Summer 1998

Fuzzy logic applied to system control to enhance commercial appliance performance

Glenn Moffett
Louisiana Tech University

FUZZY LOGIC APPLIED TO SYSTEM CONTROL TO ENHANCE COMMERCIAL APPLIANCE PERFORMANCE

by

Glenn Moffett. B.S.E.E.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Engineering

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

August 1998

UMI Number: 9839743

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

August 3, 1998
Date

We hereby recommend that the dissertation prepared under our supervision

by Glenn Moffett

entitled Fuzzy Logic Applied To System Control To Enhance Commercial Appliance Performance

be accepted in partial fulfillment of the requirements for the degree of Doctor of Engineering.

Mickey D. Cox
Supervisor of Dissertation Research

Program Chair

Electrical Engineering
Program

Recommendation concurred in:

James D Lowther
Charles Dorsett
Louis Roemer
Dale Johnson

Advisory Committee

Approved:
Richard J. Greechie
Director of Graduate Studies

Leslie K. Guice
Dean of the College

Approved:
Terry McConathy
Dean of Graduate School

ABSTRACT

The purpose of this research is to determine the usefulness of fuzzy logic and fuzzy control when applied to a commercial appliance. Fuzzy logic is a structured, model-free estimator that approximates a function through linguistic input/output associations. Fuzzy rule-based systems apply these methods to solve many types of "real-world" problems, especially where a system is difficult to model, is controlled by a human operator or expert, or where ambiguity or vagueness is common.

This dissertation presents fuzzy sets, fuzzy systems, and fuzzy control, with an example conveying the use of fuzzy control of a consumer product and an overview of fuzzy logic in the field of artificial intelligence. Ultimately, it demonstrates that the use of fuzzy systems makes a viable addition to the field of artificial intelligence and, perhaps, more generally to the application of other consumer products to reduce energy consumption and increase the ease of operation.

Topics such as classical logic, set theory, fuzzy set theory, and fuzzy mathematics are developed in this research to provide a foundation in fuzzy logic. Fuzzy logic is an excellent development of a basic home appliance to a provide a powerful and user-friendly device. Fuzzy logic allows an engineer without a great knowledge of control systems and mathematical modeling a viable alternative in product creation. The fuzzy logic toolbox of the program MATLAB™ developed by

The Mathworks Corporation is used to build and test the fuzzy logic systems explored by this dissertation.

Again, in this dissertation the concept of fuzzy logic shall be explored in detail. Background and theoretical information shall be derived to provide a good base for applications. Classical logic, crisp sets, fuzzy sets, and operations on fuzzy sets are explained in order to cover a wide spectrum of applications. The focus or cumulating point will be to apply the fuzzy logic principle to any type of consumer appliance (such as a washing machine). The use of fuzzy logic will allow many household goods to be manufactured more quickly and with more options, and be energy efficient, user friendly, and cost effective.

TABLE OF CONTENTS

Page

Chapter

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF GRAPHS

LIST OF LISTINGS

Listing Page

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my mother, father, brother, friends, and other family members for their support and understanding. Without their continued moral support, none of the accomplishments of this project would have been possible. Also I would like to thank my committee: Dr. Mickey Cox, Dr. Louis Roemer, Dr. James Lowther, and Dr. Charles Dorsett.

CHAPTER I

INTRODUCTION

Many. many centuries ago Buddha philosophized the principle "A and not A." that "everything is a matter of degree." there are no sharply defined borders. the world is the color of gray rather than black and white. Fuzzy logic fits right in with this philosophy. It says that set membership functions are continuous. not discrete; that an object is "A" to some degree and "not A" to some degree. In many cases an object is 100% in "A." in other cases. an object may be 0% in "A." This is traditional single valued logic.

In 300 B.C.. the Greek philosopher Aristotle came up with binary logic (0.1). which is now the principle foundation of modern mathematics. It came down to one law: A or not A. either this or not this. For example. the sky is either blue or not blue. It cannot be blue and not blue. Every statement or sentence is true or false or has the truth value 1 or the false value 0. This is Aristotle's law of bivalence and was philosophically accepted for over two thousand years. Two centuries before Aristotle, Buddha had the belief that contradicted the theory of true and false. which threw aside the worship of binary logic and saw the world as it is. filled with contradictions. with things and not things. For example, he stated that the sky could be to a certain degree completely blue. but at the same time could also be at a certain degree not blue. That is. the sky can be blue and not blue at the same time. This is truly a leap of faith, because this implies that an entity can belong to a group or set of objects but not belong to that group or set of objects.

Table 1 - 1 : Glass Status Truth Table

Glass Full Status	True	False
Full	X	
Not Full		X
Half - Full	?	?
Half - Not Full	?	?

Conventional (Boolean) logic states that a glass can be full or not full of water. However, suppose one were to fill the glass only halfway. This is the gray area where binary logic ceases to become useful and falls apart. Everyone knows that a glass can be in a state of full, empty, half-full or half-empty. Clearly, this disproves Aristotle's law of bivalence. Table 1 - 1 displays the basic meaning of 2 - valued logic. Table 1 - 1 also shows that Boolean logic cannot explain the intermediate values of half-full and half-not full. This concept of multi-valence or multi-valued logic is the fundamental concept, which propelled Professor Zadeh of the University of Berkley in the 1960's to introduce fuzzy logic.[1]

Fuzzy logic is basically a multi-valued or infinite valued logic that allows intermediate values to be defined between conventional evaluations like *on/off, true/false, cold/hot,* etc. It links traditional logic (Boolean logic) with real world connotations. This idea by any means is not new (as told above) but has been overlook because of its imprecise definition. Therefore, Boolean logic is a special case of fuzzy logic. But without multi-valued logic systems descriptions such as *rather warm* or *pretty cold* can not be formulated mathematically and processed by computers. In this way an attempt is made to apply a more human-like way of thinking in the programming of computers.

Formally, fuzzy logic is a structured, model-free estimator that approximates a function through linguistic input/output associations. Fuzzy rule-based systems apply these methods to solve many types of "real-world" problems, especially where a system is difficult to model, is controlled by a human operator or expert, or where ambiguity or vagueness is common. A typical fuzzy system consists of a rule base, membership functions, and an inference procedure. Fuzzy systems are an alternative to traditional notions of set membership and logic that has its origins in ancient Greek philosophy, and applications at the leading edge of artificial intelligence. Yet, despite its long-standing origins, it is a relatively new field and leaves much room for development. This dissertation will present fuzzy sets, fuzzy systems, and fuzzy control, with an example conveying the use of fuzzy control of a consumer product and another example of fuzzy logic in the field of artificial intelligence. Ultimately, it will be demonstrated that the use of fuzzy systems makes a viable addition to the field of artificial intelligence and, perhaps, more generally to the application of other consumer products to reduce energy consumption and increase the ease of operation.

Sousa, Bose, and Cleland (1995) embarked on the improvement of adjustable-speed-drive system efficiency with the use of fuzzy logic, not only from the view point of energy savings and cooling system operation. but also from the broad perspective of environmental pollution.[5]

Lai. Nakano, and Hsieh (1996) proposed a system that combines the excellent speed regulation of the phase-locked loop techniques and the advantages of fuzzy logic (intuitiveness. simplicity. easy implementation. and minimal knowledge of system dynamics) to obtain a robust. fast. and precise control of motor speed.[4]

Mir. Elbuluk. and Zinger (1994) studied the fuzzy implementation of direct self-control of induction machines. In many applications. direct control of torque is very beneficial. There is always a need for torque control at different stages of machine operation. such as start-up and load disturbance. In this research. field oriented control schemes were considered to control the induction motor. However. field oriented control is highly dependent on machine parameters and speed. which is undesirable. Another scheme is stator direct self-control. which uses only one parameter in the control scheme. stator resistance. But in direct self-control the stator flux and torque are regulated to their command values by selecting the switching state that gives the proper changes in the flux and torque. This scheme uses error signals from electric torque. stator flux. stator position. and the stator flux vector. The state changes are determined from large error changes in the system. It is important to realize that large error changes can occur during startup and during a step change. This is undesirable because the system reacts sluggish or slow at these times. To overcome this drawback. fuzzy logic and expert knowledge of the system was called upon. Through the use of a single board computer. the starting flux and torque response and the responses to step changes in command torque with fuzzy implementation showed a considerable improvement over the conventional control. The intriguing part of the research is the fuzzy logic controller. There are three inputs (flux error, torque error, and flux position) and one output (inverter switching state). A total of 120 rules are used to control the system.[2]

Guillemin (1996) analyzed the different aspects of fuzzy logic in the control of a universal motor. In this work, fuzzy logic is implemented in a standard micro-controller to regulate the speed of a universal motor by real time adjustment of the motor current. The application of this study is to improve home appliance features, user friendly interfaces, and security features. The home appliance presented in Guillemin's research is a typical food processor. The food processor has a 400-W universal motor, which is supplied by a direct current source. A DC to DC converter, which is shown in figure 1 - 1, controls the motor's power supply. The DC to DC converter is classified as a buck type converter operating in the continuous region. The converter operates on the

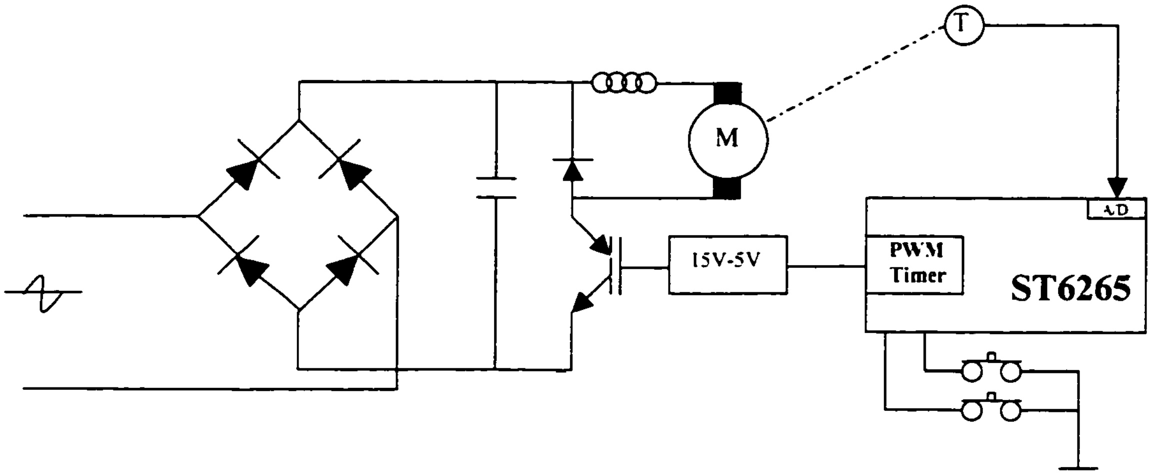


Figure 1 - 1 : System Diagram

principle of pulse width modulation (PWM). The frequency of the converter is not fixed hence it changes with tachogenerator voltage and PWM duty cycle. But before a fuzzy logic control system is implemented, the system behavior must be known. For this task experiments were made to yield graphs of open loop motor speed versus PWM duty cycle and tachogenerator voltage versus motor speed. The benefit of such data is not to obtain a model of the system, but specific characteristics of the system.

