy logic are: Bus Time Table : How accurately do the schedules predict the actual travel time on the bus?, Temperature Control: The trick in temperature control is to keep the room at the same temperature consistently, Auto Focus on Camera : How Does the camera even know what to focus on?, Medical Diagnosis: How many of what kinds of symptoms will yield a diagnosis?, Predicting Travel Time: This is especially difficult for driving, since there are plenty of traffic situations that can occur to slow down travel, Anti Lock Breaking System: It's probably something you hardly think about when you're slamming on the brakes in your car the point of an ABS is to monitor the braking system on the vehicle and release the brakes just before the wheels lock.

There are many other applications of fuzzy logic like automatic washing machine etc. All engineering disciplines have already been affected, to various degree, by the new methodological possibilities opened by fuzzy sets and fuzzy measures, by developing fuzzy controllers, which are currently the most significant systems based on fuzzy theory, electrical engineering was the first engineering discipline with in which the utility of fuzzy sets and fuzzy logic was recognized. One important category of problems in civil engineering for which fuzzy set theory has already proven useful consists of problems of assessing or evaluating existing constructions. Typical examples of these problems are the assessment of fatigue in metal structure, the assessment of quality of highway pavements and the assessment of damage in buildings, after as earthquake.[7]

One long standing as well as challenging problem is to apply traffic signal control so as to

1) Maximize the efficiency of existing traffic systems without new road construction,
2) Reduce the vehicle delay or equivalently queue length.
3) Minimize the air and noise pollution.

There are two different types of traffic signal control:

1) Fixed Time Control: Fixed time signal control is based on the historical traffic data, assuming traffic conditions are unchanged in the time periods. **A** local field signal controller uses the preset timing plan to control intersection signals. Since the traffic demands are changing all time in the real world, the fixed time signal control is obviously not satisfactory.
2) Adaptive Signal Control: As an improvement over fixed time control, this control scheme uses the online data that are collected from sensors and detectors installed at an intersection. The operation of signal control depends on the actual traffic situation.

As adaptive signal controller is complex and very expensive as compared to fixed time control. In Fixed time control, time was already fixed for a particular lane irrespective of traffic load on road. Controller has to just change the signal for a fixed amount of time, but an adaptive controller can do two things

1) On roads, at different timing different traffic load would be there, so to set the controller according to different timings. For example: on a road in morning timing, load is more as compared to noon timing.
2) A dynamic model, in which traffic conditions is measured every time, when a signal changes. [8]

Fuzzy based Automatic Breaking of Trains: The intelligence for braking is provided by a fuzzy logic controller. The fuzzy logic controller is simulated using Matlab fuzzy logic Toolbox. Generally the Indian railways use two persons to operate a train and they employ a manual procedure for stopping the train in each station. The proposed fuzzy logic controller helps in reduction of manpower for the train operation. The simulated fuzzy logic controller gets activated at about 500m from the station based on its decision with reference to the speed and distance, the train stops at the station smoothly and automatically. The first step in the design is to select the number of stations where the train stops. The distances between the stations are calculated and stored. The fuzzy logic controller is fed with the instantaneous values of speed and distance. The controller constantly compares the distance between the previous and next station to distance traveled by train towards the approaching station, as and when the train is about 500m from the station the braking system in train gets activated. The use of fuzzy logic controller thus gives a smooth braking system. Developing and testing a prototype model of fuzzy logic controller can further prove the accuracy of the system for braking in subway trains.[9]

The fuzzy logic traffic lights controller performed better than the fixed-time controller or even vehicle actuated controllers due to its flexibility. The flexibility here involves the number of vehicles sensed (distance covered by vehicles) at the incoming junction and the extension of the green time. In the fixed-time controller, being an open-loop system, the green time is not extended whatever the density of cars at the junction. For vehicle actuated traffic light controllers, which is an enhanced version of fixed-time controller, the green time is extended whenever there is a presence of a vehicle. However, these times are fixed in advance up to a maximum time limit. For example when a car is detected, the green time is extended for another 5 or 10 seconds until the maximum time limit is reached. In the fuzzy logic controller, the extension time is not a fixed value. They are all fuzzy variables such as long, medium and small. The number of cars sensed at the input of the fuzzy controllers are also converted into fuzzy values, such as very small, small, medium, too many, etc. A simulation experiment was carried out to compare the performance of the fuzzy logic controller with a fixed-time conventional controller. The flow density of the simulation is varied according to real life traffic conditions. It can be observed from the results that the fuzzy logic control system provides better performance in terms of total waiting time as well as total moving time. Less waiting time will not only reduce the fuel consumption but also reduce air and noise pollution.[10] Fuzzy Controller can be further analyzed by using more number of input variables.

A new control method multi-phase Control is presented. Vehicles are generated based on detection. The additional detectors give possibilities to match vehicle movement with detector pulses. The exact traffic situation data and the good traffic model also provide an opportunity to predict the changes in the traffic flow, for example during the next 10 seconds. The fuzzy timing and selection algorithms make decisions based on this data and send the decisions to the controller unit and to the signal groups. the fuzzy rule base works at three levels:

1) Traffic situation level; the traffic situation is divided into three different levels (low demand, normal, oversaturated).
2) Phase and sequence level; the main goal of this level is to maximize the capacity by minimizing inter green times.
3) Green ending level or extension level; the main goal of this level is to determine the first moment to terminate a signal group.

Result is showing at least equal or better performance than the traditional vehicle actuated control.[11]

Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA will package functionality as a suite of interoperable *services* that can be used within multiple, separate systems from several business domains. Service Oriented Architecture is an architectural paradigm and discipline that may be used to build infrastructures enabling those with needs (consumers) and those with capabilities (providers) to interact via services across disparate domains of technology and ownership. Services act as the core facilitator of electronic data interchanges yet require additional mechanisms in order to function. Several new trends in the computer industry rely upon SOA as the enabling foundation. These include the automation of Business Process Management (BPM), composite applications (applications that aggregate multiple services to function), and the multitude of new architecture and design patterns generally referred to as Web 2.0.[12]

An artificial neural network (ANN), usually called neural network (NN), is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data. Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function f: X->Y or a distribution over X or both Xand Y, but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ANN model really means the definition of a *class* of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity).[13]

Neural Networks are very powerful tools that have been used in many domains. They can be applied to any problem of prediction, classification or control where there exists sufficient amount of observation data. Neural Networks owe this popularity to their powerful capacity to model extremely complex non linear functions and to their relatively easy use that is based on training-prediction cycles. In the training cycle the user presents to the network a *training pattern* that contains a set of inputs and a set of desired outputs that corresponds to the inputs. Next, in prediction cycle, the network is supposed to be able to supply the user with output values corresponding to input values that it has never seen thanks to its generalization capability. A good generalization is generally a complex task where the training set must contain sufficient information representing all cases so that a valid general mapping between outputs and inputs can be found. Furthermore, the training sets must be sufficiently large.[14]

A Neuro-fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples. Modern neuro-fuzzy systems are usually represented as special multilayer feedforward neural networks. In those neuro--fuzzy networks, connection weights and propagation and activation functions differ from common neural networks.[15]

These are the major surveys which helped me to implement the proposed system defined in next section.

4.1.2. Problem Definition

Transportation research has the goal to optimize transportation flow of people and goods. As the number of road users are increasing and resources provided by infrastructure is are limited, Intelligent Control of traffic is an important issue i.e. optimal control of lights using sophisticated sensors and intelligent optimization algorithms might be very beneficial. The main goal of traffic signal control is to ensure safety at signalized intersections by keeping conflict traffic flows apart. In fixed time controller, Size of signal time to cross the road was fixed means signal controller will give same time for green or red signal irrespective of the nature of traffic on the road. As road is occupied by different number of vehicles at different timing; signal time should not be same at different timings, so vehicles in urban areas experiences long travel time due to inefficient traffic light control. This problem is alleviated by using Real time controller; where signal controller gives optimize time for green or red signal based on the nature of traffic. One of the method for implementing the real time controller is Fuzzy logic; where controller gives output (Signal time required to cross the Traffic) as per current traffic conditions.

Fuzzy controller or Adaptive Signal Controller was implemented with the distance covered by the vehicles over road [10]; here one queue is considered for vehicle, length of that queue is used to calculate the time for controller. This controller is better than Preset Time Controller as signal time is calculated dynamically; but there are other factors also like Number or Type of vehicles, Total free space between vehicles; which affect the time required to cross the junction. Fuzzy controller is mostly dependant on the rules formed based on input variable. These rules are the mapping between input and output variables. Great effort or care is required to make these rules. In fuzzy controller, first detectors or sensors detects the traffic conditions and data is collected dynamically and controller will select the rules based on the collected data and finally gives output as traffic signal time required to cross the junction. Here, In Proposed work; Fuzzy controller is implemented by considering other variables also. Two approaches are used to design fuzzy controller. The approaches are as follows:

1) First approach considers two input variables; along with distance covered by vehicles

over the road, one more parameter is considered here; number of input vehicles over the road. as different vehicle take different time to cross the road. So, average number or type of vehicle is an important factor affecting the fuzzy controller. Numbers of vehicles are calculated using the average weight of vehicles on the road. Different type of traffic will take different time to cross the signal. Truck and buses will take more time to cross the road as compared to two wheelers and same distance can be covered by less number of four wheelers but more number of two wheelers; so number of vehicles cannot be ignored on the road.

2) Second approach considers the distance covered by vehicle over the road and the total free space between vehicles over the road occupied. Total space between the vehicles can be calculated by Image processing of the image of road covered by the vehicle. While calculating the free space between vehicles; a minimum value considered below which free is not taken into account because two wheeler stops at signal in haphazard or messy manner, so very small distance below a threshold is not taken.

In proposed work, it is also considered to partition the length of queue of vehicle in two parts if the Distance covered by vehicles on road is more. One part is allowed to pass and second part is allowed to cross in next signal timing. As Nature of Road is divided into three parts according to the type of vehicles over road:

1) Road mostly covered by Heavy Traffic vehicles.
2) Road mostly covered by Light Traffic vehicles.
3) Road mostly covered by Mix Traffic of vehicles.

As In case of heavy vehicles, space between vehicles is more and in case of light traffic vehicles (car, jeep, other four wheelers etc.) total free space between vehicles is less than heavy traffic vehicles, but in case of mix traffic vehicles total free space between vehicle is more less as compared to other two type of traffics. As in mix type of traffic, two wheelers will be there and two wheeler stands in messy manner that's why the total free space between vehicles is less.
If we consider the case of two metropolitan city: Delhi and Hyderabad; In Delhi most of the road will be covered by light four wheelers (cars); as a result total free space between vehicles will be more, where as in Hyderabad; mostly road will be covered by two wheelers such as Scooters, bikes etc.; as a result Total free space between vehicles will be less. So, Fuzzy controller will give different time for signals in both the metropolitan cities as nature of traffic

in both the cities is entirely different.

In proposed work, Input and output variables are divided into various categories like distance covered by vehicle be very small or small etc. Both approaches for fuzzy controller is designed, and Simulated in Fuzzy Toolbox of MATLAB. Results of both the approaches are compared, and analyzed.

4.2 Fuzzy Based Traffic Signal Controller

While Using the conventional traffic signal controller (fixed time controller), where time for each signal was fixed, whether there is any traffic or not. Many drivers ask why they have to wait so long for a signal to change. In addition to an increase in accident frequency, unjustified traffic signals can also cause excessive delay, disobedience of signals, and diversion of traffic to residential streets. Fuzzy logic based controllers are designed to capture the key factors for controlling a process without requiring many detailed mathematical formulas. Due to this fact, they have many advantages in real time applications.

The controllers have a simple computational structure, since they do not require many numerical calculations. The IF THEN logic of their inference rules does not require much computational time; Also controllers can operate on a large range of inputs, since different sets of control rules can be applied to them. If the system related knowledge is represented by simple fuzzy IFTHEN-rules, a fuzzy-based controller can control the system with efficiency and ease.

The main goal of traffic signal control is to ensure safety at signalized intersections by keeping conflict traffic flows apart. The entire knowledge of the system designer about the process, traffic signal control in this case, to be controlled is stored as rules in the knowledge base. Thus the rules have a basic influence on the closed-loop behavior of the system and should therefore be acquired thoroughly. The development of rules is time consuming, and designers often have to translate process knowledge into appropriate rules. Fuzzy logic technology allows the implementation of real-life rules similar to the way humans would think. For example, humans would think in the following way to control traffic situation at a certain junction: “if the traffic is heavier on the south and north lanes and the traffic on the east and west lanes is less, then the traffic lights should stay green longer for the north and south lanes”. Such rules can now be easily accommodated in the fuzzy logic controller. But enough knowledge is required to prepare knowledge database or rules for a controller.

4.2.1 Structure of fuzzy Controller

Fuzzy logic controller is responsible for controlling the length of green or red lights. Fuzzy based traffic controller consists of two parts.

(1) A dynamic model which tells the traffic conditions on the road i.e. the number of vehicles on the road etc.

(2) The output of this model is fed to the fuzzy controller (fuzzy model) to calculate the length of green signal.

Diagram below shows gathering and flow of information in the fuzzy traffic controller. Here first detector detects the traffic conditions and data is collected dynamically. And then fuzzification is applied to convert the real terms into fuzzy terms. In Fuzzy Inference system rules are there. Then rules are applied on fuzzy terms. hen we get the output in fuzzy terms. Then again fuzzy terms are converted into real terms or mathematical terms with appropriate units. After proper mapping, the output of controller is used as the length of green or red signals.

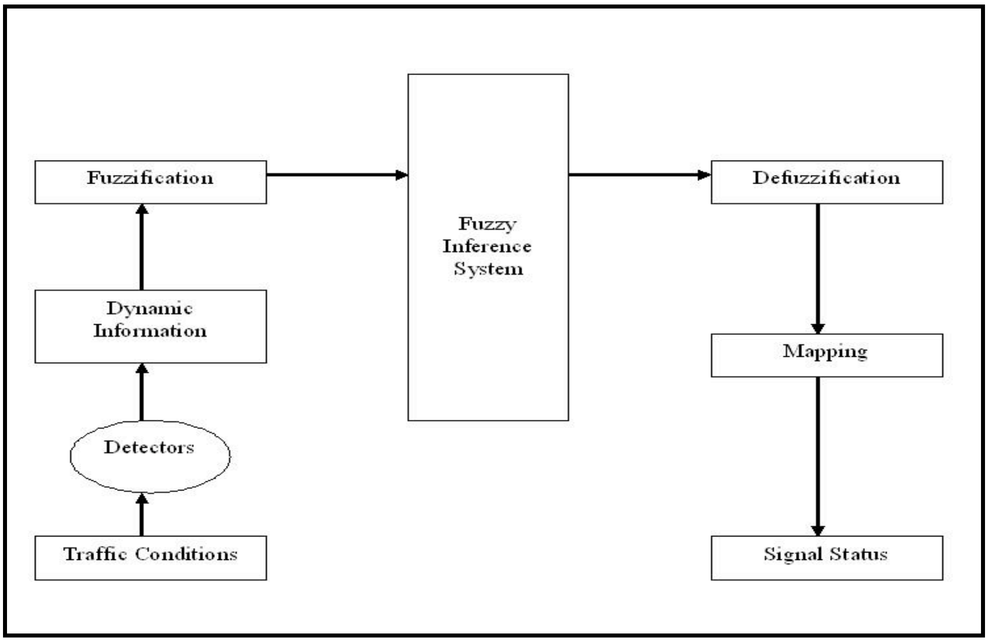


Figure 4.1. Structure of Fuzzy Traffic Controller

For making rules for fuzzy inference system, a lot of experience or knowledge is required.