The two input variables used by the fuzzy logic controller are speed error and speed error variation. Speed error is equal to the measured speed subtracted from the target speed. The speed error variation measurement, just as the speed error measurement, takes advantage of the processor timer for a time base. Speed error variation is the last speed measurement minus the present speed $\Delta\varepsilon = V_{tacho}(n) - V_{tacho}(n-1)$. The system has one output variable, which is generated by the fuzzy inference kernel. The micro-controller calculation of the PWM duty cycle to be applied to the gate of the IGBT (Insulated gate bipolar transistor) is done as follows: $\delta\%(n) = \delta\%(n-1) \pm \Delta\delta$. It must also be noted that the duty cycle calculated by the micro-controller ranges from 0 to 100% with a resolution of 0.4%.

The development software used for this application is called "*fuzzy*TECH ST6 Explorer Edition", which covers all the steps of a fuzzy logic design from definition of the project, the linguistic variables, and the rules. In addition, the "*fuzzy*TECH ST6 Explorer Edition" also generates executable code for the micro-controller used in this application. There are four steps of project design when using the development tool: project definition, linguistic variables definition, rule definition, and system behavior optimization. The project definition is done by a part of the program known as the project editor. It is a graphical interface that allows the designer to directly access linguistic variables and rule definitions. This interface is far quicker than command line programming because of the function block, and also drop and connect ease of the software. The next step in project development is the linguistic variables definition. Membership functions of input and output variables are created by the graphical interface of the fuzzy logic software. When defining these variables the software allows the user to define two representations for the variables: the "shell value" and the "code value." The shell value exemplifies the real word value that the variables represent and the code values are the 8-bit internal values that the micro-controller uses to calculate results and range from 0 to 255. Membership functions for speed error, speed error variation, and PWM duty cycle variations are defined by triangular shapes and lines. It must be noted that the purpose of this step is to map the input values to linguistic variables and the output linguistic variables to a PWM duty cycle. The most important step of the project development is the rule definition. With too many rules the system can become restricted and sluggish, but with too few rules the system can become unstable at times. There must be a true medium meet between the two extremes for a well defined system. By an understanding of the system behavior, the rules were assigned accordingly. There were approximately 15 rules used to control the system as shown in Table 1 - 2. The structures of the rules are set up in an if-then type format. Speed error and speed error variation are defined by eight different classifications:

Table 1 - 2 : System Rules

Spreadsheet Rule Editor				
Matrix	**IF**		**THEN**	
Rule	**Speed Error**	**Speed Error Variable**	**DOS**	**PWM Variable**
1	Zero	Neg_Slow	1.00	Pos_Medium
2	Zero	Nul	1.00	Zero
3	Zero	Pos_Slow	1.00	Neg_Medium
4	Pos_Small	Neg_Slow	1.00	Zero
5	Pos_Small	Nul	1.00	Neg_Medium
6	Pos_Small	Pos_Slow	1.00	Neg_Medium
7	Neg_Small	Neg_Slow	1.00	Pos_Medium
8	Neg_Small	Nul	1.00	Pos_Medium
9	Neg_Small	Pos_Slow	1.00	Zero
10	Neg_Big	Nul	1.00	Pos_Big
11	Pos_Big	Nul	1.00	Neg_Big
12	Neg_Big	Pos_Slow	1.00	Pos_Medium
13	Pos_Big	Pos_Slow	1.00	Neg_Big
14	Neg_Big	Neg_Slow	1.00	Pos_Big
15	Pos_Big	Neg_Slow	1.00	Neg_Medium

Zero. Pos_small. Neg_small. Neg_Big. Pos_Big. Neg_Slow. Zero. Nul.Pos_Slow. and Neg_Slow. DOS stands for "degree of support" for rules defined. The degrees of support for the rules are set to one. meaning all rules have the same weight. When the inputs fall into the range of the 15 rules the output of the PWM duty cycle is mapped into five classes : Pos_Big, Pos_Medium. Zero, Neg_Big Neg_Medium. These classifications allow the user very easy implementation. interpretation. and also modification at a latter date. if needed. The last step in this program process is system behavior optimization. The interactive debugging mode allows a graphical verification of every design step even while the design is being performed. This is an off-line function of the program that allows the user to optimize the rules and membership functions. Another development tool is the batch mode. The batch mode records the output variables versus each input variables for testing and evaluating the design's performance. After all of the former tools are used the program generates executable code for the microprocessor.[3]

This is an excellent development of a basic home appliance to a very powerful and user-friendly device. It allows the designer without a great knowledge of control systems and mathematical modeling a viable alternative in product creation.

In the previous research cases. fuzzy logic was used to enhance a system with out the rigor of a precise mathematical model. But why doesn't fuzzy logic have a strong base in engineering sciences in the United States? Main stream engineering shows that this is a cultural match with eastern philosophies, and this is also why 30 years ago Dr. Zadeh and his ideas were eagerly accepted in Japan while they were being vehemently attacked and shouted down in the United States. As a result. Japan holds a commanding lead in fuzzy logic technology; the United States. motivated by market demands and not science or mathematics. is only now struggling to catch up. Again. in this dissertation the concept of fuzzy logic shall be explored fully. Background and theoretical information shall be derived to provide a good base for applications. Fuzzy sets. crisp sets. and operations on fuzzy sets will be explained so in order to cover a wide spectrum of applications. The focus or cumulating point will be to apply the fuzzy logic principle to any type of consumer appliance (such as a washing machine). This will allow many household goods to be manufactured more quickly and with more options. and be energy efficient. user friendly. and cost effective.

CHAPTER II

THE CREATORS OF SET THEORY AND FUZZY SETS

One of the most basic structures of mathematics is the concept of sets. The aim of several chapters later to come is to introduce and distinguish classical sets from fuzzy sets. But first a brief overview and history of the development of set theory is in order.

Sets give a precise definition to a collection of mathematical and non-mathematical objects. Once the concept of sets is established, one can compare them, define operations similar to addition and multiplication on them, and use them to define new objects such as various kinds of number systems. In fact, most of the topics in modern analysis are ultimately based on sets. Therefore, it is important to have a basic understanding of sets, and we will review an array of set topics in the next chapter.

The history of set theory is rather different from the history of most other areas of mathematics. For most areas a long process can usually be traced in which ideas evolve until an ultimate flash of inspiration, often by a number of mathematicians almost simultaneously, produces a discovery of major importance. Set theory, however, is rather different. It is the creation of one person, Georg Ferdinand Ludwig Philipp Cantor. Georg Cantor, born March 3, 1845, and died January 6, 1918, was a Russian-born German mathematician best known as the creator of Set Theory and for his discovery of the transfinite numbers. He also advanced the study of trigonometric series, was the first to prove the nondenumerability of the real numbers, and made significant contributions to dimension theory. Cantor received his doctorate in 1867 and accepted a position at the University of Halle in 1869, where he remained. Closely related to Cantor's work in transfinite set theory was his definition of the continuum as a connected, perfect set. He never doubted the absolute truth of his work, but following the discovery of the paradoxes of set theory, he left the defense of transfinite set theory to younger mathematicians such as David Hilbert, Bertrand Russell, and Ernst Zermelo.[23]

As Cantor developed set theory, Lotfi Zadeh conceived the concept of fuzzy set theory. Zadeh was born as Lotfi Aliaskerzadeh in 1921 in Baku, Soviet Azerbaijan. While at a young age of 16, Zadeh received several patents, one for the rotary engine. In 1942, he was one of only three electrical engineering students to gain a degree in electrical engineering from the University of Teheran in Iran. In the mists of World War II Zadeh left Iran for the United States. Upon reaching the United States he changed his name to Lotfi Asker Zadeh. In the fall of 1944, he entered MIT as a graduate student. He found MIT very easy, and not as rigorous as the University of Teheran. He received a master's degree in electrical engineering in 1946. By this time his parents had moved to New York, so Zadeh applied to Columbia University. He not only received admission in the Ph.D. program, but also a job as an instructor. In 1949 Zadeh earned his Ph.D. degree and became an assistant professor at Columbia a short time later. In 1965, 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 functions (or the values False and True) operate over the range of real numbers [0.0. 1.0]. He is now a Professor Emeritus and Director of the UC Berkeley's Initiative on Soft Computing. He has won numerous awards including an Honorary Doctorate from Paul-Sabatier in 1986. Japan's Honda Award in 1989. the IEEE Education Medal in 1973. the IEEE Centennial Medal in 1984. and the IEEE Richard W. Hamming Medal in 1992.[1]

CHAPTER III

CLASSICAL REASONING

Without basic set theory or classical sets. fuzzy logic would not be possible. And without classical logic. basic set theory would not exist. This is formal logic. the same logic of Greek philosopher Aristotle. which provided the first systematic account of correct forms of reasoning. These correct forms of reasoning can be condensed into five distinct areas: negation. conjunction. disjunction. implication. and equivalence. These five areas of reasoning, symbols. and English equivalents are shown in Table 3 -1 below.

Table 3 - 1 : Reasoning Forms

Operation	Symbol	English Equivalent
Negation	$\neg$	Not
Conjunction	$\wedge$	And; but; however
Disjunction	$\vee$	Or; unless
Implication	$\Rightarrow$	If.....then; only if
Equivalence	$\Leftrightarrow$	If and only if

The first form of reasoning is negation and it is a simple concept. Negation consists of a true and false adage. To negate a particular object is. in other words, to make a true statement false and a false statement true. Negation is also one of the oldest forms of reasoning. For example, a phrase saying: *the sky is blue* would be transformed by the negation process to *the sky is not blue.* This action is also demonstrated by the tables below.