Larger the number of rules better is the fuzzy controller, because the fuzzy data has to pass these numbers of rules. For both approaches we are considering roads with 2 lanes. For more lanes input and output parameters change accordingly.

4.2.2 Design of Fuzzy Controller (Approach 1)

In order to design a fuzzy controller, first road condition is to be identified. For this purpose, two variables are considered.

(1) Distance of Road Occupied.

(2) Number of Vehicles on the road.

Here, Number of vehicles is calculated using the weight of vehicles on covered road. We are considering average load of all the vehicles, based on that we are calculating the number of vehicles. The output of the controller is the length of green signals so as to optimize the traffic flow. Here it is considered that the vehicle can go either straight or right side. The Diagram for the controller is as below:

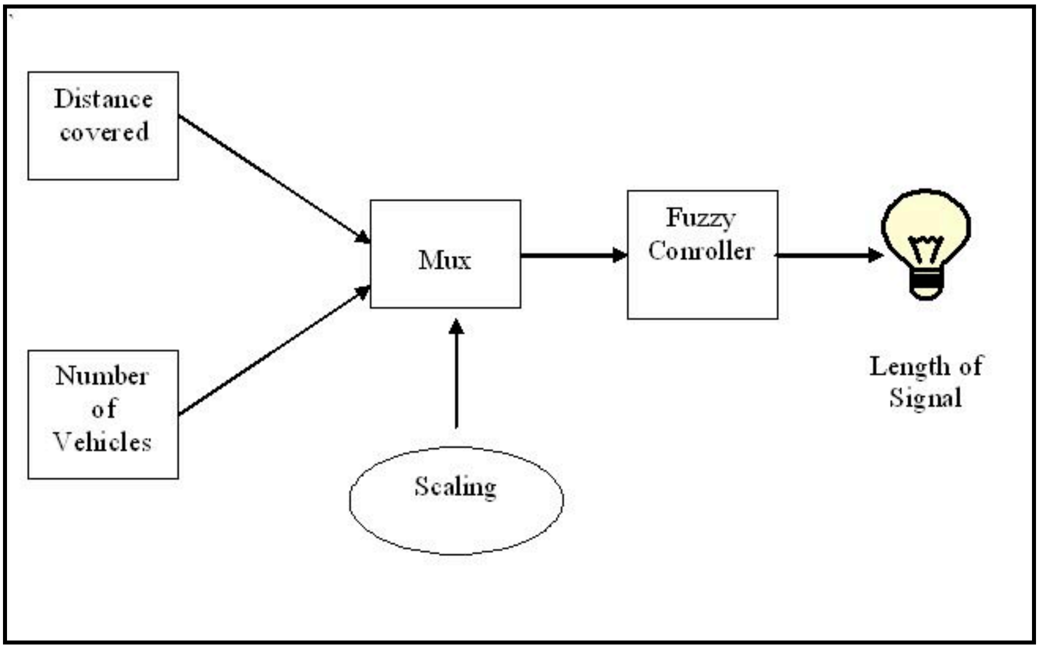


Figure 4.2. Fuzzy Controller

With two inputs (Number of vehicles and distance covered) and output (time of signals) are used for the controller. MATLAB Tool is used here for the simulation of Fuzzy Controller.

Here First Memberships functions for the Input and Output are defined. Then Rules are generated for the Inference system.

All variables can have four values. Distance variable can be vary small, small, far ,very far and number of vehicles can be very less, less, large and very large. Output variable can also have four values very less, less, medium and large.

The table can be formed from the values of variables. As both the input variable can have four values so we can have 4x4 = 16 rules. In sixteen rules output can have either of four values of time variable. As each value of input variable can combine with four value of other input variable.

Distance	**No. of vehicles**	**Time of Signal**
Very small	Very Less	Very less
Small	Less	Less
Far	Large	Medium
Very far	Very Large	large

So 16 rules formed from input variables by using if-then format (each combination contain AND connector)are:

(1) If (Distance is very small) and (No. of Vehicle is very less) then (Time is very less)

(2) If (Distance is very small) and (No. of Vehicle is less) then (Time is less)

(3) If (Distance is very small) and (No. of Vehicle is large) then (Time is less)

(4) If (Distance is very small) and (No. of vehicle is very large) then (Time is medium)

(5) If (Distance is small) and (No. of Vehicle is very less) then (Time is less)

(6) If (Distance is small) and (No. of Vehicle is less) then (Time is medium)

(7) If (Distance is small) and (No. of Vehicle is large) then (Time is medium)

(8) If (Distance is small) and (No. of Vehicle is very large) then (Time is medium)

(9) If (Distance is far) and (No. of Vehicle is very less) then (Time is less)

(10) If (Distance is far) and (No. of Vehicle is less) then (Time is medium)

(11) If (Distance is far) and (No. of Vehicle is large) then (Time is medium)

(12) If (Distance is far) and (No. of Vehicle is very large) then (Time is large)

(13) If (Distance is Very far) and (No. of Vehicle is very less) then (Time is less)

(14) If (Distance is Very far) and (No. of Vehicle is less) then (Time is medium)

(15) If (Distance is Very far) and (No. of Vehicle is large) then (Time is large)

(16) If (Distance is Very far) and (No. of Vehicle is very large) then (Time is large)

These are the rules formed. Whenever input is there, one of the sixteen rules is chosen by inference system and accordingly output is given. These rules are fed to fuzzy controller. Based on that time of green or red signal is calculated for each value of distance covered and number of vehicles on the road.

4.2.2.1 Implementation of fuzzy Controller

Using MATLAB (Fuzzy Tool Box), Membership Function for the Input and output is defined. For input variable Distance, values taken are as below. Here we have taken distance as meter. We have used minimum distance as 25 m, as seen by various road intersections, minimum distance covered by vehicles is 25 m. and minimum 20 vehicles can come at the distance of 25 m. and maximum time used to clear these traffic in very small distance is 25 sec. Total number of vehicles are in number (calculated by average weight of different types of vehicle). Time here considered in seconds.

Distance
0<Very small<=25
20<=Small<=45
35<=Far<=70
60<=Very far

For input variable No. of vehicle, values taken are as below:

No. of Vehicle

0<Very less<=20

15<=less<=35

30<=large<=45

40<=Very large

For output variable Time of signal, values taken are as below:

Time

0<Very Less<=25

20<=Less<=50

40<=Medium<=80

75<=Large

Membership Function Distance:

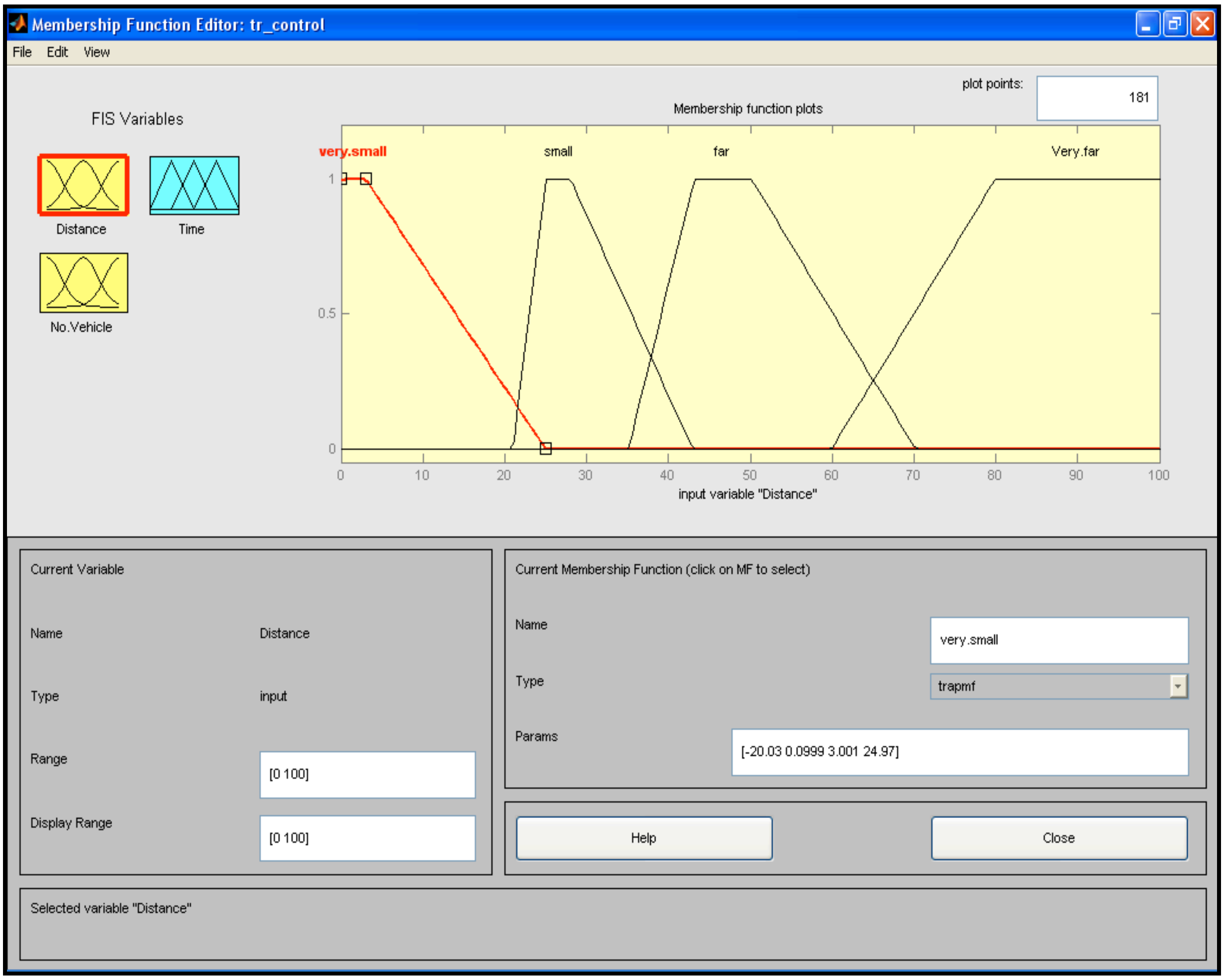


Figure 4.3. Membership function of Distance variable (approach1)

Membership Function No. of Vehicle:

Above defined value of no. of vehicles is fed to the MATLAB and we get the diagram as below: Total numbers of vehicles are calculated using the weight of all the vehicles on the road. Each type of vehicle is considered, Hence average weight of vehicles is considered.

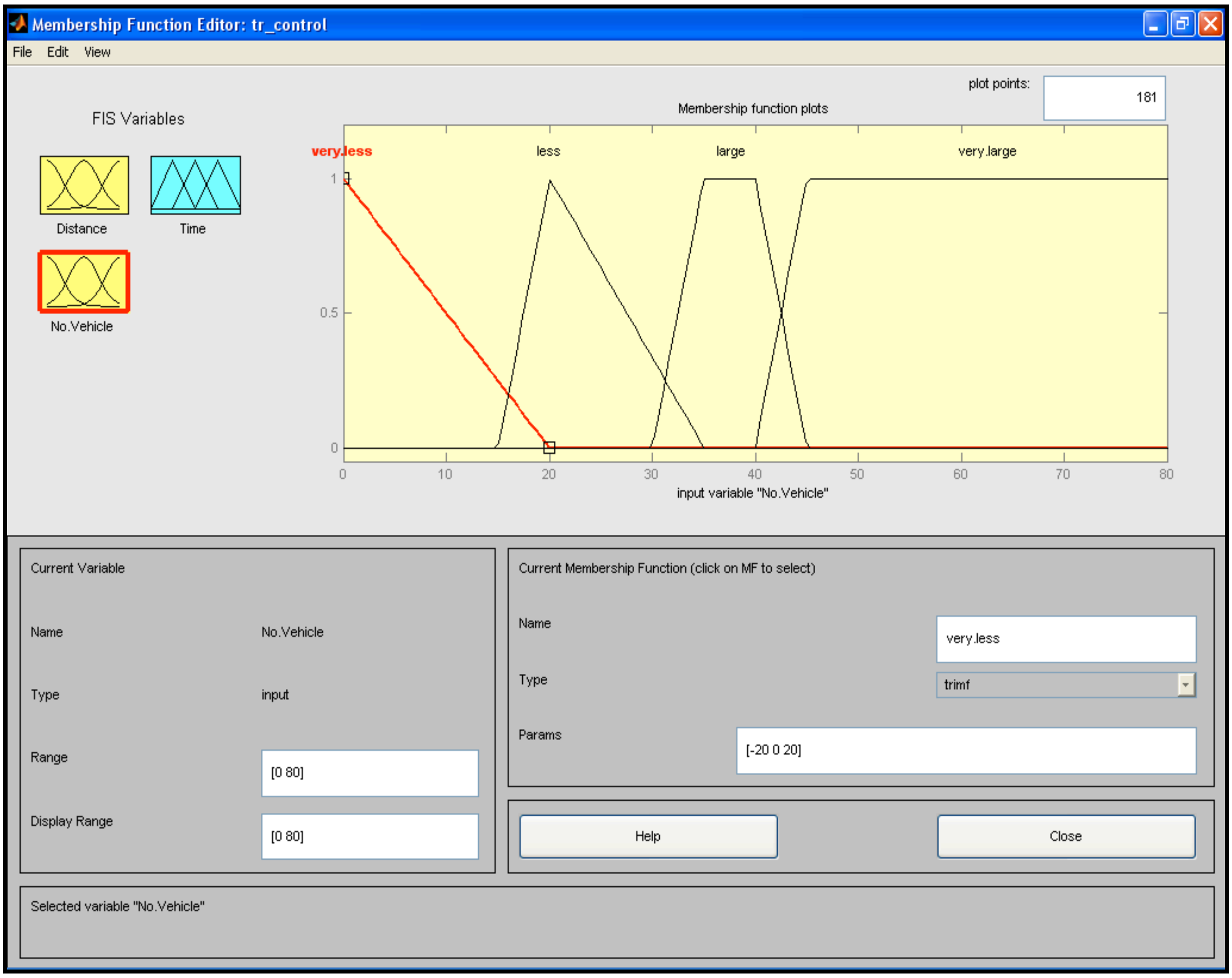


Figure 4.4. Membership function of No. of Vehicle variable (approach1)

In diagram we have membership value for each of the four status of the number of vehicle. for some status membership diagram is trigonometric and for some status membership diagram is trapezoidal. These shapes are as per the value of the no. of vehicle at proper timing. We can change the shape of the membership diagram by changing the value of the variables. But the membership value for any variable can be between 0 and 1. Zero represents complete non membership where one represents the complete membership.

Membership Function Time: Like input variables, there is a membership diagram for output variables also. Diagram below represents the membership Diagram of output variable. Depend on values; membership function changes its shape. Below we have two shape either triangular or trapezoidal.