Table 3 - 2 : Negation Equivalence (English Phrase versus Symbolic Representation)

English Phrase	Symbolic Representation
The sky is blue	a
The sky is NOT blue	¬a

Table 3 - 3 : Negation Truth Table

A	¬a
True (1)	False (0)
False (0)	True (1)

The next form of reasoning is conjunction. Conjunction is not a single value or a single phrase operation. The conjunction operation contains two statements separated by a conjunction such as and, but, however, etc. The rules of this operation are identical to the AND gate in electronics. A typical phrase, such as *the sky is blue*, then a conjunction, and then another phrase, such as *the weather is not good.* All of the former put together is *the sky is blue and the weather is not good* makes up a complex prepositional phrase. The tables below show how the conjunction function operates.

Also, another way to define the action of the conjunction is with a minimum statement shown in the following equation.

$$|a \wedge b| = \min\left[|a|,|b|\right] \text{ .... Equation 3 -1}$$

Table 3 - 4 : Conjunction Equivalence (English Phrase versus Symbolic Representation)

English Phrase	Symbolic Representation $(a \wedge b)$
The sky is not blue and the weather is not good	$(a \wedge b)$ = False
The sky is not blue and the weather is good	$(a \wedge b)$ = False
The sky is blue and the weather is not good	$(a \wedge b)$ = False
The sky is blue and the weather is good	$(a \wedge b)$ = True

Table 3 - 5 : Conjunction Truth Table

A	B	$a \wedge b$
False (0)	False (0)	False (0)
False (0)	True (1)	False (0)
True (1)	False (0)	False (0)
True (1)	True (1)	True (1)

The disjunction operation is very similar to the conjunction function in structure. The disjunction is true if at least one of the two propositions involved is true. The rules of this operation are identical to the OR gate in electronics. A typical phrase, such as *the sky is blue*, then a disjunction or, then another phrase, *the weather is not good.* All of the former put together is ***the sky is blue or the weather is not good*** makes up a complex prepositional phrase. Also, another way to define the action of the disjunction is with a minimum statement is shown in the following equation. The tables below show how the disjunction function operates for all combinations of inputs.[24]

$$|a \vee b| = \min\left[1, |a| + |b|\right] \quad \text{.......Equation 3 -2}$$

Table 3 - 6 : Disjunction Equivalence (English Phrase versus Symbolic Representation)

English Phrase	Symbolic Representation (a∨b)
The sky is not blue or the weather not good	(a∨b) = False
The sky is not blue or the weather is good	(a∨b) = True
The sky is blue or the weather is not good	(a∨b) = True
The sky is blue or the weather is good	(a∨b) = True

Table 3 - 7 : Disjunction Truth Table

A	B	a∨b
False (0)	False (0)	False (0)
False (0)	True (1)	True (1)
True (1)	False (0)	True (1)
True (1)	True (1)	True (1)

Another form of reasoning is implication. Implication is a consequential type of reasoning that includes an if...then format. This type of reasoning is prevalent in many if not the majority of computer programming languages. This type of reasoning is not as restrictive as conjunction, but its format is not similar to conjunction or disjunction. Implication requires a sense of forethought and a higher level of reasoning, than the former three types.

The two parts of this type of reasoning are the antecedent and the consequent. The antecedent is the *if* part of the statement, and the consequent is the *then* part of the statement. The only odd conclusion that is involved with this logic is when the antecedent is false and the consequent is true. When the former two situations occur, the total statement is true. This is hard to believe, but when the antecedent is false one cannot disprove the consequent. Therefore the whole statement is true. An example antecedent, such as *if the sky is not blue*, then a consequent phrase, *the weather is good*. All of the former put together is *if the sky is not blue then the weather is good*, makes up a complex prepositional phrase. This is a true statement, meaning when the sky is not blue the weather is good. The function (Equation 3 – 3) and truth tables (Tables 3 –8 and 3 – 9) that describe this type logical reasoning are found below.

$$|a \Rightarrow b| = \min\left[1, 1 + |a| - |b|\right] \text{.......................................Equation 3 – 3}$$

The last type of reasoning is equivalence. Equivalence is a very simple concept to implement. The format is similar to the implication type of reasoning. But, this type of reasoning is totally inclusive or exclusive. In this type of format, as the previous, it includes an antecedent and a consequent. The antecedent and consequent are separated by the connecting phrase *if and only if*. The equivalence operation cannot be compared to a single electronic gate, because it takes a combination of gates to achieve the equivalence result. This is also called combinational reasoning or semi-complex reasoning. Complex reasoning relies on a mix and match of the former four concepts to achieve a desired result. The equivalent equation, constructed from the previous types of reasoning, is represented by Equation 3 – 4, whereas the equation for the truth-functional is illustrated in Equation 3 - 5. The truth tables that describe this logical function are found in the Tables 3 - 10 and 3 - 11.

Table 3 - 8 : Implication Equivalence (English Phrase versus Symbolic Representation)

English Phrase	Symbolic Representation (a⇒b)
If the sky is not blue then the weather is not good	(a⇒b) = True

If the sky is not blue then the weather is good	(a⇒b) = True
If the sky is blue then the weather is not good	(a⇒b) = False
If the sky is blue then the weather is good	(a⇒b) = True

Table 3 - 9 : Implication Truth Table

A	B	a⇒b
False (0)	False (0)	True (1)
False (0)	True (1)	True (1)
True (1)	False (0)	False (0)
True (1)	True (1)	True (1)

$$(a \Rightarrow b) \wedge (b \Rightarrow a) \text{ ....Equation 3 – 4}$$

$$|a \Leftrightarrow b| = |a| \cdot |b| + |\neg a| \cdot |\neg b| \text{ ....Equation 3 – 5}$$

Table 3 - 10 : Equivalence Equivalent (English Phrase versus Symbolic Representation)

English Phrase	Symbolic Representation (a⇔b)
The sky is not blue if and only if the weather is not good	(a⇔b) = True
The sky is not blue if and only if the	(a⇔b) = False

weather is good	
The sky is blue if and only if the weather is not good	(a⇔b) = False
The sky is blue if and only if the weather is good	(a⇔b) = True

Table 3 - 11 : Equivalence Truth Table

A	B	a⇔b
False (0)	False (0)	True (1)
False (0)	True (1)	False (0)
True (1)	False (0)	False (0)
True (1)	True (1)	True (1)

These five former types of reasoning are the basics on which all reasoning is founded. There are several other types of reasoning, not explained in this chapter, such as complex propositions, contradictions, tautologies, etc. These all are a combination of the five basic forms of reasoning mentioned in detail in the chapter.

In conclusion the five types of reasoning lay a foundation for set theory and in turn, fuzzy set theory. Classical reasoning, even though basic, provides a true insight to the fundamental background of set theory operations.

CHAPTER IV

CLASSICAL SET THEORY

Set theory is the natural evolution of classical reasoning in the truest sense. The evolution of set theory was done by one person. Georg Cantor. Once basic set theory is established. one can compare sets. define operations similar to addition and multiplication on them. and use them to define new objects such as various kinds of number systems. In fact. most of the topics in modern analysis are ultimately based on sets and the manipulation of these sets.

Any collection of objects is called a set. and set theory is the study of the relationships existing among sets. Set theory underlies the language and concepts of modern mathematics--both pure and applied. The study of sets, especially infinite ones. has also become a fascinating branch of mathematics in its own right. Set theory began with the work of Georg Cantor in the 19th century, but its roots in logic go back much further--to Aristotle and Plato. The prevailing view in mathematics today is that every mathematical object can ultimately be described as some sort of set. A set may be specified in one of two basic ways. The roster method. or tabulation method, simply lists all the elements in the set. The descriptive method, or set-builder notation, gives a rule for determining which things are in the desired set and which are not. For an example, a set may contain a collection of all seventh grade students. To simplify this notation one can express and should express the set as shown in Equation 4 - 1. But this set is a member of a larger and more infinite set called a universal set. The universal set contains all sets and subset of the particular represented sets. The universal set of these seventh graders may contain all junior high school students in the

B = {x | all seventh grade students}....................................Equation 4 - 1

state. A Venn diagram of this universal set and the previously defined set of seventh graders are shown in Figure 4 – 1. B can be called a subset of the universal set A. These definitions of sets are very precise or crisp. A student is contained in the universal set or the student is not contained. There is no half or partial inclusion in this set definition.

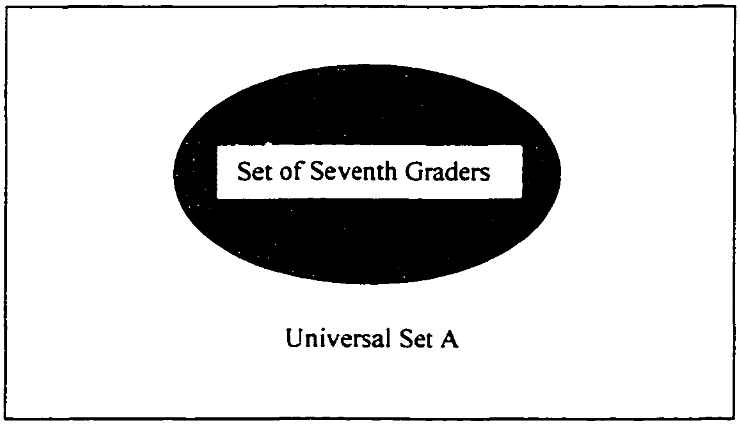


Figure 4 - 1 : A Set Defined in a Universal Set.