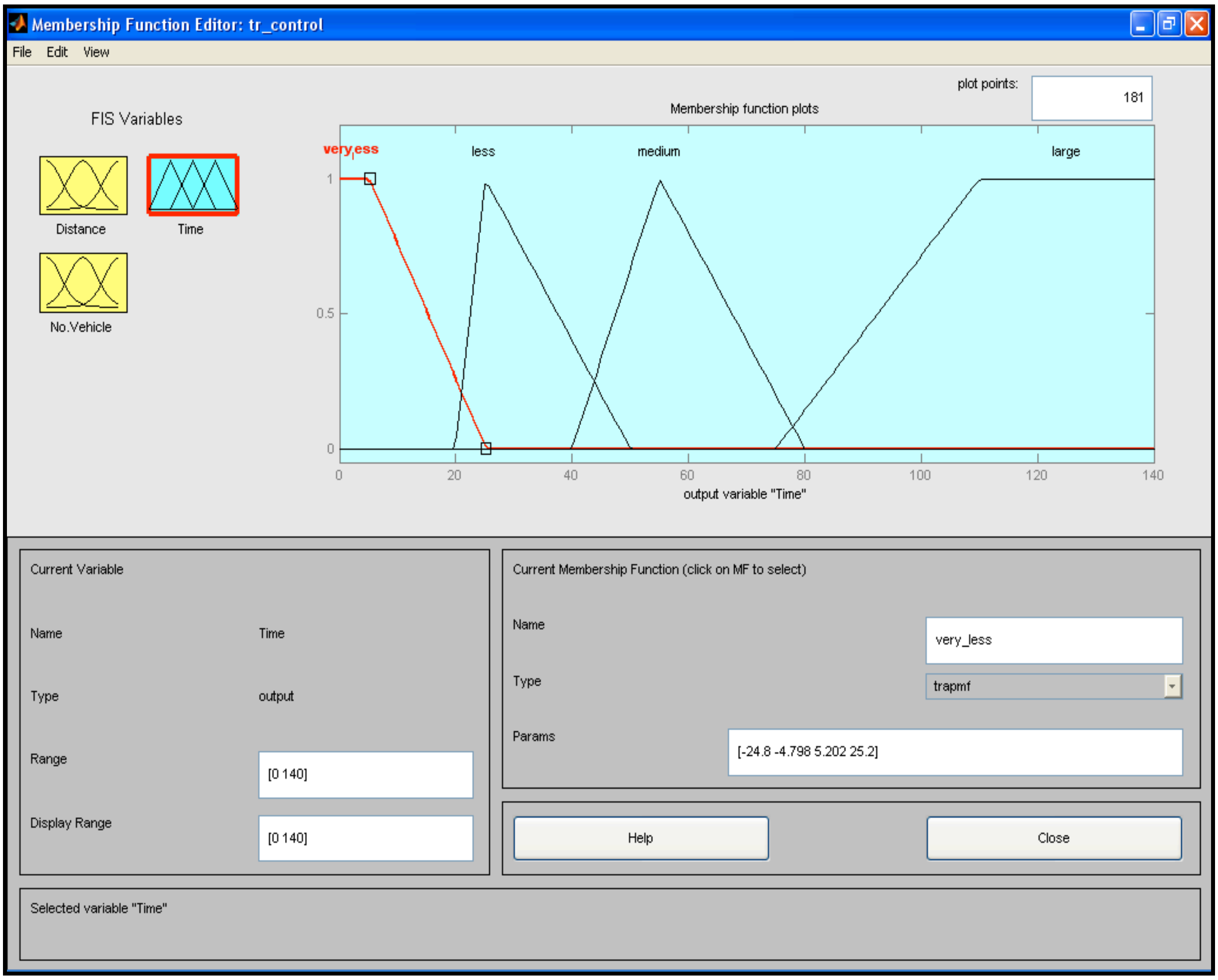


Figure 4.5. Membership function of Time variable (approach 1)

4.2.2.2 Output of Fuzzy Controller

Now three variables and each variable are having four values. By mapping each value of input And output variables; sixteen rules are there. By combining the all these four we have the two diagrams; one is surface diagram i.e. three dimensional mapping of all three variable and second one is rule editor showing all sixteen rules formed using input and output variables. Both diagrams are shown below:

Final Output of Fuzzy Controller: Diagram below is surface diagram of controller

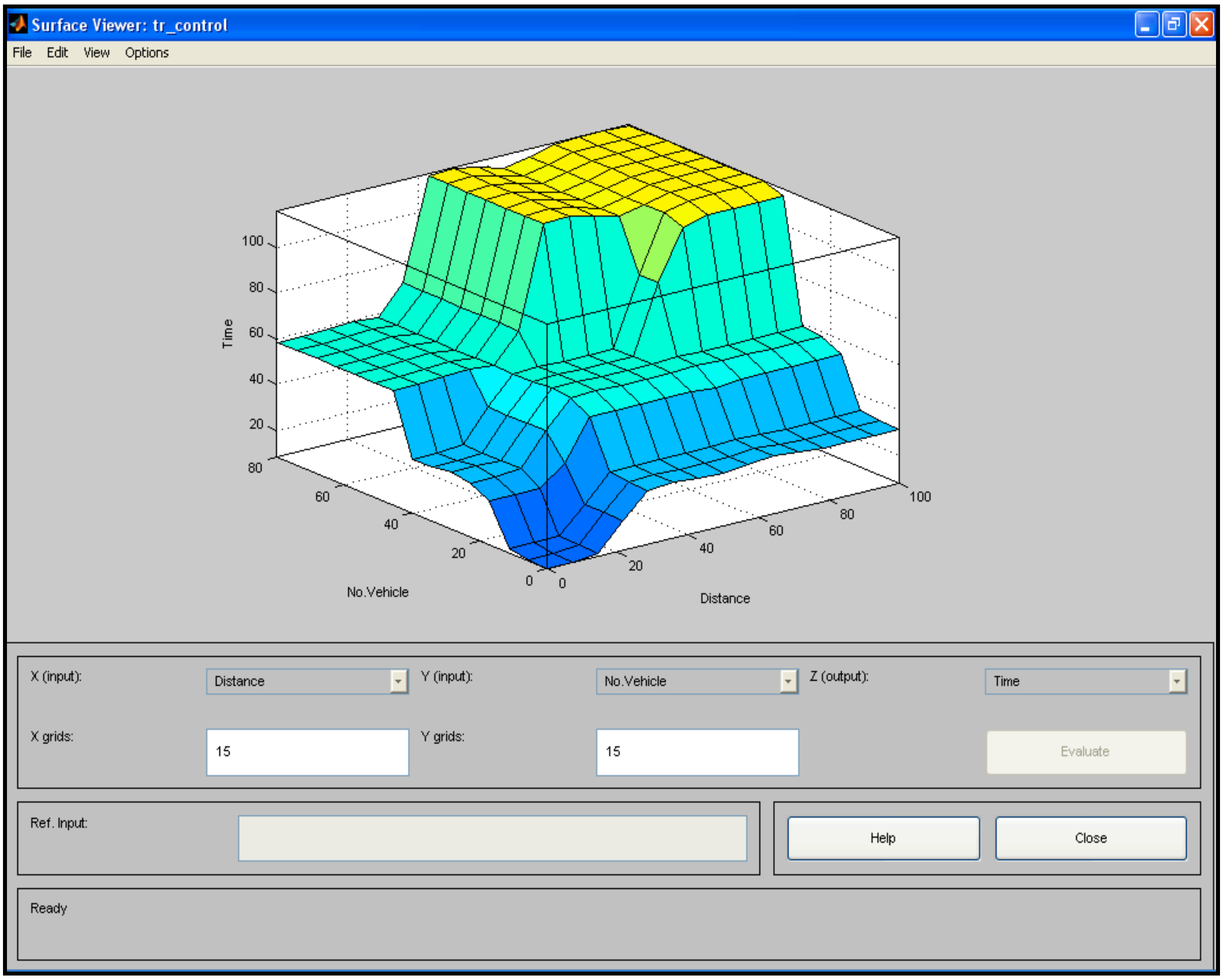


Figure 4.6. Surface view of Fuzzy Controller (approach 1)

As in diagram, we have three axes. Along x axis ,we have distance, along y axis, we have No. of Vehicle and along z axis, we have time of signal. For each value of distance and number of vehicle we can evaluate the corresponding time. Next two diagrams are related to rules. First represent the 16 rules and second represent the diagrammatic representation. We can increase or decrease the number of rules by increasing or decreasing the number of values of input variable.

Rule Editor: Diagram below shows the 16 rules required to designed fuzzy controller. Thee rules are mapping between input variables i.e. distance and Total number of vehicles and output variable i.e. time required for a signal. Number and value of input variables can change the number of rules. Suppose, If number of input variables is two and each input variables is having 5 values; then number of rules will be 25.

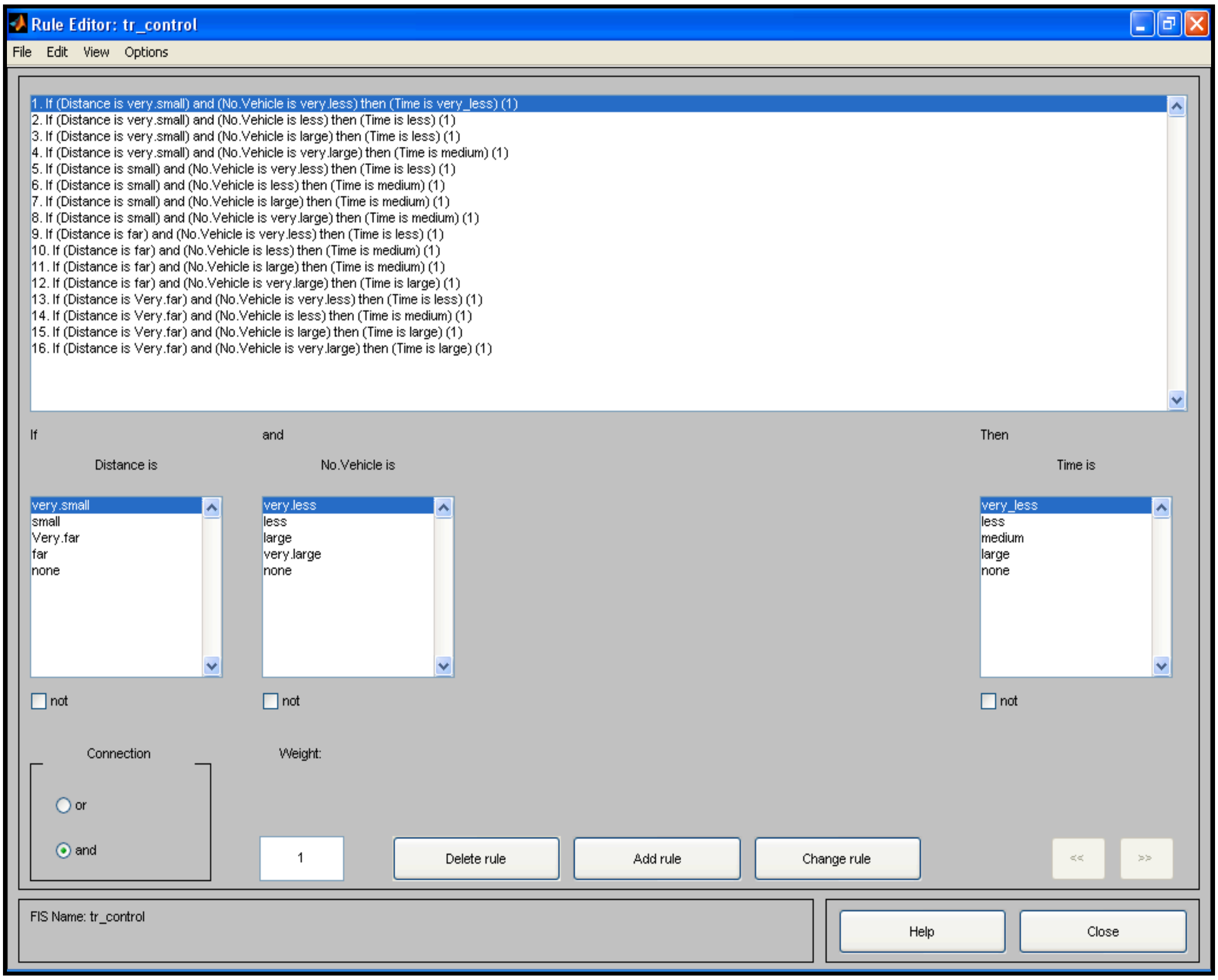


Figure 4.7. Rule Editor for Fuzzy Controller (approach 1)

As in diagram, we have three lists one is of distance, second is of no. of vehicles and third is of time. We have three options:

(1) Add the rule

(2) Delete the rule

(3) Change the rule

Two connectors are there

(1) OR Operator

(2) AND Operator

One more option NOT is given to reverse or negate the value of the variable. We can use various combinations and create rule, add rule and change rule according to requirement of desired Controller.

Diagrammatic Representation of Rules:

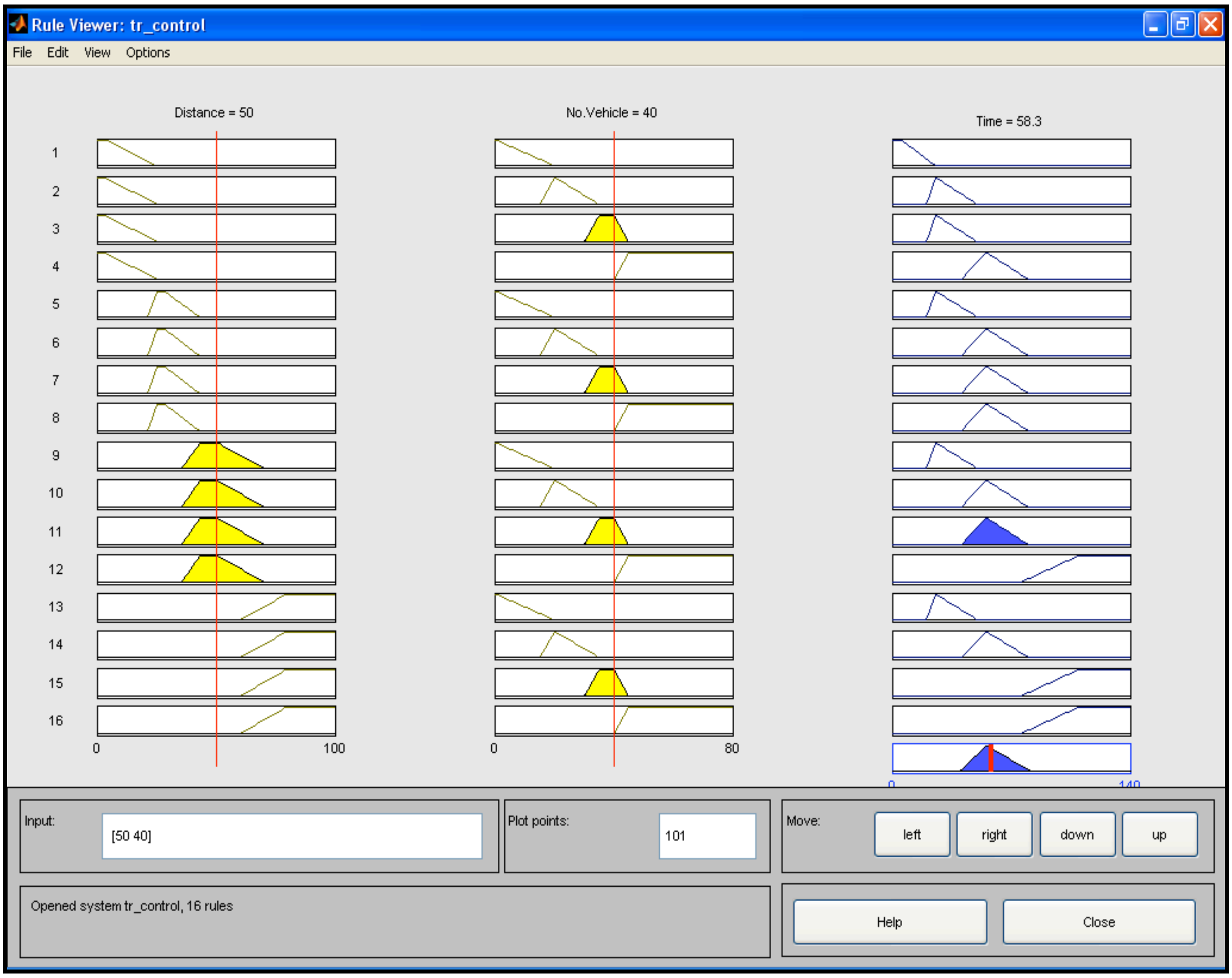


Figure 4.8. Diagrammatic Representation of Rules (approach 1)

Overall we say, Fuzzy Controller does:

(1) The vehicle stop times in the controlled intersections was reduced.