The versatility of set theory is the concept of subsets. Subsets are sets contained within another set. All data that is contained in the subset is also contained within the inclusive or parent set. The previous example set, which contains the seventh graders, is actually a subset of the universal set. But this is somewhat misleading because any set is a subset of the universal set. For a true example of subsets the set of seventh graders can by divided into seventh grade boys and seventh grade girls. A Venn diagram of this subset in relation to the previous set and universal set is shown in Figure 4 – 2. The equations for the two sets that contain all the seventh grade boys and seventh grade girls are show in Equations 4 – 2 and 4 – 3. It must be noted that a pair of

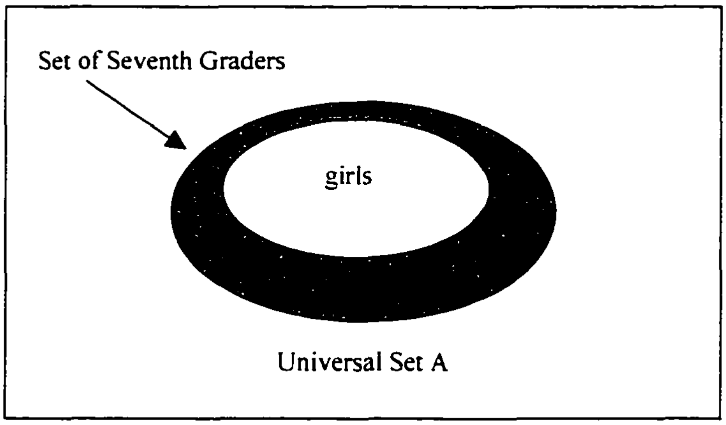


Figure 4 - 2 : A Set and Subset Defined in a Universal Set.

$C = \{x \mid \text{all seventh grade girls}\}$ .... Equation 4 - 2

$D = \{x \mid \text{all seventh grade boys}\}$ .... Equation 4 - 3 braces (curly brackets) surrounding its description designates a set. Bear in mind that not every description that seems to make sense actually denotes a set. If it did, many inconsistencies, such as Russell's Paradox, would arise.

Russell's Paradox is the most famous of the logical or set-theoretical paradoxes. The significance of Russell's Paradox can be seen once it is realized that, using classical logic, all sentences follow from a contradiction. In the eyes of many, it therefore appeared that no mathematical proof could be trusted once it was discovered that the logic and set theory apparently underlying all of mathematics was contradictory. The paradox itself stems from the idea that any coherent condition may be used to determine a set (or class). Attempts at resolving the paradox therefore have typically concentrated on various means of restricting the principles governing the existence of sets. Naive set theory contained the so-called unrestricted comprehension (or abstraction) axiom. This is an axiom, first introduced by Georg Cantor, to the effect that any predicate expression P(x), containing x as a free variable, will determine a set. The set's members will be exactly those objects which satisfy P(x), namely all x's which are P. It is now generally agreed that such an axiom must be either abandoned or modified. Russell's response to the paradox is contained in his theory of types. His basic idea is that we can avoid reference to S (the set of all sets, which are not members of themselves) by arranging all sentences into a hierarchy. This hierarchy will consist of sentences about individuals at the lowest level, sentences about sets of the next lowest level, etc. It is then possible to refer to all objects for which a given condition (or predicate) holds only if they are all at the same level or of the same "type."[23]

The paradox arises within naive set theory by considering the set of all sets. which are not members of themselves. Such a set appears to be a member of itself if and only if it is not a member of itself. This seems to be a contradiction of terms but in a way this is a paradox. Most people think of a paradox in the sense of time travel. If a person could go back into time. before they were born and murder their mother or father. a paradox insues. The paradox is. when you return to your own time you could not have been born. and. in turn. you cannot exist.

The basic types of sets are the universal set. subset. standard set and the power set. The power set. not mentioned previously. consists of all possible subsets of a given set X. The expression for the power set is shown in Equation 4 – 4. But when a power set of X is finite containing n elements. the number of subsets of X are 2^n. This

$$P(X) \qquad \text{Equation 4 - 4}$$

can be shown in Equation 4 – 5.

$$|P(X)| = 2^n \qquad \text{Equation 4 - 5}$$

These previous types of sets can be manipulated in four elementary set operations: complement. intersection. union. and difference. For a brief description, the complement of A in X is A - X. the set of all elements in A that are not in X. If every element of a set A is also an element of the set B, set A is a subset of B. The intersection of A and B is the set of all elements that are in both A and B. The union of A and B is the set of all elements that are either in A, or in B, or in both. For difference A is subtracted from B or B is subtracted from A.

To achieve a more detailed development of elementary set manipulations, equations and graphical methods will be employed.

To complement a set, true boundaries must be defined to begin any set operations. The complement of a single set contained in a universal set is done by noting the set as shown in Equation 4 - 6. If the double complement or compound noting is done to a set, then the conclusion is the original set. The complement of a universal set is a null set or empty set. The universal set contains all possible sets and subsets, and the opposite of containing everything is to contain nothing. Shown in Equations 4 - 7 and 4 - 8 are null set and double negation operations, respectively. A graphical representation of a set complement is shown in Figure 4 - 3. The portion of Figure 4 – 3 that is labeled "not A" is the complement of A. The complement of A is also defined by the subtraction of the universal set and set A.

$\overline{A} = \{x \mid x \in X \textit{ and } x \notin A\}$ ....Equation 4 - 6

$\overline{X} = \emptyset$ and $\emptyset = \overline{X}$ ....Equation 4 - 7

$\overline{\overline{A}} = A$ ....Equation 4 - 8

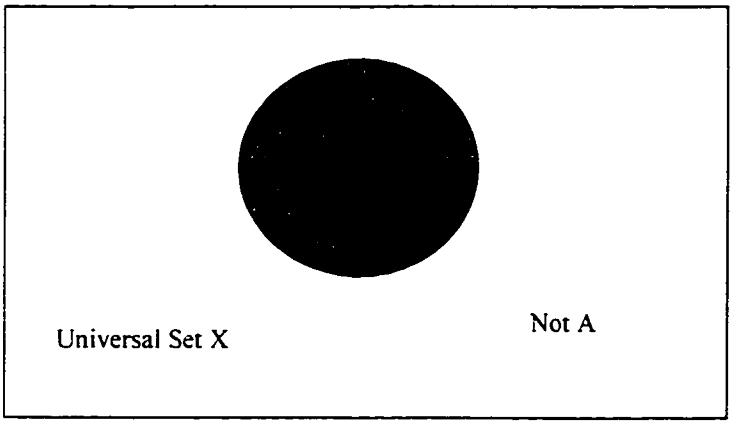


Figure 4 - 3 : Complement of Set A

The next elementary set operation is union. The union operation takes place between two distinct sets. The universal set is not included in the operation but the universal set contains the operation. But when the universal set is in union with another set the product is the universal set. as shown in Equation 4 – 9. When the union operation is imposed on two sets. those two sets actually become one set. All contents

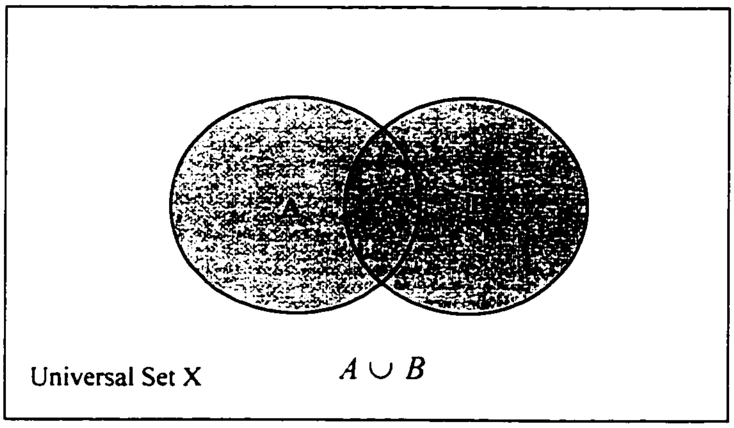


Figure 4 - 4 : Union of Set A and B

of the first set and the second set are combined within the universal set. The Venn diagram in Figure 4 - 4 shows the union operation graphically. The union operation can also express the *law of the excluded middles* as illustrated in Equation 4 – 10. Most importantly, the union operation of two sets contained in a universal set is expressed formally in Equation 4 – 11.

$$A \cup X = X \quad \text{.......... Equation 4 – 9}$$

$$A \cup \overline{A} = X \quad \text{.......... Equation 4 – 10}$$

$$A \cup B = \{x \mid x \in A \ or \ x \in B\} \quad \text{.......... Equation 4 – 11}$$

The intersection operation is very similar to the union operation on sets. When the union operation is acted on two sets, the sets are combined to represent one unified set. However, the intersection operation does not join both sets completely unless the sets are equal. Intersection between two sets can be represented in Equation 4 – 12.

$$A \cap B = \{x \mid x \in A \ and \ x \in B\} \quad \text{.......... Equation 4 – 12}$$

Where the two sets overlap or intersect is where the above definition is true. A good example of the operation is when given two sets, one containing all tall people and another set containing all women, the intersection between these two sets would be all tall women. The intersection would not contain all the contents of the set, tall people or all women. But the intersection will contain a portion of both sets, as pictured in Figure 4 – 5 by the dark elliptical region.

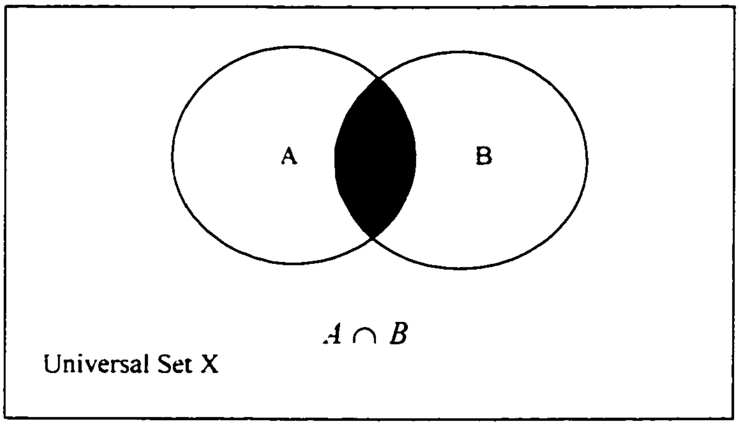


Figure 4 - 5 : Intersection of Sets A and B

Some other interesting facts concerning intersection is the intersection with subsets and the intersection with null sets. If set A contains all the contents of set B and more, then B is a subset of A. The formal notation of this action is shown in Equation 4 - 13.

$B \subseteq A$ ....Equation 4 – 13

There can be more than one subset for a given set. When there is an intersection between a set and its subset, the result can only be the subset. The reason is that the subset is intersecting with its own contents. When a null set or a set that contains no objects is intersected with a normal set, however the result is a null set. The Equations for both of these operations are shown in Equations 4 – 14 and 4 – 15, respectively.

$$If\ B \subseteq A\ then\ A \cap B = A$$ ....Equation 4 – 14

$$A \cap \varnothing = \varnothing$$ ....Equation 4 – 15

The last elementary set operation is difference. The difference operation is not symmetrical like the union or intersection operation. That is, the difference of set A and set B is not the same as the difference of set B and set A. This concept is described formally in Equation 4 – 16. A good example of intersection is to again consider a set of all tall people (set A) and a set of all women (set B). The difference operation can be applied to these two sets in two different ways. In Figure 4 – 6 set A is subtracted from set B, meaning the set of all tall people is subtracted from the set of all women. The result is all tall people excluding all tall women. As in Figure 4 – 7, the reverse is considered. The set of all women is subtracted from the set of all tall people. Hence, the result is all women excluding the tall women. The difference operation can also be applied to the universal set and an included normal set. The difference between the

$$A - B = \{x \mid x \in A\ and\ x \notin B\}$$ ....Equation 4 – 16

universal set and a normal set is the complement of the normal set. When the situation is reversed, the result is a null or empty set. Equations 4 – 17 and 4 – 18 show these previously discussed relationships.

$X - A = \overline{A}$ ....Equation 4 – 17

$\varnothing$

$A - X =$ ....Equation 4 – 18

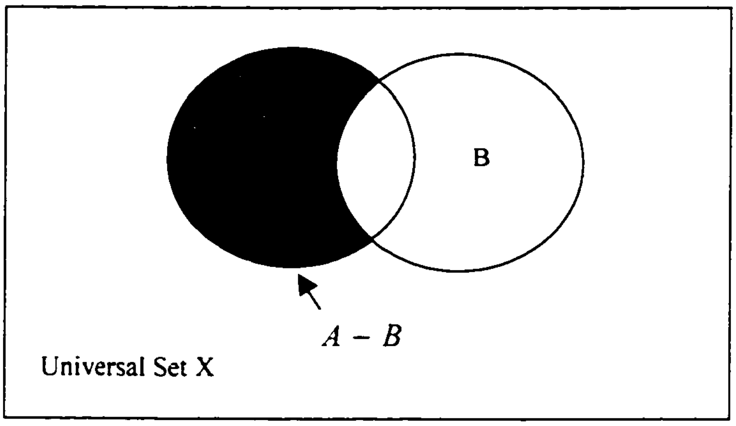


Figure 4 - 6 : Difference of Sets (A - B)

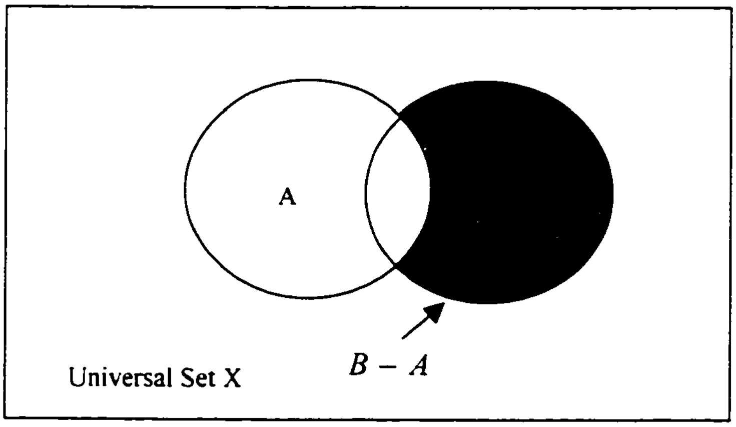


Figure 4 - 7 : Difference of Sets (B - A)

Now that the elementary set operations have been defined. more complex combinations and theoretical set manipulations can be explored. The next section includes some of the most popular identities and theories associated with complex set theory. Table 4 – 1 illustrates six very popular set manipulations. It is very apparent that these six set manipulations build upon all of the elementary set operations.

Table 4 – 1 : Set Properties

Property Name	**Property Set Equivalent**
Absorption	$A \cap (A \cup B) = A.A \cup (A \cap B) = A$
De Morgan Laws	$\overline{A \cap B} = \overline{A} \cup \overline{B}.\ \overline{A \cup B} = \overline{A} \cap \overline{B}$
Commutativity	$A \cup B = B \cup A.\ A \cap B = B \cap A$
Associativity	$A \cap (B \cap C) = (A \cap B) \cap C.\ A \cup (B \cup C) = (A \cup B) \cup C$
Idempotence	$A \cap A = A.\ A \cup A = A$
Distributivity	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C).$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

The first set identitified is absorption. Absorption utilizes the union and the intersection operation. As shown in Table 4 – 1. absorption results in a single set solution. There is a union between set A and set B that is denoted in parentheses. Then the result of the former equation is intersected with set A. This gives a result of set A. This is illustrated below in Figure 4 – 8. Two examples of this property can be shown by a set of all tall people and by a set of all women. The union between the set of all tall people and the set of all women gives a product of a set of all tall women. The second half of the equation is the intersection of tall people and all tall women. Since the set of all tall women is a subset of all tall people. the result of this action is the set of all tall people. The second example is equivalent to the first example. Given an intersection between the two sets. all tall people and all women. The result is a set of all tall women. The second step is to union the previous set with the set of all tall people. The conclusion is a set of all tall people. This conclusion is also shown in Figure 4 – 9. The same result was obtained by the previous example. proving that these two equations are interchangeable.

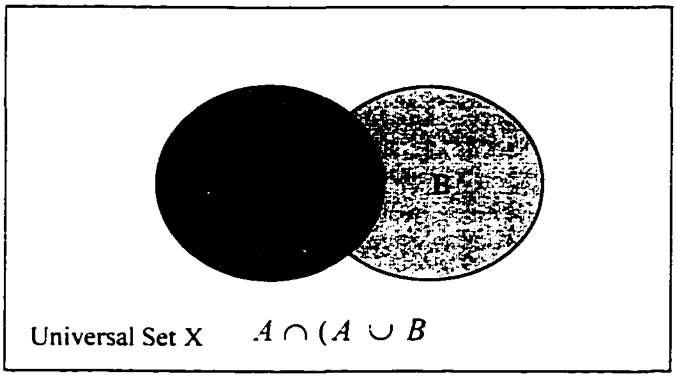


Figure 4 - 8 : Absorption Sets

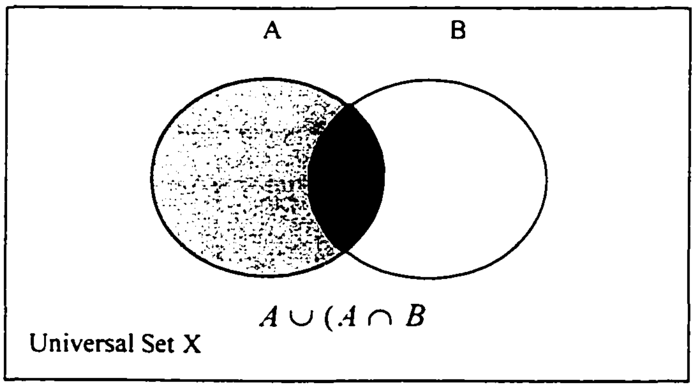


Figure 4 - 9 : Absorption Sets

The next property is known as De Morgan's laws, which are named after Augustus De Morgan, an important innovator in the field of logic. In addition, he had many contributions to the field of mathematics and the chronicling of the history of mathematics.

Augustus De Morgan was born in Mandura, India, on June 27, 1806. His father was a colonel in the Indian Army. His family soon moved to England where they lived first at Worcester and then at Taunton. His early education was in private schools where he learned Latin, Greek, Hebrew, mathematics and a dislike of exams. He entered Trinity College, Cambridge, in 1823, and graduated four years later.

After graduation, De Morgan reached the point of deciding what to do with the rest of his life. Dubious of competitive fellowships and master degrees, he refused to continue his education. Fearful of hypocrisy and religious bigotry, he also rejected his parents' wish of becoming a priest. After contemplating medicine and law, he finally decided to become a mathematician. In 1828, he was awarded the position of first Professor of Mathematics at University College in London.

His time at the university was far from quiet. In 1831, he resigned on principle after another professor was fired without explanation. He regained his job five years later when his replacement died in an accident. He would resign again in 1861. As a teacher he was highly praised at making mathematics alive and interesting to his students. In addition, he wrote textbooks on numerous subjects in mathematics and logic.

He was married in 1837 to Sophia Frend, who would later write his biography. During his life, De Morgan was constantly involved in various activities. A member of the Astronomical Society and the Society for the Diffusion of Useful Knowledge, he founded the London Mathematical Society and was its first president. He wrote many books and articles on mathematics, logic, philosophy and many other subjects. In addition, he assembled a large personal library of over 3000 books, a vast feat considering he was never wealthy. Unfortunately with all his work, he had little time for the rest of his life, but he was known as a kind and humorous individual. Augustus De Morgan died on March 18, 1871, in London, England. His library was later donated to the London University library.

De Morgan contributed many accomplishments to the field of mathematics on many different subjects. He was the first person to define and name "mathematical induction" and developed De Morgan's rule to determine the convergence of a mathematical series. His definition of a limit was the first attempt to define the idea in precise mathematical terms. In addition, he also devised a decimal coinage system, an almanac of all full moons from 2000 B.C. to 2000 A.D. and a theory on the probability of life events which is used by insurance companies.

However, De Morgan's biggest contribution was in the field of logic. His most important work, Formal Logic, included the concept of the quantification of the predicate, an idea that solved problems that were impossible under the classic Aristotelian logic. For example, the following is only workable using De Morgan's method: In a particular group of people,

- most people have shirts
- most people have shoes
- therefore. some people have both shirts and shoes.

He devised the idea around the same time as a Scottish philosopher. Sir William Hamilton. who accused him of stealing his ideas. However. it is clear that De Morgan's work is clearer. more developed and all around superior to Hamilton's version. With no evidence to back him up. the Scot's charge of plagiarism has been dismissed as sour grapes. De Morgan's other works include a system of notations for symbolic logic that could denote converses and contradictions and the famous De Morgan laws.

De Morgan laws rely heavily on the negation principle. The two sets involved are manipulated by an intersection or union and then negated. As shown in Table 4 – 1. De Morgan laws have two parts such as the absorption property. but the concepts of De Morgan laws are more involved than absorption. The set A and set B can be easily intersected. as done earlier in this chapter. The negations of the intersected sets are shown in Figure 4 – 10 by the dotted area.