(2) The vehicle delays were reduced.

(3) The average vehicle velocity was improved.

(4) The times that the traffic accidents happened was cut.

By using Fuzzy controller, there would be minimum conflict at the junction. The fuzzy signal control can be multiobjective and more efficient than conventional adaptive signal control nowadays;

4.2.3 Design of Fuzzy Controller (Approach 2)

In Second approach, to design a fuzzy controller, two variables are considered, these variables are as below:

(1) Distance covered over road.

(2) Free Space between vehicles on the road or Space b/w vehicles.

Free space between vehicles can be calculated using the image processing of the image of the road covered. In An image of road, color of unoccupied road and occupied road will be different. The output of the controller is Time of signal required to control the traffic; which is considered as output variable. All input and output variables are having four values. Distance variable can be vary small, small, far ,very far and Space between vehicles (free space between vehicles on road) can be very less, less, medium and large. Output variable can also have four values very less, less, medium and large. The values of input and output variables are taken as:

Distance	**Space b/w vehicles**	**Time of Signal**
Very small	Very Less	Very less
Small	Less	Less
Far	Large	Medium
Very far	Very Large	large

As each input variable are having four values. So 16 rules formed from input variables by using if-then format (each combination contain AND connector) are:

(1) If (Distance is very small) and (Space b/w Vehicle is very less) then (Time is less)

(2) If (Distance is very small) and (Space b/w Vehicle is less) then (Time is less)

(3) If (Distance is very small) and (Space b/w Vehicle is medium) then (Time is very less)

(4) If (Distance is very small) and (Space b/w Vehicle is large) then (Time is very less)

(5) If (Distance is small) and (Space b/w Vehicle is very less) then (Time is medium)

(6) If (Distance is small) and (Space b/w Vehicle is less) then (Time is less)

(7) If (Distance is small) and (Space b/w Vehicle is large) then (Time is less)

(8) If (Distance is small) and (Space b/w Vehicle is very large) then (Time is medium)

(9) If (Distance is far) and (Space b/w Vehicle is very less) then (Time is large)

(10) If (Distance is far) and (Space b/w Vehicle is less) then (Time is large)

(11) If (Distance is far) and (Space b/w Vehicle is large) then (Time is medium)

(12) If (Distance is far) and (Space b/w Vehicle is very large) then (Time is large)

(13) If (Distance is Very far) and (Space b/w Vehicle is very less) then (Time is large)

(14) If (Distance is Very far) and (Space b/w Vehicle is less) then (Time is large)

(15) If (Distance is Very far) and (Space b/w Vehicle is large) then (Time is medium)

(16) If (Distance is Very far) and (Space b/w Vehicle is very large) then (Time is large)

When ever traffic conditions are sensed by sensor, Fuzzy controller chooses one of the rules and controller calculates the time for the signal. Basically rules defined the mapping between input and output variables.

4.2.3.1 Implementation of fuzzy Controller

Implementation of fuzzy controller (as per design) is done in Fuzzy Tool box using MATLAB (Fuzzy Tool Box), the value of input variable Distance and space between vehicles is defined as below. As in Approach 1; here also distance is taken in meters and space between vehicles is taken in square feet (as free space is the unoccupied area between two vehicles that is dependent on the type of traffic on the road) and minimum free space considered is 3 square feet. Time is taken in seconds.

Distance **Space b/w vehicle**

0<Very small<=25	0<Very less<=30
20<=Small<=45	25<=less<=55
35<=Far<=70	50<=Medium<=75
60<=Very far	70<=Very large

For output variable Time of signal, values taken are as below:

Time

0<Very Less<=25

20<=Less<=50

40<=Medium<=80

75<=Large

Membership Function Distance:

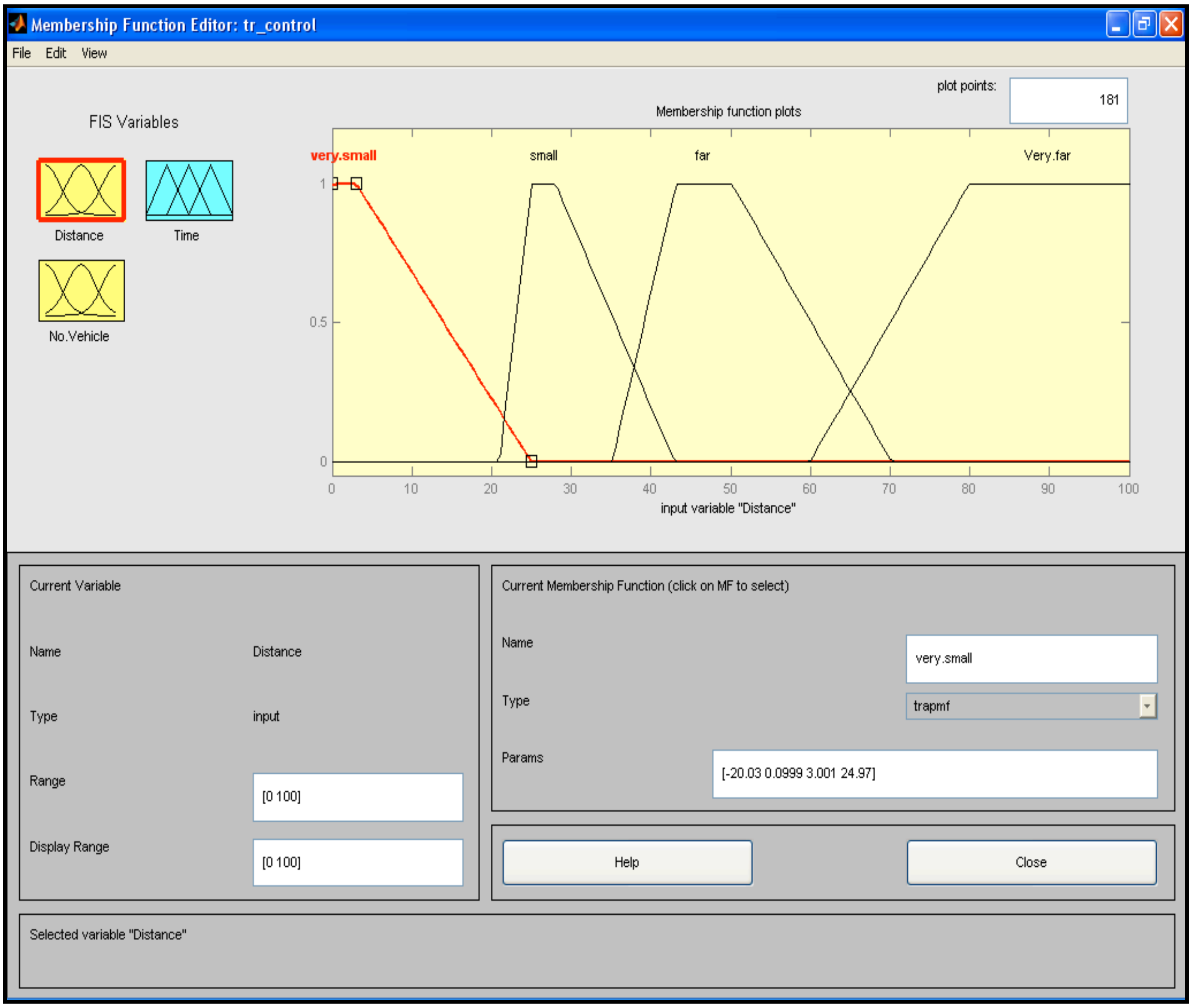


Figure 4.9. Membership function of Distance variable (approach 2)

As above figure shows the membership diagram for the input variable ‘Distance. Input variable is having four values so four membership functions have been defined. If we increase the number of values, number of membership function is increase.

Membership Function Space b/w Vehicle:

Above defined value of Space b/w Vehicle is fed to the fuzzy toolbox in MATLAB and we get the diagram as below and shape of membership diagram is either triangular or trapezoidal here, depend on the value of variable.

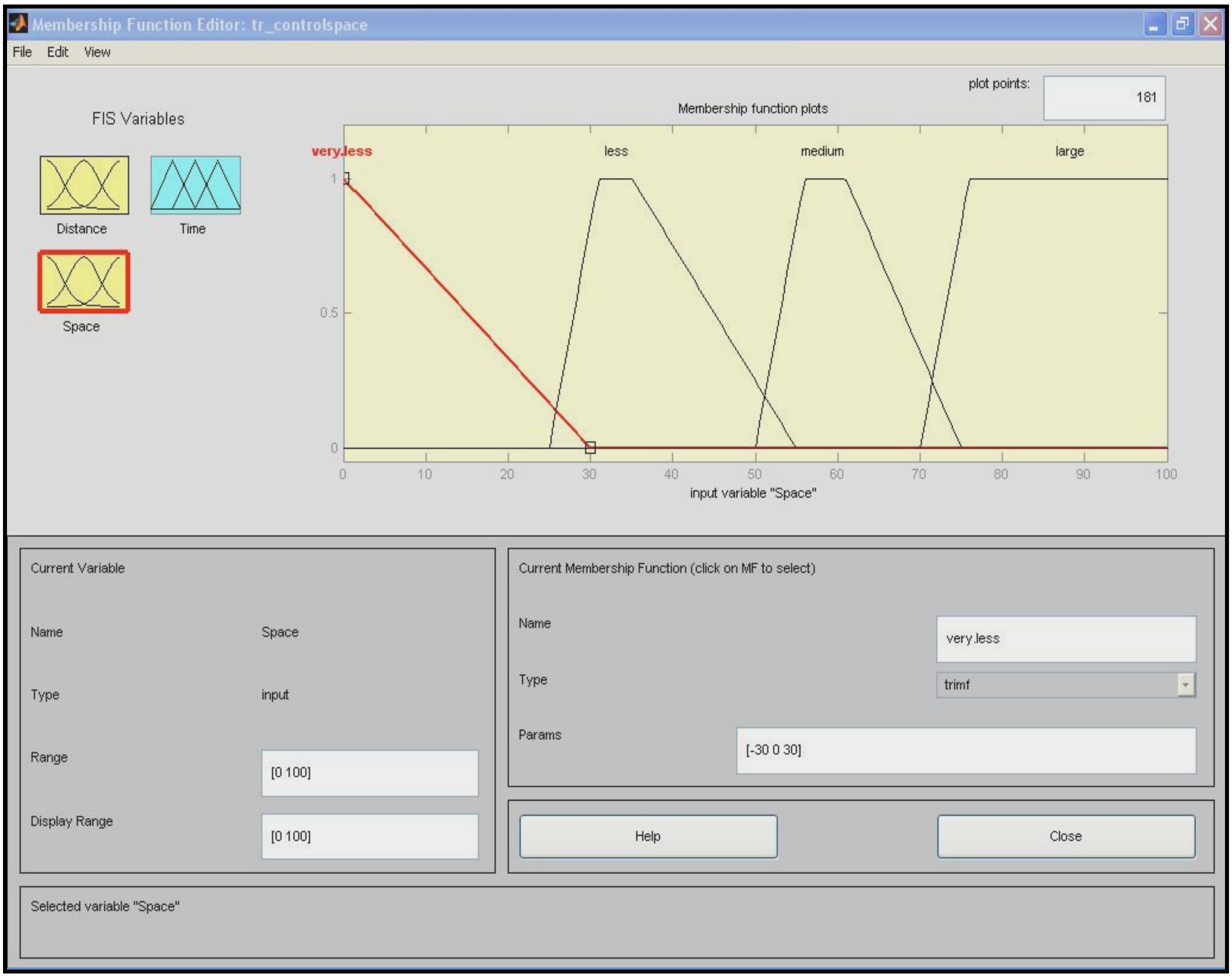


Figure 4.10. Membership function of Space variable (approach 2)

Membership Function Time: Diagram below Represent the membership function for output variable(Time of signal Required).

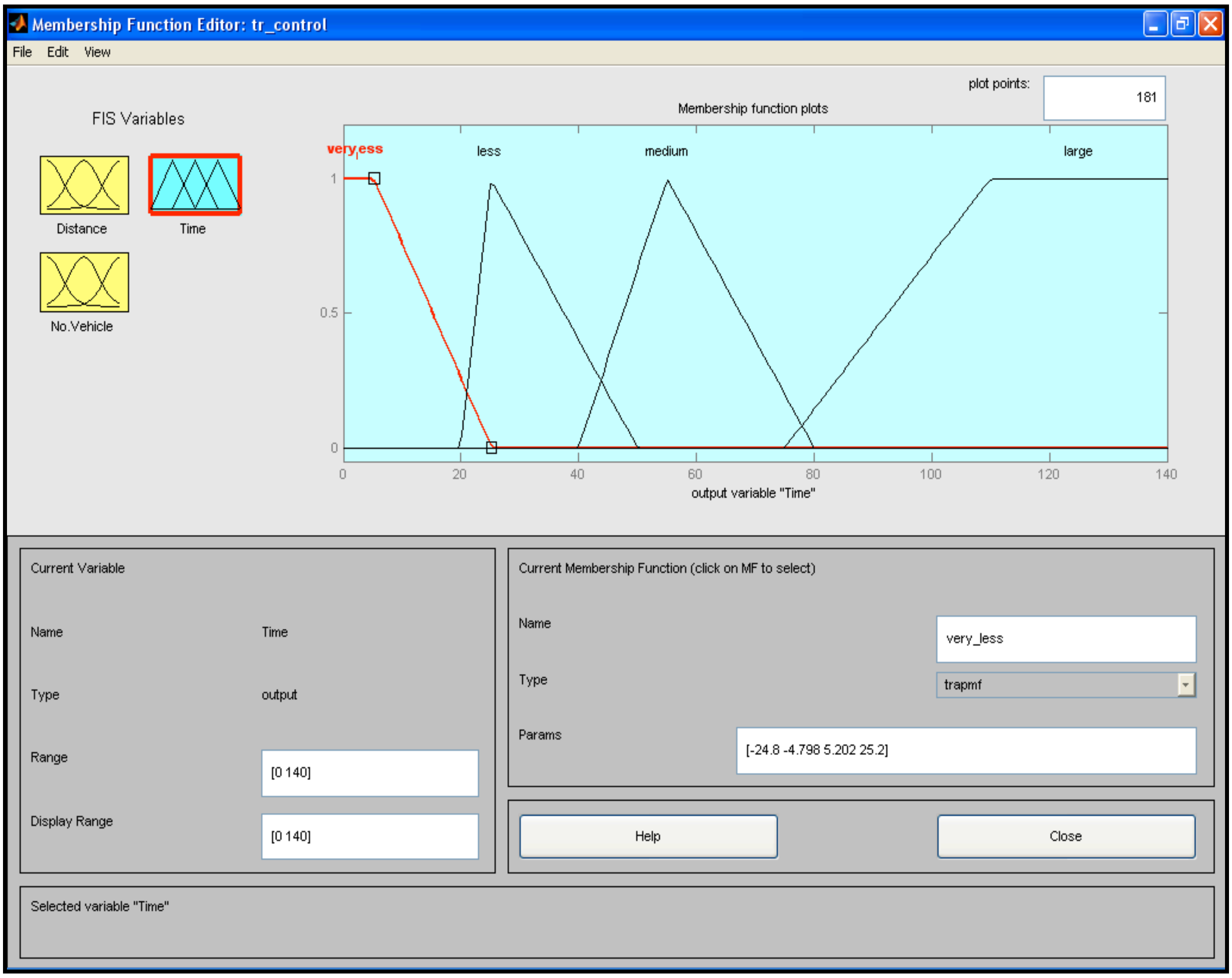


Figure 4.11. Membership function of Time variable (approach 2)

There are the membership functions of the two inputs and one output variable. After that sixteen rules were formed, they were written in rule window.

4.2.3.2 Output of Fuzzy Controller

Now three variables and sixteen rules are there. By combining the all these four we have the Surface formed (output is produced) as below.

Final Output of Fuzzy Controller: Diagram below represents the surface diagram of the fuzzy controller.

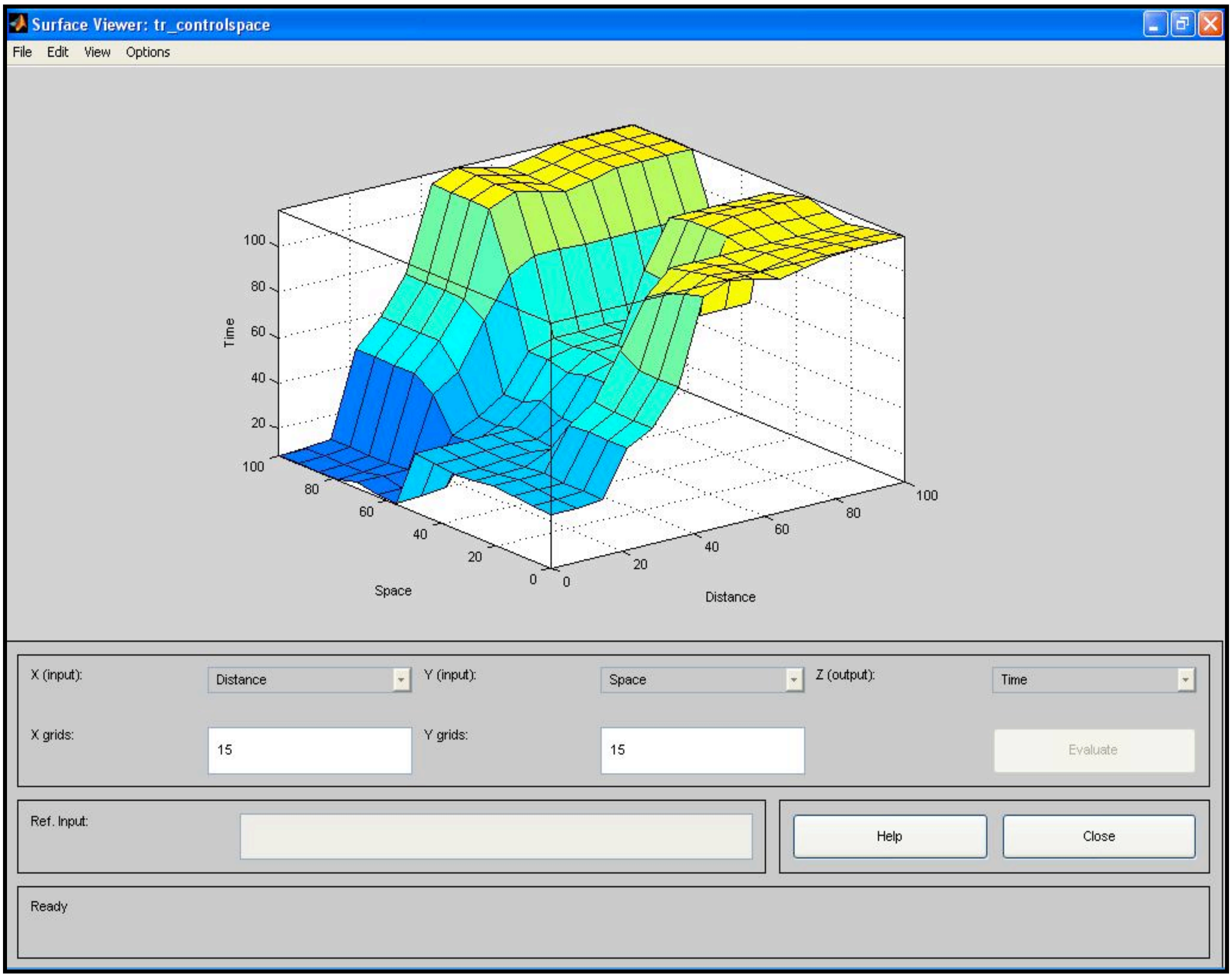


Figure 4.12. Surface view of Fuzzy Controller (approach 2)

In Above diagram three axes. Along x axis, we have distance, along y axis, we have Free Space b/w Vehicle and along z axis, we have time of signal.

Diagrammatic Representation of Rules:

Following diagram shows the mapping between the input and output variable. What can be the possible values of time of signal for the various values of input variables.

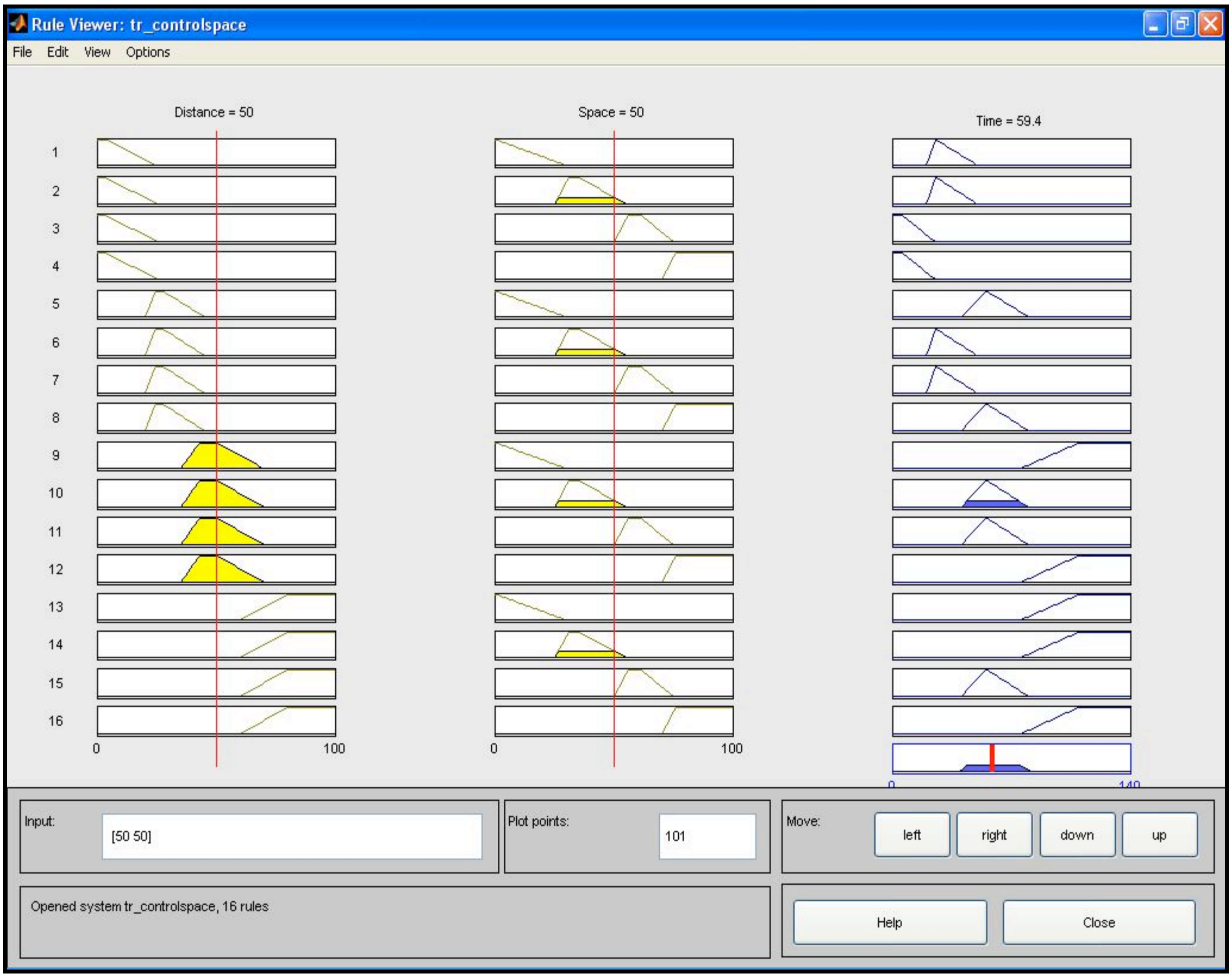


Figure 4.13. Diagrammatic Representation of Rules (approach 2)

Here we are not considering the Image processing for the controller with approach 2. Image processing is not a part of this project. In approach 2, there is significant change from approach 1.

4.2.4 Analysis and Comparison between Two Approaches

Now, there are two approaches to design fuzzy controller. In first approach, we are considering two input variable, one is distance covered by vehicle over road and second one is Number of vehicles over road. The output variable is time of signal. In second approach; input variables taken are Distance covered and Total Free space between vehicles. The output of controller is Time of signal. Here we are considering three cases of traffic load over roads. Numbers of vehicles are calculated using the total weight of vehicles on road covered; where as total free space between vehicles can be calculated by image processing of the image of the road covered by vehicles.

1) When road is mostly covered by Heavy Traffic Vehicles.
2) When road is mostly covered by Light Traffic Vehicles.
3) When road is covered by Mix Traffic Vehicles.

Here each of above case is considered one by one.

1) Analysis: When load is mostly covered by Heavy Traffic: In fuzzy Controller, variable Distance is having four values. We are considering the case in each value as follows:

a) When distance covered is very small,
Number of vehicles on road is very less,
Free Space b/w vehicles is medium (Total free space)
Time of signal (Approach1) = very less
Time of Signal (Approach2) = very less

b) When distance covered is small,
Number of vehicle on road is very less,
Free Space b/w vehicles is medium (Total free space)
Time of signal (Approach1) = less
Time of Signal (Approach2) = less

c) When distance covered is far,
Number of vehicle on road is less,
Free Space b/w vehicles is medium (Total Free space)
Time of signal (Approach1) = medium
Time of Signal (Approach2) = large

d) When distance covered is very far,
Number of vehicle on road is large,
Free Space b/w vehicles is large (Total Free Space)
Time of signal (Approach1) = large
Time of Signal (Approach2) = large

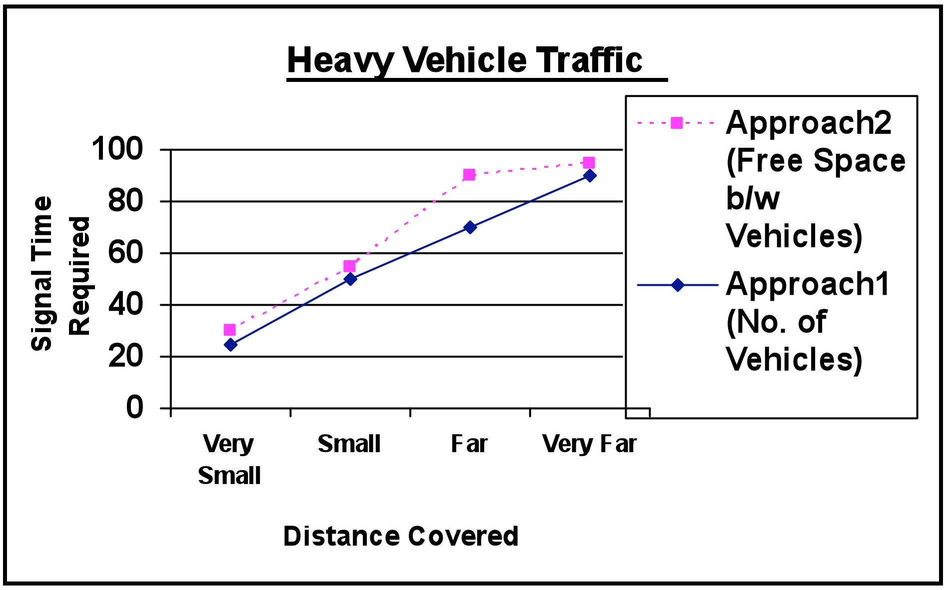


Figure 4.14. Heavy Traffic Vehicle Comparison

Here in Figure 4.14. Fuzzy controller with approach 1 gives slightly better result than approach 2. As in Heavy traffic case, When distance covered is very small then number of vehicles (calculated by load of vehicles on road) is very less and free space between vehicles is medium then time calculated by both the approach is same i.e. very less and When distance covered is small then number of vehicles (calculated by load of vehicles on road) is very less and free space between vehicles is medium then time calculated by both the approach is same i.e. less. When distance covered is far then number of vehicles is less and free space between vehicles is medium then time calculated by both the approach shows significant difference. Approach 1 is showing less time (better result) than Approach 2. When distance covered is very far then number of vehicles is very large and free space between vehicles is large then time calculated by both the approach is same i.e. large, so As per simulation results, Approach 1 for fuzzy Controller is slightly better than Approach 2 in case of heavy traffic.