De Morgan laws state that the complement of the intersection of two sets is equivalent to the union of their individual complements and the complement of the union of two sets is equivalent to the intersection of their individual complements.

Commutativity is the third property of set operations. The commutativity operation proves that the set order can be interchanged without any effect to the result. The union or intersection operation can be applied to sets and the former will hold. It

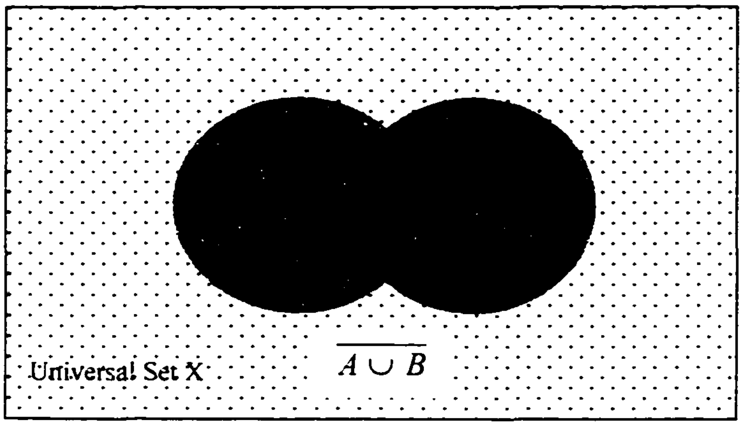


Figure 4 - 10 : De Morgan Laws

does not take much effort to prove this operation. Consider two sets, one of all tall people (set A) and the other of all women (set B). When set A is intersected with set B, the set order does not matter. The result will still be the same, a set of tall women.

The next set operation is called associativity. Associativity deals with the ordering of the operation. This is true with the union or intersectioin operation. The operation is non-dependent on parentheses placement, meaning that the result of the operation will be the same no matter where the parentheses are placed. The AND function which is used in digital logic, displays the associativity operation described in Table 4 – 2. The associated gates with this truth table are illustrated in Figures 11 and 12.

Idempotence is the next property of sets. Idempotence deals with the intersection and union operations. According to this property, the intersection of any set with itself results in the original set. This is also true when the union operation is applied to a specified set. This property is very useful when simplification is needed to collapse long set equations. Idempotence can also be used when collapsing several union operations contained in a singular function. This operation is shown in Equation 4 - 19.

$$A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n = \bigcup_{i=1}^{n} A_i \text{ ....Equation 4 – 19}$$

Table 4 - 2 : AND Function of Associative Property

X = (AB) C				Y = (BC) A			
A	B	C	X	A	B	C	Y
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	0	0	1	1	0
1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1

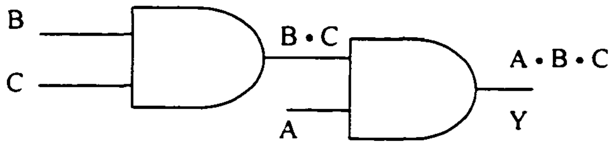


Figure 4 - 11 : Gate Equivalent of Associative Property

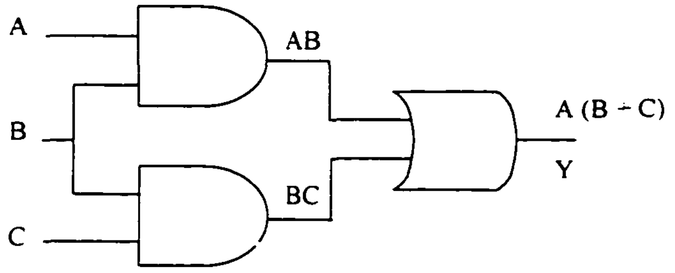


Figure 4 - 12 : Gate Equivalent of Distributive Property

The last property associated with set operation is called distributivity. This property is used extensively in combinational logic and linear algebra. As shown in Table 4 – 1 a set function can be easily expanded or collapsed with this property. The associated gates with this truth table are illustrated in Figures 13 and 14. The AND function used in digital logic displays the distributivity operation described in Table 4 – 3.

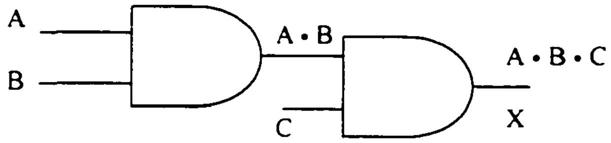


Figure 4 - 13 : Gate Equivalent of Distributive Property

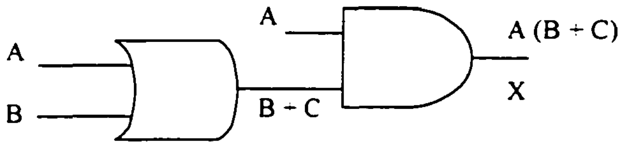


Figure 4 - 14 : Gate Equivalent of Distributive Property

Table 4 - 3 : AND Function of Distributivity Property

X = A (B + C)				Y = AB + AC			
A	B	C	X	A	B	C	X
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	0	0	1	1	0
1	0	0	0	1	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

The last subject associated with classical set theory is the characteristic function. This concept is very important in fuzzy set theory and will materialize in later chapters.

The basic idea of a function is to map or correlate a set of data to another set of data. The function's purpose is to link both sets of data. A simple function is a set of data complemented. The set could contain data such as hot, tall, boy, and wet. The result of the function could be cold, short, girl, and dry. This is by no means the extent of function operation or complexity, but it shows very compactly function operation. Function operation is not limited to sets or set theory, although that is the length on which functions will be used in this text.

The definition of a characteristic function is to declare elements of X which are members of the set and are not members of the set. In concept X denotes the universe of discourse, or universal set. Equation 4 – 20 illustrates the characteristic function, X_A,

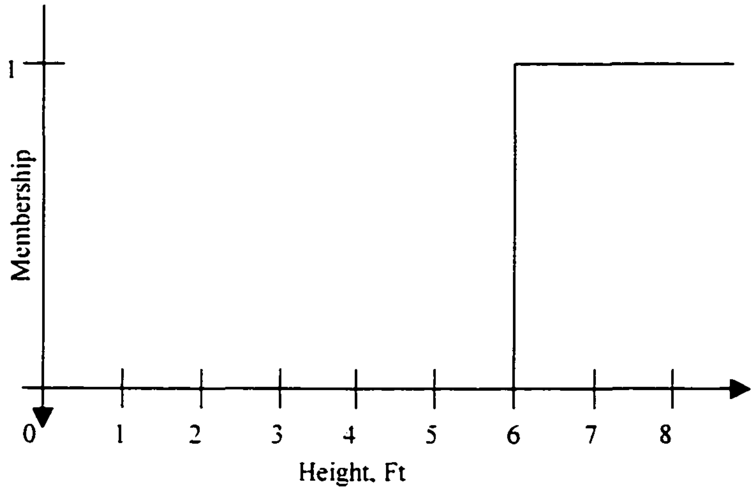
$$X_A(x) = \begin{cases} 1 & for\ x \in A \\ 0 & for\ x \notin A \end{cases} \quad \text{..........Equation 4 – 20}$$

of set A. As shown by Equation 4 – 20, there cannot be an element belonging and not belonging to the set. The correct terminology is actually a subset of the said set. Moreover, this is the downfall of crisp theory.

For the lower limit of all tall people being 6 feet, all people under this criteria will be considered short. In Graph 4 – 1 there is a distinct line between tall and short people. This line restricts a 5 foot and 11 inch person from being tall, which is truly ridiculous in real life.

The characteristic function can also be used with various set operations. The most common and popular operations used in conjunction with the characteristic function are complement, union, and intersection, but it is not uncommon to see the other operations as well. The associated equations for the complement, union, and intersection operations are shown in Equations 4 - 21, 4 - 22, and 4 - 23, respectively.

$$X_{\bar{A}}(x) = 1 - X_A(x) \quad \text{..........Equation 4 – 21}$$



Graph 4 – 1 : Membership of Tall and Short People

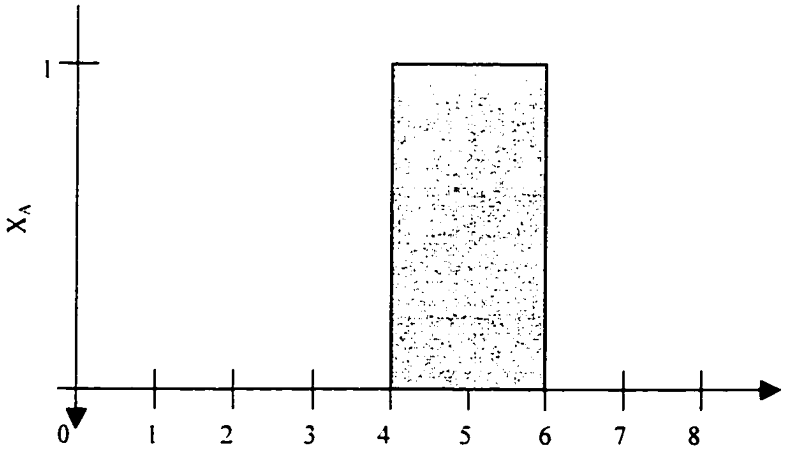
$X_{A \cup B}(x) = \max(x_A(x), x_B(x))$ .... Equation 4 – 22

$X_{A \cap B}(x) = \min(x_A(x), x_B(x))$ .... Equation 4 – 23

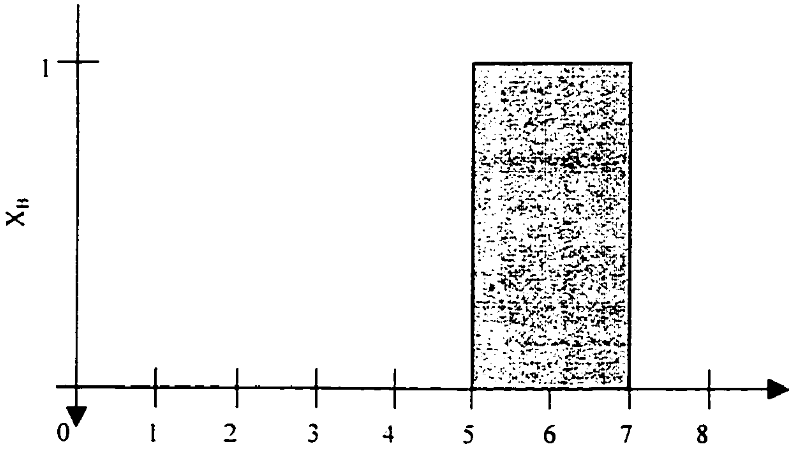
To explain these concepts in greater detail, two characteristic functions are defined with these operations in mind. The first function is of all short people, which includes heights from 4 feet to 6 feet tall. The next function shall include all tall people with heights from 5 feet to 7 feet tall. Both of these functions are shown in Graphs 4 – 2 and 4 – 3. These former functions are somewhat skewed from real world occurrence, but for the sake of demonstration the function definitions make good examples.