2) Analysis: When load is mostly covered by Light Traffic: In fuzzy Controller, variable Distance is having four values. We are considering the case in each value as follows:

a) When distance covered is very small,
Number of vehicles on road is less,
Free Space b/w vehicles is very less (Total Free Space)

Time of signal (Approach1) = less

Time of Signal (Approach2) = less

b) When distance covered is small,

Number of vehicles on road is less,

Free Space b/w vehicles is less (Total Free Space)

Time of signal (Approach1) = medium

Time of Signal (Approach2) = medium

c) When distance covered is far,

Number of vehicles on road is very large,

Free Space b/w vehicles is less (Total Free Space)

Time of signal (Approach1) = large

Time of Signal (Approach2) = large

If we plot the graph between for both the approaches, it is as follows:

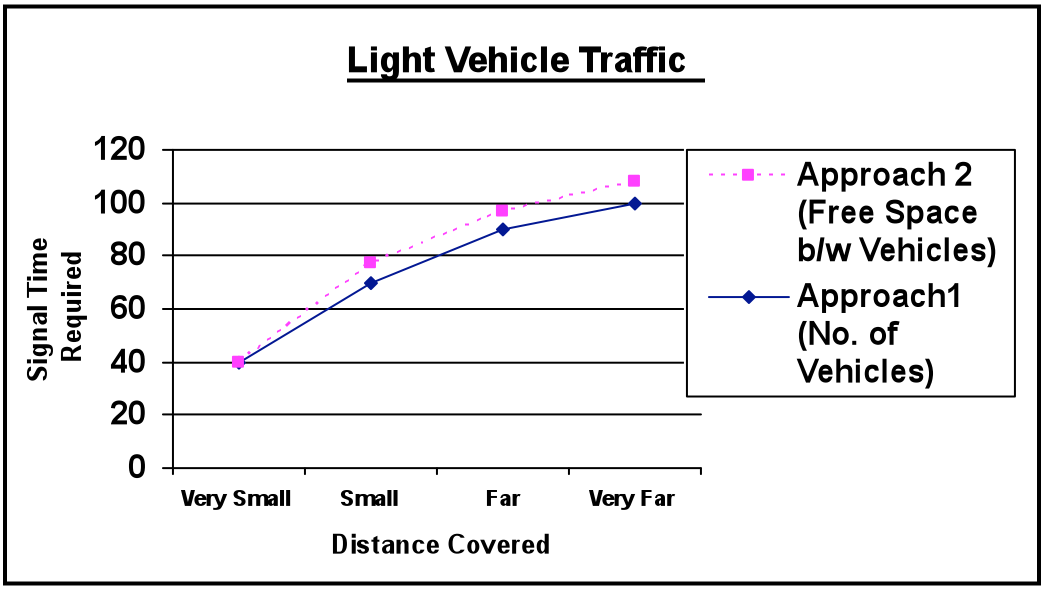


Figure 4.15. Light Traffic Vehicle Comparison

d) When distance covered is very far,

Number of vehicles on road is very large,

Free Space b/w vehicles is very large (Total Free Space)

Time of signal (Approach1) = large

Time of Signal (Approach2) = large

As per simulation done using MATLAB Tool, When Two Approaches for fuzzy Controller are compared in case of light Traffic vehicles, result comes as Figure 4.14. In case of Light traffic, When distance covered is very small then number of vehicles (calculated by load of vehicles on road) is less and free space between vehicles is very then time calculated by both the approach is same i.e. less and When distance covered is small then number of vehicles (calculated by load of vehicles on road) is less and free space between vehicles is less then time calculated by both the approach is same i.e. medium. When distance covered is far then number of vehicles is large and free space between vehicles is less then time calculated by both the approach is same i.e. large. When distance covered is very far then number of vehicles is very large and free space between vehicles is very large then time calculated by both the approach is same i.e. large, so in all four case , Approach 1 for fuzzy Controller is giving approximately same result as Approach 2.

3) When load is mostly covered by Mix Traffic: In fuzzy Controller, variable Distance is having four values. We are considering the case in each value as follows:

a) When distance covered is very small,
Number of vehicles on road is very less,
Free Space b/w vehicles is very less (Total Free Space)
Time of signal (Approach1) = very less
Time of Signal (Approach2) = less

b) When distance covered is small,
Number of vehicles on road is less,
Free Space b/w vehicles is large (Total Free Space)
Time of signal (Approach1) = medium
Time of Signal (Approach2) = less

c) When distance covered is far,
Number of vehicles on road is large,
Free Space b/w vehicles is less (Total Free Space)
Time of signal (Approach1) = medium
Time of Signal (Approach2) = large

If we plot the graph between for both the approaches, it is as follows:

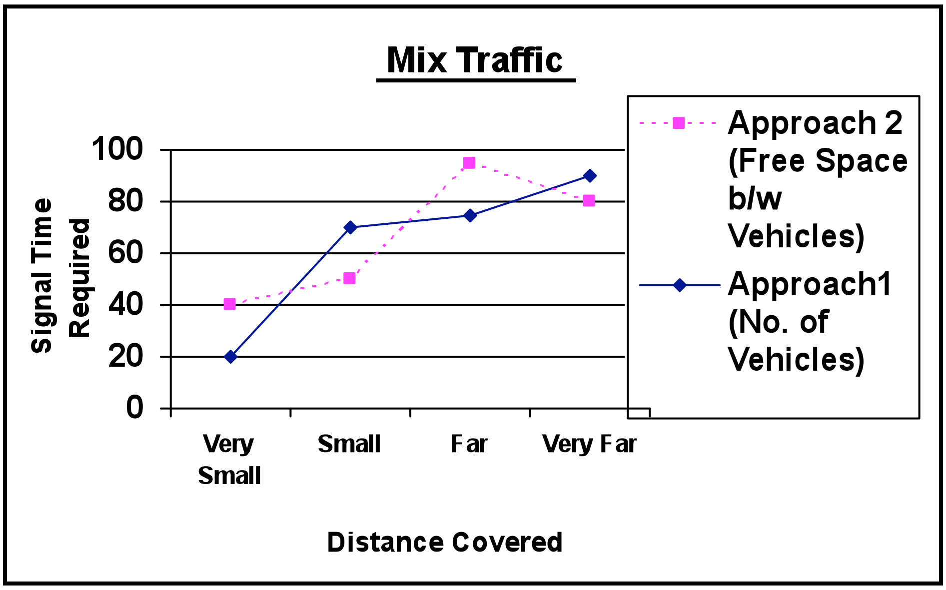


Figure 4.16. Mix Traffic Vehicle Comparison

d) When distance covered is very far,

Number of vehicles on road is large,

Free Space b/w vehicles is large (Total Free Space)

Time of signal (Approach1) = large

Time of Signal (Approach2) = large

Comparison between two approaches of Fuzzy controller is shown in figure 4.16 as per simulation results. As in Mix traffic case, When distance covered is very small then number of vehicles (calculated by load of vehicles on road) is very less and free space between vehicles is very less then time calculated by the Approach 1 is very less where as for approach 2 is less i.e. approach 2 is giving better result (It takes more time to clear traffic) and When distance covered is small then number of vehicles (calculated by load of vehicles on road) is less and free space between vehicles is large then time calculated by the Approach 1 is medium where as for approach 2 is less. When distance covered is far then number of vehicles is large and free space between vehicles is less, time calculated by the Approach 1 is medium where as for approach 2 is large, In three cases, there is significant difference between time calculated by both the approach. In fourth case, When distance covered is very far then number of vehicles is large and free space between vehicles is large then time calculated by both the approach is same i.e. large, in this case, Approach 1 is allowing all traffic vehicles to pass where as Approach 2 is dividing the traffic into two parts, approach 2 allows one part to go while other part is to wait. So As per simulation results, Approach 2 for fuzzy Controller is better than Approach 1 in case of Mix traffic.

4.3. SERVICE ORIENTED ACHITECTURE FOR MULTIPLE LINKED FUZZY CONTROLLER

The widespread emergence of the Internet in the mid 1990s as a platform for electronic data distribution and the advent of structured information have revolutionized our ability to deliver information to any corner of the world. While the introduction of Extensible Markup Language (XML) as a structured format was a major enabling factor, the promise offered by SOAP based web services triggered the discovery of architectural patterns that are now known as Service Oriented Architecture (SOA). [12]

The reality in IT enterprises is that infrastructure is heterogeneous across operating systems, applications, system software, and application infrastructure. Some existing applications are used to run current business processes, so starting from scratch to build new infrastructure isn't an option. Enterprises should quickly respond to business changes with agility; leverage existing investments in applications and application infrastructure to address newer business requirements; support new channels of interactions with customers, partners, and suppliers; and feature an architecture that supports organic business. SOA with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments.

Service Oriented Architecture is an architectural paradigm and discipline that may be used to build infrastructures enabling those with needs (consumers) and those with capabilities (providers) to interact via services across disparate domains of technology and ownership. Services act as the core facilitator of electronic data interchanges yet require additional mechanisms in order to function. Several new trends in the computer industry rely upon SOA as the enabling foundation. These include the automation of Business Process Management (BPM), composite applications (applications that aggregate multiple services to function), and the multitude of new architecture and design patterns generally referred to as Web 2.0.

Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains and implemented using various technology stacks. In general, entities (people and organizations) create capabilities to solve or support a solution for the problems they face in the course of their business. It is natural to think of one person’s needs being met by capabilities offered by someone else; or, in the world of distributed computing, one computer agent's requirements being met by a computer agent belonging to a different owner. The term owner here may be used to denote different divisions of one business or perhaps unrelated entities in different countries.[12]

There is not necessarily a one-to-one correlation between needs and capabilities; the granularity of needs and capabilities vary from fundamental to complex, and any given need may require a combination of numerous capabilities while any single capability may address more than one need. One perceived value of SOA is that it provides a powerful framework for matching needs and capabilities and for combining capabilities to address those needs by leveraging other capabilities. One capability may be repurposed across a multitude of needs. SOA is a "view" of architecture that focuses in on services as the action boundaries between the needs and capabilities in a manner conducive to service discovery and repurposing.

Figure 4.17. Service Oriented Architecture

[http://pssiusa.files.wordpress.com/2010/10/soa_pic.jpg**]**

Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services. One can envisage SOA as a sort of continuum, as opposed to distributed computing or modular programming.

SOA for Integrated Controllers

The integration of Traffic Signal Controllers (TSC) is the foundation of urban intelligent traffic control. With the development of social economy and the promotion of urbanization, urban vehicle possession is sharply increasing, and the traffic load is continuously growing, more and more road intersections need for the traffic signal control. Accordingly, traffic control software platform is under increasing pressure. The research of software platform, based on large-scale TSC system, is very

With the rapid development of Internet and Web technology, the traditional distributed computing technology, such as DCOM and CORBA, has been unable to properly apply to the Web environment in the platform independence, the interoperability, and the firewall penetrability to call WEB components.

Intelligent Transport Information Sharing (ITIS) refers to so many regions, organizations and domains, huge amounts of heterogeneous data stored in very large database or produced by real-time traffic detectors become both the gold mine and the obstacle for ITIS. SOA technology has been considered as an effective approach that promises to solve the contradiction. The multilayer SOA for ITIS, can be divided into four layers including Resource or Hardware Layer, Service Layer, ITIS Sub-system Layer and ITIS Application Layer.

1) Resource or hardware Layer: This layer provides all kinds of data resources for upper layers. In this layer, resource can be divided into two categories: physical resource and logical resource. In large-scale network TSC system, the former contains all the TSC, and the latter contains the data in the database and the relevant real-time data in the memory.

2) Service Layer: This layer provides a secure, transparent and semantic-based data access, data sharing and data management for upper layers to build advanced application-oriented services. In large-scale network TSC system, it contains the basic services, for example, querying the communication state, querying the historical data, querying and configuring the scheme information, and so on.

This layer also provides ITIS-Oriented advanced services such as data mining service, data fusion service, information subscription service, event notification service and dynamic service flow management service. In large-scale network TSC system, it contains the data processing service, the instant control service, and so on.

3) ITIS Subsystem Layer: Based on the ITIS Advanced Service Layer, many ITIS subsystem can be easily developed and deployed. In large-scale network TSC system, it contains the part

of regional coordination and the Client.

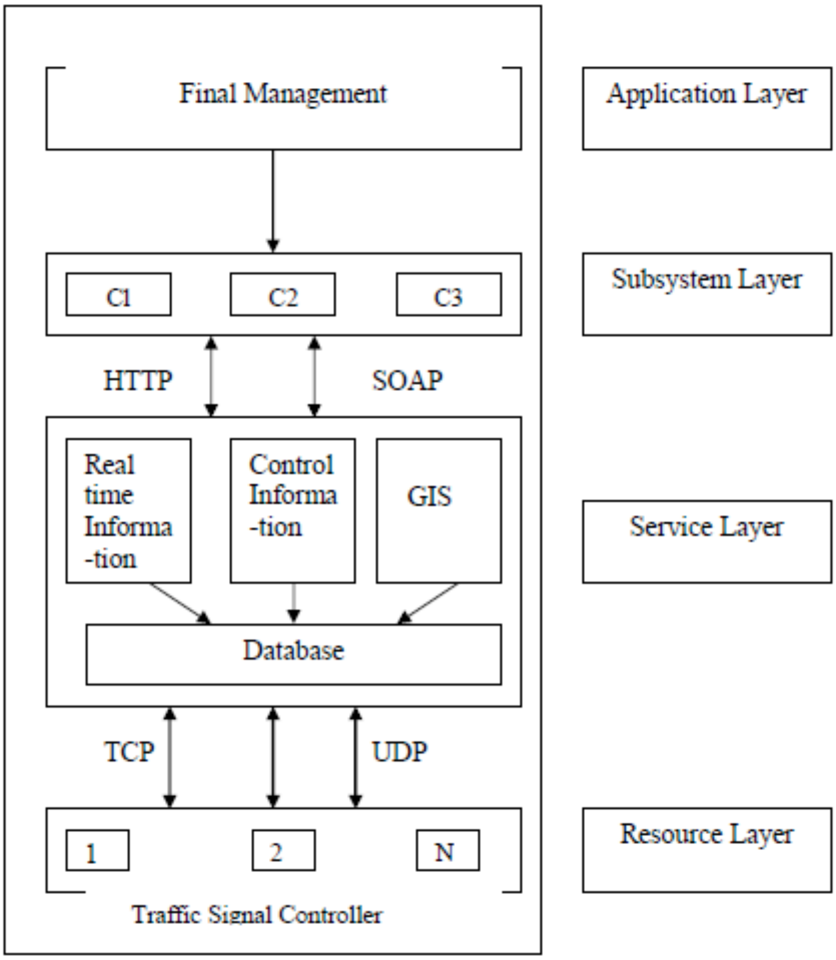


Figure 4.18. Architecture of Integrated Traffic Signal Controllers

4) ITIS Application Layer: This Layer contains many end users' requests for ITIS, which are defined as user services in National ITIS Architecture. This layer is the ITIS management platform extension of component object technology in the Internet, and it is also the component deployed in the Internet.

Here, above Architecture of integrated Traffic signal controller based on Service oriented architecture (SOA). Initially we have traffic signal controller at a single network or at single road intersection as shown by 1, 2, … N in above diagram. The output of each controller is then combined and put in a database, then we have three module 1. Real Time information (includes the current situation of the traffic) 2. GIS- A Geographic Information System (GIS), or Geographical Information System is any system that captures, stores, analyzes, manages, and presents data that is linked to location, and 3. Control Information or process used to control the overall process.