The first operation is complement and is carried out on function X_A. The result of the complement operation is heights less the 4 feet and heights greater than 6 feet. The union operation utilizes both functions. The union action takes the maximum values of both functions and disregards the minimum. This is done by combining all values of both sets to achieve a result of heights between 4 and 7 feet. Conversely the intersection action only returns the minimum of both functions. Hence the result is heights between 5 and 6 feet.

The set operations and characteristic functions have several faults when applied to certain types of sets and tested against real world situations. From these anomalies fuzzy set theory becomes an invaluable tool. In the next chapter fuzzy set theory will address these problems.



Graph 4 – 2 : Characteristic Function of Short People



Graph 4 – 3 : Characteristic Function of Tall People

CHAPTER V

FUZZY SET THEORY

When a person gathers data, no matter how precise he or she is there will be a level of inaccuracy. But in all data gathering rounding is assumed, and some bad or inaccurate data enters the data set for many reasons. The former is dreadfully true in engineering applications, where the range of data can vary in the ratios greater than one million. But how can data be taken from the field and interpreted to a satisfactory and usable level of acceptance?

Acknowledging that the data is somewhat imprecise is the first step. The next step is to represent the data in a way that the inaccuracy does not compromise the situation further. But even explaining the set of data can be inaccurate. Telling someone the data taken is pretty good, tolerable, or even good enough, can make the situation very confusing. Inaccuracies in measurement and verbal explanations make it almost impossible to take data and represent it correctly. One cause is the English language. Not realized by most people, the English language is very complex and ambiguous. Fuzzy logic is an organized method, which allows the digital world of the computer to deal with the imprecision of data, especially that which deals with human reasoning. In reality, information is often puzzling and unclear. However, people have the ability to sort through muddled information and come out with sound conclusions. In fuzzy logic, the imprecise data being considered are called fuzzy sets.[18]

A fundamental concept is the distinction between fuzzy sets and crisp sets. Crisp sets represent any set of data that denote membership by a certain criteria. Suppose there is a set of all young women (18 –25 years of age) in a small town in the United States and that a modeling agency would like a new crop of models for the spring catalogues. One criterion that the agency is looking for is tall women. This is not the only criterion for a model, but this is a start. On the surface it would seem a easy task to pick all the tall women and disregard the rest, but it is not. The reason for this difficulty is the definition of tall.

Table 5 – 1 : Set of All Potential Models

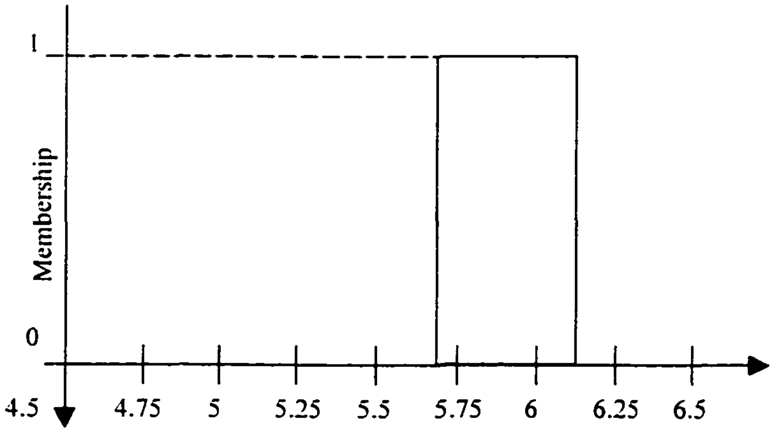
Name	**Height**
Angela	4 feet 8 inches
Anne	5 feet 2 inches
Bobbie	5 feet 5 inches
Crystal	5 feet 6 inches
Diana	5 feet 7 inches
Gloria	5 feet 7 inches
Jane	5 feet 8 inches
Jessie	5 feet 11 inches
Karen	6 feet 1 inches
Susan	6 feet 3 inches

Everyone would agree that a 7-foot woman is tall but how about a 5 foot 7 inch woman. In crisp sets there must be a criteria to discern between tall and short women. To pick this criterion, experience is needed in the specified field of concern. The modeling agency would be the most qualified to ascertain such a height. Suppose the agency picked 5 feet and 7 inches as the minimum height and 6 feet 2 inches as a maximum of height. From this criterion Table 5 – 2 was spawned including 5 women. The master or universal set of all women in consideration is shown in Table 5 – 1.

Table 5 – 2 : Set of all Potential Models that Poses Suitable Height

Name	**Height**
Diana	5 feet 7 inches
Gloria	5 feet 7 inches
Jane	5 feet 8 inches
Jessie	5 feet 11 inches
Karen	6 feet 1 inches

The new set in Table 5 – 2 contains all women taller than 5 feet 6 inches and shorter than 6 feet 2 inches. This set of data is also demonstrated with a membership function of all tall women in Graph 5 – 1. The crisp set does truly represent the criteria, that the modeling agency specified. But when considering a tall person one inch does not make that person short or tall. But how can a crisp set relay this apparent fact to the resulting data? The answer is that it cannot. The problem with crisp sets is the interpretation of boundaries. The boundaries in the model agency example are 5 feet 7 inches and 6 feet 2 inches. Crisp sets are too precise to make a reasonable judgement. When a person says another person is tall, the difference of one inch is not a factor in the decision. But with crisp sets, one inch can mean the difference between inclusion and exclusion. Fuzzy sets take into account the gray area of data under consideration as opposed to a set being considered as simply true or false. The gray area in our example is the region of data adjacent to 5 feet 7 inches and 6 feet 2 inches. These data gray areas are represented in a one-dimensional Euclidean space diagram in Figure 5 – 1. Gray areas are where fuzzy sets reign superior over crisp sets. Fuzzy sets embrace the gray areas and incorporate them into the set operation. In other words, fuzzy sets allow partial membership in the set structure, whereas crisps sets by definition have no way to incorporate the concept of partial membership.[21]



Graph 5 – 1 : Membership of Tall Women Models

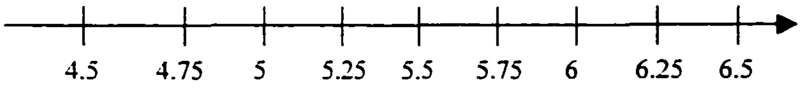


Figure 5 – 1 : Grey Area Mapped on a One Dimensional Euclidean Space

The foundation of a fuzzy set is the membership function. As shown in Equation 5 - 1 the membership function assigns to each element x of X a number. A(x). in the

A: X → [0. 1]..Equation 5 - 1

closed unit interval [0. 1] that characterizes the degree of membership of x in A. It must be noted that when one is defining a membership function. the universal set X is always assumed to be a classical set. To transform a crisp set to a fuzzy set a technique known as mapping is used. Mapping is not a new concept; it has been around a very long time. The process is very simple: data is transferred from one set to another set. As an example. consider again the modeling agency and. more importantly. the models. Each model must be assigned a membership value between 0 and 1. This value or numbers shows the degree of memberships displayed by each model. The number 0 would categorize the model of having no membership or very short (greatly less than 5 feet 7 inches) or very tall (greater than 6 feet 2 inches). All of the heights in between shall have a specific level of membership. This level of membership is graduated depending on the height of the model. The closer she is in the interval height of 5 feet 7 inches to

6 feet 2 inches the higher degree of mapping she receives. To carry crisp sets one step further a full representation of all potential models is shown in Table 5 – 2 and Graph 5 - 2 shows the membership function with a fuzzy interval. This is a graph of a fuzzy set and a intermediate step to achieve fuzziness. This graph is similar to 5 – 1 in some ways but the most important difference is the trapezoidal function shape. There are two lines drawn from the top of the trapezoid to the x-axis. which represents the interval of total inclusion or where every mapped data point is equal to 1. This shape includes all data before and after the critical or included set. In Table 5 – 3 the complete sets of potential models are mapped to a fuzzy set using the concept of degree of membership. One can see from Table 5 – 3 that the taller the person is the higher degree of membership is allocated. until a height greater than 6 feet 1 inch is reached. This is truly different than crisp set theory where the potential models under 5 feet 7 inches and over 6 feet 2 inches were disregarded and were not in membership. But with fuzzy sets these models are in contention to become a model.

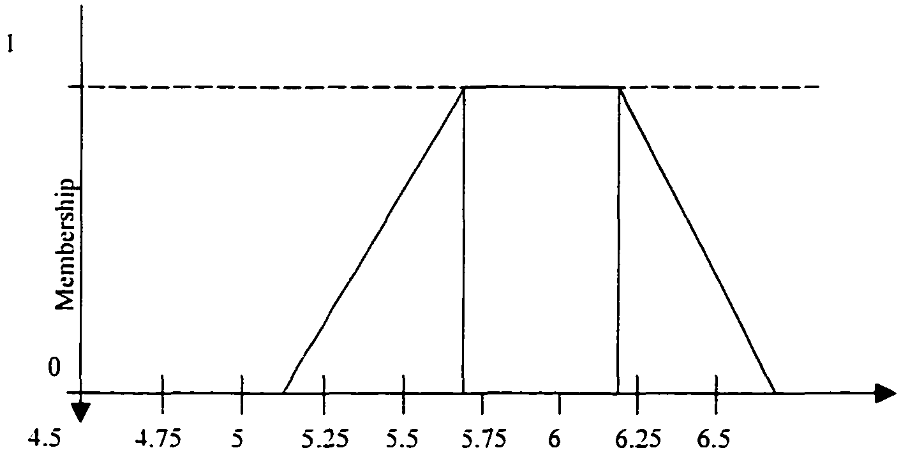
Another point about Table 5 – 3 is that there are three sets listed. One set is the name of every model. the second set is the height of the models. and the third set. which is the fuzzy set. is the degree of membership of each model. Graph 5 – 2 can be used in the realm of fuzzy logic and is used in many fuzzy programs. because of its simplicity and ease of manipulation. The down side is that data does not usually follow a trapezoidal function in nature. Data usually conforms to open ended curves, bell shaped curves, Gaussian distributions. and so on. This typical membership function is known as a fuzzy interval because it satisfies the following.