Then, we have sub sytem so that they control a particular set of network, as we are using both SOAP (Simple Object Access Protocol) for distributed system(SOAP, is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) as its message format, and usually relies on other Application Layer protocols (most notably Remote Procedure Call (RPC) and HTTP) for message negotiation and transmission. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. This XML based protocol consists of three parts: an envelope - which defines what is in the message and how to process it - a set of encoding rules for expressing instances of application-defined data types, and a convention for representing procedure calls and responses. or HTTP (Hyper text Transfer Protocol) for the same network. Both SMTP and HTTP are valid application layer protocols used as Transport for SOAP, but HTTP has gained wider acceptance as it works well with today's Internet infrastructure; specifically, HTTP works well with network firewalls.

We can also attach the controller of two or more consecutive road intersection. So that the overall delay for the traffic further optimizes. Suppose , we have two intersection X and Y. we can attach the output of X in Y. so that we further optimize the controller Y.

As Fuzzy traffic controller is much more efficient than the previous fixed time controller. As every time we are getting the real data or we can say, we are tuning the signal time as per dynamic scenario of roads. It gives the result as per our expectations. But the fuzzy controller is little bit expensive. We cannot use it over each road intersection. The controller can be used at main crossing in streets. Or the term crossing is itself fuzzy. It means we can have three types of crossing. First one is most busy and main crossing, second one is busy and third one simple crossing (where fixed time controller can be used).

For Main important crossing we can use fuzzy controller. For simple crossing we can use fixed time controller. But we require some optimal solution for busy crossing

Artificial Neural Networks:

An artificial neural network (ANN) [13], usually called "neural network" (NN), is a mathematical model or computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

The word *network* in the term 'artificial neural network' arises because the function $f(x)$ is defined as a composition of other functions $gi(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables.

A widely used type of composition is the *nonlinear weighted sum where* $f(x)=K\left(\sum w_i g_i(x)\right)$, where K (commonly referred to as the activation function) is some predefined function, such as the hyperbolic tangent. It is convenient to refer to a collection of functions gi as simply a vector

.

Traffic Controller Based on Artificial Neural Network:

To make the traffic control system responsive to the changing traffic conditions, optimization algorithm has to be applied each time the traffic conditions change. Since traffic conditions

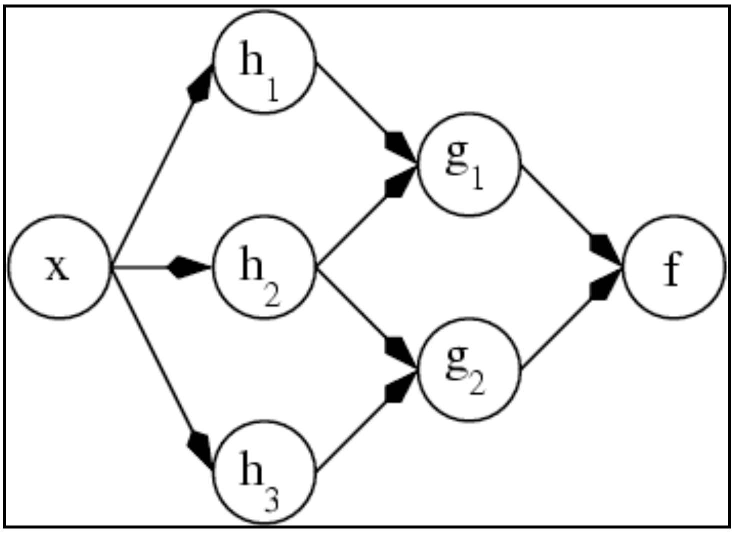


Figure 4.19. Artificial Neural Network [13]

change after every few seconds, this requires ample computer processing power to generate real time response. One solution to real time optimization is the closed loop system. Closed-loop systems operate either in Time of Day (TOD) mode or Traffic Signal Plan (TSP) mode. In TOD mode time plans are calculated for different times of the day and a plan is selected based on the current time.

In TSP mode, based on historical data, multiple traffic demand scenarios are extracted, demands with similar features are clustered and optimal timing plan is calculated for each cluster. Current traffic demand is mapped to one of the predetermined clusters and its timing plan is implemented. One major problem is the excess of setup factors and thresholds which need to be correctly determined for the successful implementation of TSP mode. Recent developments in the closed-loop system have improved setup methodology which has significantly decreased misclassification errors. but not completely eliminated. Besides this, other problems still exist, such as traffic demands in the same cluster always have some variations but a single time plan assigned to them, which flattens these variations. Also new traffic demand patterns appear with time and old patterns become obsolete. An Artificial Neural Network (ANN) based approach can be used to eliminate these problems associated with TSP mode of operations.

A closed-loop system is composed of a series of signalized intersections operated by a single master controller. The master controller issues commands to implement timing plans stored in the local controllers. Master controller coordinates the connected signalized intersections. In closed-loop systems a number of timing plans are calculated for different traffic scenarios and an appropriate plan is selected based on the current traffic conditions. The first step in designing a closed-loop system is to determine the total number of demand states. This can be achieved by recording traffic conditions for every possible traffic scenario. Subsequently these records are presented to a clustering algorithm to extract natural grouping into it.

Closed-loop system operates in two modes: Time Of Day (TOD) mode and Traffic Responsive Plan Selection (TSP) mode. In TOD mode it is assumed that traffic patterns repeat itself in recursive manner in time by the day of week. Hence a relationship is established between the time and demand. The selection function in TOD mode can be represented by

$$TP = f(T)$$

Where TP is the new timing plan and T is the Current time of day.

The problem with TOD mode is that the time-demand relationship is violated whenever special events, e.g. game matches, functions, convocations etc, occur. On the other hand TSP does not require this time- demand relationship. In TRPS mode real time data is collected from the detectors and optimal timing plan is selected based on the current traffic demand. Selection function in TRPS mode can be represented by.

$$TP = f(D)$$

Where D is the current traffic demand.

Since TSP mode depends on the real time data and does not require any time-demand relationship, which may induce errors, it can operate more efficiently if properly configured.

Mathematical Notation : If f is a function of a set of n variables and m constants represented by xi , where $i = 1,...,n$, and $c\ j$, where $j = 1,...,m$, respectively such that each ci either scales a variable xi in multiplicative manner or an additive scalar term in the function. Minor changes made to the scalars' set will not change the function landscape in a sense that no peak or valley will either disappear or change its neighborhood.

That is, minor changes in the scalars' set will only shift the landscape. Minor changes exclude changes that either reverse the sign or introduce high order increase or decrease in the magnitude of any member of the scalar set. Suppose f is a unimodal function and its optimizer * xt at time t is found. At time $t+1$, minor changes are introduced in the scalars' set which shift the landscape of f and invalidate * xt .

To find correct optimizer at time *t* +1, i.e. * *xt*+1 , optimization algorithm has to be applied again to *f* . Now suppose a set of hypothetical functions *gi* , where *i* = 1,...,*n* , which traces *x** in the shifted landscape at time *t* +1. Each *gi* is defined on (1, 2 ,..., *m*) Y_m ={ y_1, y_2, ... y_n} such that *y j* represents an offset from *c j* and *gi* gives the *ith* element of * *xt*+1 at time *t* +1 . It is clear from this definition that if $Y_m = (0,0,...,0)_m$ then * *xt* = * *xt*+1 . If the mathematical model of each *gi* could be found then *x** can be traced on the shifted landscape incase of any change introduced into the scalar set. This in turn can help us to avoid repetitive optimization of *f* which could save us from a lot of processing.

Algorithm Used : Algorithm used here for the ANN is Multilayer Perceptron Neural Network(MLP-NN). Figure below shows the configuration of a multilayer perceptron with one hidden layer and one output layer. In this MLP each neuron is connected to each neuron in the next layer. The output of the MLP is described by the following equation [14]:

$$y_p = F_O\left(\sum_{j=0}^{N} w_{jp}^{H}\left(F_h\left(\sum_{i=0}^{N} w_{ij}^{I} x_i\right)\right)\right)$$

for p = 1, 2 . . . N

Where:

• w^H_{jp} represents the weights from neuron j in the hidden layer to the pth output neuron
• x_i represents the ith element in the input layer
• F_H and F_O represent the activation functions in the hidden and output layers respectively.
• w^I_{ij} are the weights from neuron i in the input layer to the neuron j in the hidden layer.

The learning phase consists of the minimization of the cost [14]:

$$E = \frac{1}{2}\sum_{p=1}^{N}(y_P - d_p) = \frac{1}{2}\sum_{p=1}^{N} e_p^2$$

Where yp is the pth output value calculated by the network and and dp represents the expected value.

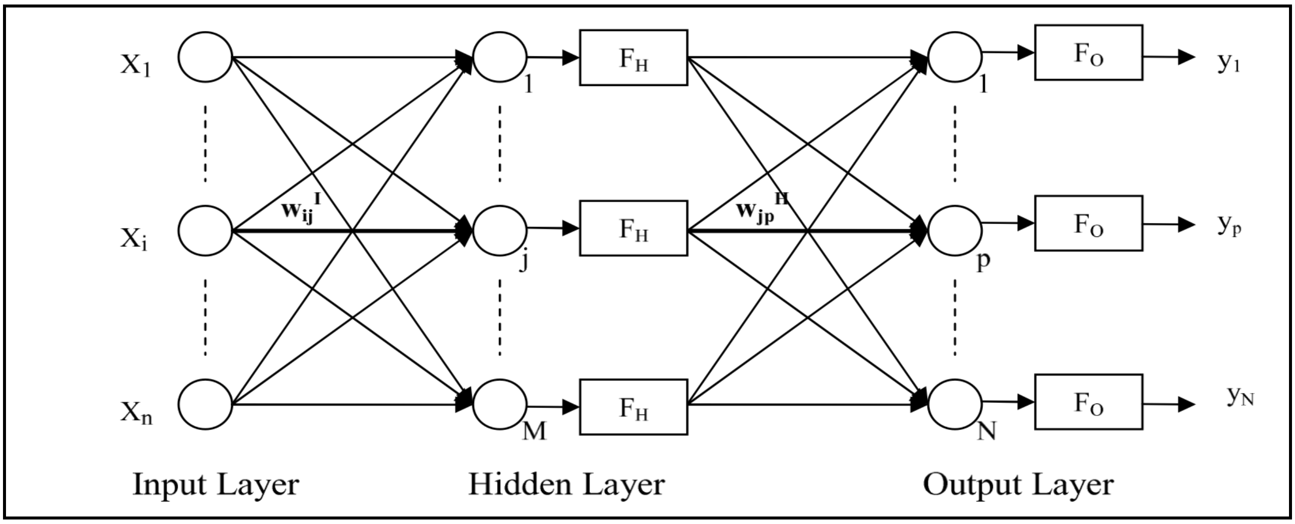


Figure 4.20. Fully connected Multi-Layer Perceptron with one hidden layer [14]

In ANN, here we have input as various plans for a particular cluster. The threshold is the performance index (average vehicle delay, fuel consumption) etc. the output will be one of the plan with better performance index (of various combination of plans). The new plan is further feed to new intersection, so that it can be optimized by time plan of previous intersection Algorithm used is Multilayer Perceptron Neural Network. The performance index can be improved by comparing with current performance index..

5. CONCLUSION & FUTURE WORK

The improvement of urban traffic condition is largely dependent on the modern ways of traffic management and control. As effective management and control equipment, advanced traffic signal controllers and control system contribute to the improvement of the urban traffic problem. An intelligent controller implemented Proposed here is a Fuzzy Controller with two approaches. Both approaches are simulated in MATLAB Tool. As in India, There is no separate queue for vehicles want to go straight and vehicles want to take right turn. Also time to cross the intersection depend upon the type of vehicle or Total free space between vehicles on road covered Fuzzy controller is definitely reducing the waiting time for a vehicle as compared to predefined signal controller.

1) First Approach is considers two input variable as Distance covered by vehicles on road and Number of Vehicles over the road. Numbers of vehicles are calculated by Total weight of vehicles over the road.
2) Second Approach Considers two input variables as Distance covered by vehicles over road and Total free space between vehicles over road. Total Free space between vehicles over road calculated by image processing of the picture of road.

Output of both Approaches is same i.e. time required to cross the traffic. When two approaches are compared. The result is seen as

1) As in Heavy traffic case, When distance covered is very small then number of vehicles (calculated by load of vehicles on road) is very less and free space between vehicles is medium then time calculated by both the approach is same i.e. very less and When distance covered is small then number of vehicles (calculated by load of vehicles on road) is very less and free space between vehicles is medium then time calculated by both the approach is same i.e. less. When distance covered is far then number of vehicles is less and free space between vehicles is medium then time calculated by both the approach shows significant difference. Approach 1 is showing less time (better result) than Approach 2. When distance covered is very far then number of vehicles is very large and free space between vehicles is large then time calculated by both the

approach is same i.e. large, so As per simulation results, Approach 1 for fuzzy Controller is slightly better than Approach 2 in case of heavy traffic.

2) In case of Light traffic, When distance covered is very small then number of vehicles (calculated by load of vehicles on road) is less and free space between vehicles is very then time calculated by both the approach is same i.e. less and When distance covered is small then number of vehicles (calculated by load of vehicles on road) is less and free space between vehicles is less then time calculated by both the approach is same i.e. medium. When distance covered is far then number of vehicles is large and free space between vehicles is less then time calculated by both the approach is same i.e. large. When distance covered is very far then number of vehicles is very large and free space between vehicles is very large then time calculated by both the approach is same i.e. large, so in all four case, Approach 1 for fuzzy Controller is giving approximately same result as Approach 2.

3) As in Mix traffic case, When distance covered is very small then number of vehicles (calculated by load of vehicles on road) is very less and free space between vehicles is very less then time calculated by the Approach 1 is very less where as for approach 2 is less i.e. approach 2 is giving better result (It takes more time to clear traffic) and When distance covered is small then number of vehicles (calculated by load of vehicles on road) is less and free space between vehicles is large then time calculated by the Approach 1 is medium where as for approach 2 is less. When distance covered is far then number of vehicles is large and free space between vehicles is less, time calculated by the Approach 1 is medium where as for approach 2 is large, In three cases, there is significant difference between time calculated by both the approach. In fourth case, When distance covered is very far then number of vehicles is large and free space between vehicles is large then time calculated by both the approach is same i.e. large, in this case, Approach 1 is allowing all traffic vehicles to pass where as Approach 2 is dividing the traffic into two parts, approach 2 allows one part to go while other part is to wait. So Approach 2 for fuzzy Controller is better than Approach 1 in case of Mix traffic.

After consider all three cases of types of traffic, The proposed work shows Approach 2 for fuzzy controller is Better than Approach1. Traffic signal control is a typical process, where traffic flows compete from the same time and space, and different objectives can be achieved in different traffic situations. This is benificial to both environment and economy like a decrease in each of delay, number of stops, fuel consumption, pollutant emissions, noise, and vehicle operating costs, queue length and personal time. Based on the Benefits, we can say that the fuzzy signal control can be multi objective and more efficient than conventional adaptive signal control.

Here we have calculated Number of vehicles over road by considering total weight of vehicles over road. Future Work involves the calculation of Number of vehicles by using Advanced Image Processing Techniques. Image processing techniques to calculate the Total free space between vehicles. Here we used only two input variables to design a fuzzy controller. Number of input variables can be increased (input variable can be taken as Distance covered by road, Free space between vehicles and Number of vehicles by using Advanced Image processing techniques), Result can be further analyzed by using these variables as input. Here the proposed work considered two lanes for road. Fuzzy Controller can be designed for more number of lanes. Servcie Oriented architecture is implemented for controlling the signals for whole city from single point. Neural Network is used to connect the two fuzzy controllers. so that output of one controller is effected by previous controller.

6. PUBLICATION FROM THESIS

During the period of working over this thesis work, I interacted with International community working on Fuzzy Logic. I discussed our approach for developing automatic traffic signal controller for roads using fuzzy logic with them and collected the reviews and worked over the suggestion send to us. A Research paper has been accepted in International conference ”CNC 2011” for presentation and published in their proceedings. The Proceedings is published by Springer LNCS-CCIS and it is available in the Springer Digital Library. This paper presents the factor effecting the duration of traffic signals and how fuzzy controller helps to automate the Traffic signals with effective results. Paper is attached at the end of thesis. The details of publication are as follows:

1. Conference Name : **“2nd International Conference on Advances in Communication, Network and Computing”, CNC 2011.**

URL	:	**http://cnc.engineersnetwork.org/2011**
Paper Title	:	**“Automatic Traffic Signal Controller for Roads by Exploiting Fuzzy Logic”.**
Publisher	:	**Springer LNCS-CCIS, CCIS 142, pp. 273–277, 2011.**
Location	:	**Bangalore, India.**
Conference Date	:	**Mar 10-11, 2011.**

2. One research paper titled **“Fuzzy Based Real Time Traffic Signal Controller to Optimize Congestion Delays”** is communicated in International conference. This paper presents the Design of fuzzy controller for traffic signals and its comparison and analysis results with earlier approach of fuzzy controller. The work considers all type of traffic on roads and delays caused by each type of traffic. All simulation work is done in Fuzzy Toolbox of MATLAB tool.

7. REFERENCES

[1] Hellmann M., “Fuzzy Logic Introduction”, University of Rennes, http://epsilon.nought.de/tutorials/ fuzzy/fuzzy.pdf.

[2] http://www.aptronix.com/fide/whyfuzzy.htm#fuzzy%20logic%20reduces%20the%20design %20development.

[3] J Jantzen, “Tutorial on Fuzzy Logic”, Technical University of Denmark, http://fuzzy. iau.dtu.dk/download/logic.pdf

[4] http://www.seattlerobotics.org/Encoder_mar98_fuz_fl_part4.html

[5] Jan Jantzen, “Design of Fuzzy Controller”, Technical University of Denmark, http://fuzzy. iau.dtu.dk/download/design.pdf.

[6] Sladoje Natasa, “Fuzzy Sets and Fuzzy Techniques”, Centre for Image Analysis, Uppsala University, 2007.

[7] http://www.dementia.org/~julied/logic/applications.html

[8] Lin Q, Kwan B.W., Tung L.J., “Traffic Signal Control using Fuzzy Logic”, IEEE Transaction, pp 1644-1649, 1997.

[9] G. Sankar, Kumar Saravana S,” Fuzzy Logic Based Automatic Braking System in Trains”, TENCON 2006, IEEE Region 10 Conference, pp. 1-4, 2006

[10] Tan Kok Khiang, Khalid Marzuki and Yusof Rubiyah, “Intelligent Traffic Lights Control By Fuzzy Logic”, Malaysian Journal of Computer Science, Vol. 9 No. 2, pp. 29-35, Dec 2006.

[11] Jarkko Niittymaki, “Using Fuzzy Logic to Control Traffic Signals at Multi-phase Intersections”, http://www.springerlink.com/content/981502q147x5t82g, 2007

[12] www.adobe.com/pdfs/Services_Oriented_Architecture_from_Adobe.pdf

[13] http://en.wikipedia.org/wiki/Artificial_neural_network.html

[14] Nouir Zakaria, Sayrac Berna, Fourestie Benoit, Tabbara Walid and Brouaye Francoise, " Comparison of Neural Network Learning Algorithms for Prediction Enhancement of a Planning Tool", 2007

[15] http://fuzzy.cs.uni-magdeburg.de/nfdef.html

APPENDIX A

AN INTRODUCTION TO HISTORY OF FUZZY LOGIC [1]

When Aristotle and his predecessors devised their theories of logic and mathematics, they came up with the so-called Law of the Excluded Middle, which states that every proposition must either be true or false. Grass is either green or not green; it clearly cannot be both green and not green. But not everyone agreed, and Plato indicated there was a third region, beyond true and false, where these opposites "tumbled about." In the Aristotelian world view, logic dealt with two values. In the 19th century, George Boole created a system of algebra and set theory that could deal mathematically with such two-valued logic, mapping true and false to 1 and 0, respectively. Then in the early 20th century, Jan Lukasiwicz proposed a three-valued logic (true, possible, false), which never gained wide acceptance. In 1965, Lotfi A. Zadeh of the University of California at Berkeley published "Fuzzy Sets," which laid out the mathematics of fuzzy set theory and, by extension, fuzzy logic. Zadeh had observed that conventional computer logic couldn't manipulate data that represented subjective or vague ideas, so he created fuzzy logic to allow computers to determine the distinctions among data with shades of gray, similar to the process of human reasoning.

Although, the technology was introduced in the U.S., U.S. and European scientist and researchers largely ignored it for years, perhaps because of its unconventional name. They refused to take seriously something that sounded so childlike. Some mathematicians argued that fuzzy logic was merely probability in disguise. But fuzzy logic was readily accepted in Japan, China and other Asian countries. The greatest number of fuzzy researchers today are found in China, with over 10,000 scientists. Japan, though considered at the leading edge of fuzzy studies, has fewer people engaged in fuzzy research. A decade ago, the Chinese University of Hong Kong surveyed consumer products using fuzzy logic, producing a 100-plus-page report listing washing machines, camcorders, microwave ovens and dozens of other kinds of electrical and electronic products.

E.H. Mamdani is credited with building the world's first fuzzy logic controller, after reading Dr. Zadeh's paper on the subject Dr. Mamdani, London University, U.K., stated firmly and unequivocally that utilizing a fuzzy logic controller for speed control of a steam engine was much Superior to controlling the engine by conventional mathematically based control systems and logic control hardware. Dr. Mamdani found that, using the conventional approach, extensive trial and error work was necessary to arrive at successful control for a specific speed set-point. Further, due to the non-linearity of the steam engine operating characteristics, as soon as the speed set-point was changed, the trial and error effort had to be done all over again to arrive at effective control. This did not occur with the fuzzy logic controller, which adapted much better to changes, variations and non-linearity in the system.

A major contributor to Homo sapiens success and dominance of this planet is our innate ability to exercise analysis and control based on the fuzzy logic method. Here is an example: Suppose you are driving down a typical, two way, 6 lane street in a large city, one mile between signal lights. The speed limit is posted at 45 Mph. It is usually optimum and safest to "drive with the traffic," which will usually be going about 45 Mph. How do you define with specific, precise instructions "driving with the traffic?" It is difficult. But, it is the kind of thing humans do every day and do well.

There will be some drivers weaving in and out and going more than 45 Mph and a few drivers driving less than 45 Mph. But, most drivers will be driving 45 Mph. They do this by exercising "fuzzy logic" - receiving a large number of fuzzy inputs, somehow evaluating all the inputs in their human brains and summarizing, weighting and averaging all these inputs to yield an optimum output decision. Inputs being evaluated may include several images and considerations such as: What are the cars in front doing? How fast are they driving. Any drivers going real slow? Any trucks holding up one of the lanes. How about side traffic entering from side streets. What do you see in the rear view mirror. Even with all this, and more, to think about, those who are driving with the traffic will all be going along together at very nearly the same speed. The same ability you have to drive down a modern city street was used by our ancestors to successfully organize and strategically carry out chases to drive wooly mammoths into pits, to obtain food, clothing and bone tools.

Human beings have the ability to take in and evaluate all sorts of information from the physical world they are in contact with and to mentally analyze, average and summarize all this input data into an optimum course of action. All living things do this, but humans do it more and do it better and have become the dominant species of the planet. If you think about it, much of the information you take in is not very precisely defined, such as evaluation of the behavior of a vehicle entering from a side street and the likelihood of the vehicle pulling in front of you. We call this fuzzy input. However, some of your "input" is reasonably precise and non-fuzzy such as the speedometer reading. Your processing of all this information is not very precisely definable. We call this fuzzy processing. Fuzzy logic theorists would call it using fuzzy algorithms (algorithm is another word for procedure or program, as in a computer program).

Fuzzy logic is the way the human brain works, and we can mimic this in machines so they will perform somewhat like humans (not to be confused with Artificial Intelligence, where the goal is for machines to perform EXACTLY like humans). Fuzzy logic control and analysis systems may be electro-mechanical in nature, or concerned only with data, for example economic data, in all cases guided by "If-Then rules" stated in human language.

APPENDIX B

AN INTRODUCTION TO OPERATIONS ON FUZZY SETS [6]

Standard fuzzy operations

for fuzzy sets A, B on a reference set X , given by the corresponding membership functions A(x) and B(x):

$\overline{A}(x) = 1 - A(x)$ − fuzzy complement
$(A \cap B)(x) = \min[A(x), B(x)]$ − fuzzy intersection
$(A \cup B)(x) = \max[A(x), B(x)]$ − fuzzy union
for all $x \in X$.

Properties of the standard operations :

1) They are generalizations of the corresponding (uniquely defined !) classical set operations.
2) They satisfy the cut worthy and strong cut worthy properties. They are the only ones that does.
3) The standard fuzzy intersection of two sets contains (is bigger than) all other fuzzy intersections of those sets.
4) The standard fuzzy union of two sets is contained in (is smaller than) all other fuzzy unions of those sets.
5) They inherently prevent the compound of errors of the operands.

Other generalizations of the set :

Aggregation operators : Aggregation operators are used to combine several fuzzy sets in order to produce a single fuzzy set.

2) **Associative** aggregation operations
   fuzzy intersections
   fuzzy unions
3) **Non-associative** aggregation operations
   averaging operations - idempotent aggregation operations

Fuzzy Complements:

1) A fuzzy complement cA of a fuzzy set A is given by a function c : [0, 1] → [0, 1].
2) Function c assigns a value to each membership value A(x) of $x \in X$ to the fuzzy set

A.

3) A membership function of the set cA is defined as cA(x) = c (A(x)). Note: the value. cA(x) is interpreted not only as the degree to which x ∈ X belongs to the fuzzy set cA, but also as the degree to which x does not belong to the fuzzy set A; the value cA(x) does not depend on x , but only on A(x).

Fuzzy complements :

Axiomatic requirements

Ax c1. c (0) = 1 and c (1) = 0. boundary condition

Ax c2. For all a, b ∈ [0, 1], if $a \le b$, then $c(a) \ge c(b)$. monotonicity c1 and c2 are called axiomatic skeleton for fuzzy complements

Ax c3. c is a continuous function.

Ax c4. c is involutive, i.e., c (c (a)) = a, for each a ∈ [0, 1].

Theorem

Let a function c : [0, 1] → [0, 1] satisfy Ax c2 and Ax c4. Then c satisfies Axioms Ax c1 and Ax c3 too. Moreover, the function c is a bijection.

Nested structure of the basic classes of fuzzy complements :

1) All functions c : [0, 1] → [0, 1] IsNotC(a) = a
2) All fuzzy complements (Ax c1 and Ax c2) c (a) = 1 for $a \le t$ 0 for $a > t$
3) All continuous fuzzy complements (Ax c1- Ax c3)
   c (a) = 12 (1 + cos πa) c (1/3)
4) All involutive fuzzy complements (Ax c1- Ax c4)
   $c\lambda (a) = 1 - a /1 + \lambda a$, $\lambda > -1$ (Sugeno class)
   $c\omega (a) = (1- a^{w})^{1/w}$,$\omega > 0$ (Yager class)
   cA(x) = 1 − A(x) Classical fuzzy complement

Generators :

Increasing generators : • Increasing generator is a strictly increasing continuous function g : [0, 1] → R, such that g (0) = 0.

• A pseudo-inverse of increasing generator g is defined as

$g^{(-1)}$ = 0 for a ∈ (−∞, 0)

= $g^{(-1)}$ (a) for a ∈ [0, g (1)]

= 1 for a ∈ (g (1), ∞)

• An example:

g (a) = a^p , p > 0

$g^{(-1)}$ (a) = 0 for a ∈ (−∞, 0)

= $a^{1/}p$ for a ∈ [0, 1]

= 1 for a ∈ (1, ∞)

Decreasing generators: Decreasing generator is a strictly decreasing continuous function f : [0, 1] → R, such that f (1) = 0.

• A pseudo-inverse of increasing generator f is defined as

$f^{(-1)}$ = 1 for a ∈ (−∞, 0)

= $f^{(-1)}$ (a) for a ∈ [0, f (0)]

= 0 for a ∈ (f (0), ∞)

• An example:

f (a) = 1 − ap, p > 0

$f^{(-1)}$ (a) = 1 for a ∈ (−∞, 0)

= $(1\text{-}a)^{1/p}$ for a ∈ [0, 1]

= 0 for a ∈ (1, ∞)

Generators :

Making one from another

Theorem : Let f be a decreasing generator f . The function g, defined by g (a) = f (0) − f (a), for a ∈ [0, 1] is an increasing generator, with g (1) = f (0).

Its pseudo-inverse is then given by $g^{(-1)}$ (a) = $f^{(-1)}$ (f (0) − a), for a ∈ R.

Theorem : Let g be an increasing generator. The function f , defined by f (a) = g (1) − g (a), for a ∈ [0, 1] is a decreasing generator, with f (0) = g (1).

Its pseudo-inverse is then given by $f^{(-1)}$ (a) = $g^{(-1)}$ (g (1)− a), for a ∈ R.

Generating fuzzy complements :

Theorem (First Characterization Theorem of Fuzzy Complements.)

Let c be a function from [0, 1] to [0, 1]. Then c is a (involutive) fuzzy complement iff there exists an increasing generator g such that, for all a ∈ [0, 1]

c (a) = $g^{(-1)}$ (g (1)− g (a)).

Theorem :

(Second Characterization Theorem of Fuzzy Complements.)

Let c be a function from [0, 1] to [0, 1]. Then c is a (involutive) fuzzy complement iff there exists an decreasing generator f such that, for all a ∈ [0, 1]

c (a) = $f^{(-1)}$ (f (0)− f (a)).

Generating fuzzy complements

Examples Increasing generators

Standard fuzzy complement: g (a) = a.

Sugeno class of fuzzy complements:

$g_\lambda(a) = (1/\lambda) \ln(1 + \lambda a)$, for $\lambda > -1$

Yager class of fuzzy complements: gω (a) = aω , for ω > 0.

Decreasing generators

Standard fuzzy complement: f (a) = −ka + k for k > 0.

Yager class of fuzzy complements: $f(a) = 1 - a^w$.

Fuzzy intersections

Definition

An intersection of two fuzzy sets A and B is given by a function of the form

$i : [0, 1] \times [0, 1] \rightarrow [0, 1]$.

A value is assigned to a pair of membership values A(x) and B(x) of an element x of the universal set X . It represents membership of x to the intersection of A and B:

$(A \cap B)(x) = i(A(x), B(x))$, for $x \in X$.

Note:

• Intuitive requirements to be fulfilled by a function I to qualify as an intersection of fuzzy sets are those of well known.

• The value (A ∩ B)(x) does not depend on x , but only on A(x) and B(x).