1. A is normal.
2. The support {x : A(x) > 0} of A is bounded.
3. The α - cuts of A are closed intervals.

The concept of a fuzzy quantity being normal assumes it has a value of one. The other two properties will be explored later when fuzzy numbers are considered. This function can be simplified by deleting one of the model agencies criteria. The model height is a factor but too short out weighs being too tall. so the 6 feet 2 inch criteria is abolished.

Table 5 – 3 : Set of All Potential Models Mapped to a Fuzzy Set

Name	Height	Degree of Membership
Angela	4 feet 8 inches	0
Anne	5 feet 2 inches	0.2
Bobbie	5 feet 5 inches	0.5
Crystal	5 feet 6 inches	0.8
Diana	5 feet 7 inches	1
Gloria	5 feet 7 inches	1
Jane	5 feet 8 inches	1
Jessie	5 feet 11 inches	1
Karen	6 feet 1 inches	1
Susan	6 feet 3 inches	0.8

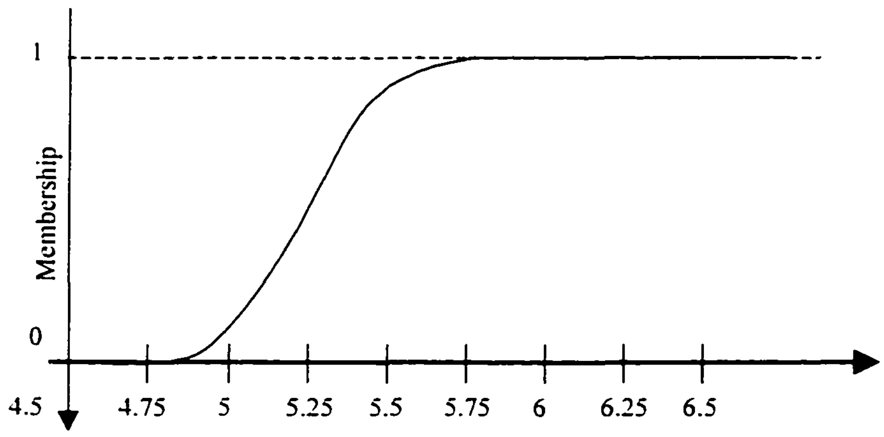


Graph 5 – 2 : Membership Function with a Fuzzy Interval

Table 5 – 4 : Set of All Potential Models Mapped to a Non-Interval Fuzzy Set

Name	Height	Degree of Membership
Angela	4 feet 8 inches	0
Anne	5 feet 2 inches	0.2
Bobbie	5 feet 5 inches	0.4
Crystal	5 feet 6 inches	0.6
Diana	5 feet 7 inches	.8
Gloria	5 feet 7 inches	.8
Jane	5 feet 8 inches	1
Jessie	5 feet 11 inches	1
Karen	6 feet 1 inches	1
Susan	6 feet 3 inches	1

This will allow simplification of the example until fuzzy numbers are introduced. With this assumption fact Table 5 - 4 is created. Another criteria is that 5 feet 8 inches is now the ideal height and every model above that height is considered in full membership. This graph shows a gradual and smooth transition from the short potential models to the potential models of 5 feet 8 inches and taller.



Graph 5 – 3 : Membership Function of Potential Models without Fuzzy Interval

In Equation 5 – 2, the degree of membership is listed in ordered pairs for potential models instead of tabular format. This equation format is found in popular fuzzy logic papers, as are Equations 5 – 3 and 5 – 4. In Equation 5 – 5, x1, x2, and so on are known as labels for corresponding potential model names. The generalized

A = { <Angela, 0>, <Anne, 0.2>, <Bobbie, 0.4>, ...Equation 5 – 2
<Crystal, 0.6>, <Diana, 0.8>, <Gloria, 0.8>,
<Jane, 1>, <Jessie, 1>, <Karen, 1>, <Susan, 1>}

$A = \{ <x_1, 0>, <x_2, 0.2>, <x_3, 0.4>, <x_4, 0.6>, <x_5, 0.8>,$ ....Equation 5 – 3

$<x_6, 0.8>, <x_7, 1>, <x_8, 1>, <x_9, 1>, <x_{10}, 1> \}$

A = 0/Angela, 0.2/Anne, 0.4/Bobbie, 0.6/Crystal, 0.8/Diana,.......Equation 5 – 4

0.8/Gloria, 1/Jane, 1/Jessie, 1/Karen, 1/Susan

notation for these membership functions is represented in Equation 5 – 5. Finally, the popular notation for a fuzzy set is μ of X, a function from the reference set X to the unit interval, is shown in Equation 5 – 6.

$A = \sum A(x)/x$ ....Equation 5 – 5

$\mu : X \rightarrow [0, 1]$ ....Equation 5 – 6

Operations on fuzzy sets are very similar to the operations done on crisp sets. The main difference is that the rules of crisp sets do not transfer over to fuzzy sets. There are three main operations considered in fuzzy set theory. These operations are complement, union, and intersection.

The complement of fuzzy set A defined on a universal set X, concludes that $\overline{A}$ is another fuzzy set on X that inverts the degrees of membership associated with A. This definition is totally different from crisps sets. In crisp set the complement of set

A, contained in a universal set X, would have all the data that was not included in set A. But as stated before, a fuzzy set can overlap its complement. The complement of all potential models are shown in Table 5 – 5. The standard complement, $\bar{A}$, of fuzzy set A is defined by Equation 5 - 7 and illustrated with A in Graph 5 - 4. The fuzzy set complement has a property that contradicts complement properties of crisp sets. This property is called equilibrium of points and is described in Equation 5 – 8. This can

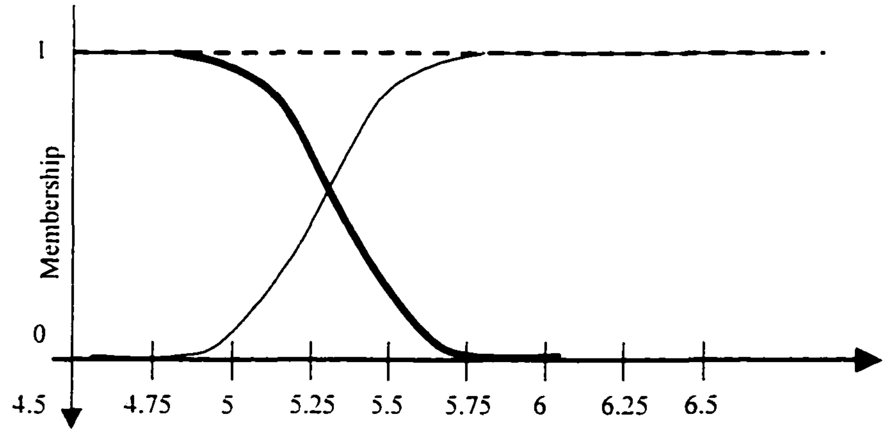
Table 5 – 5 : Set of All Potential Models Membership Functions and Complements

Name	**Degree of Membership of A**	**Complement of A**
Angela	0	1
Anne	0.2	0.8
Bobbie	0.4	0.6
Crystal	0.6	0.4
Diana	0.8	0.2
Gloria	0.8	0.2
Jane	1	0
Jessie	1	0
Karen	1	0
Susan	1	0

only happen at one point of the two curves. The reason it can only occur once is because the fuzzy set has to be convex, meaning that no α - cut can intersect a fuzzy

$\overline{A}(x) = 1 - A(x)$ ....Equation 5 – 7

$\overline{A}(x_0) = A(x_0)$ .... Equation 5 – 8



Graph 5 – 4 : Membership Function of Potential Models and Complement

curve twice. The α-cuts are discussed in detail in the next chapter, along with other advanced properties of fuzzy sets. The equilibrium point of fuzzy set A is near the value of 0.5. This point shows that set A and its complement share the same point. Crisp sets have a theorem call the law of the excluded middle to contradict this action. For crisp sets, taking the union of set A and its complement, $\overline{A}$, the result must be the universal set X, as shown in Equation 5 – 9.[15] This is absolutely is not true for fuzzy sets. as can be observed by Graph 5 – 4. This situation more apparent in the next paragraph.

$$A \cup \bar{A} = X \text{..Equation } 5-9$$

The next operation is the union or standard fuzzy union. The standard fuzzy union is an operation between two fuzzy sets. The concept uses max. an abbreviation for the maximum operator. to achieve the union function. This function is shown in Equation 5 – 10. and used in the model example shown in Table 5 – 6. The law of the

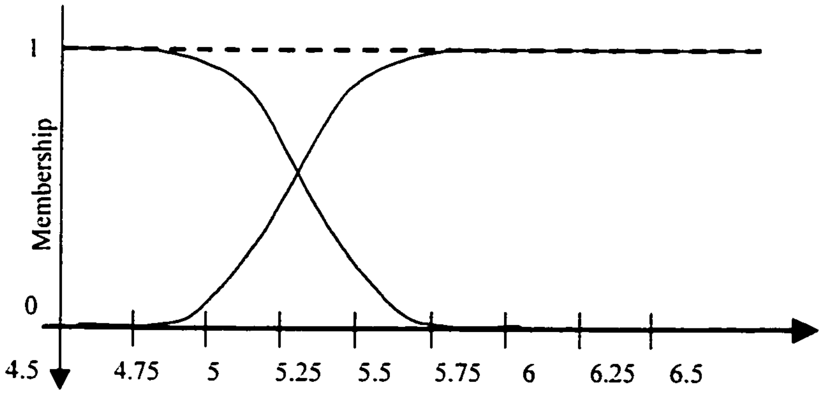
Table 5 – 6 : Union of Fuzzy Set A and B

Name	**A = Model Height**	**B = Complement of A**	**A ∪ B**
Angela	0	1	1
Anne	0.2	0.8	0.8
Bobbie	0.4	0.6	0.6
Crystal	0.6	0.4	0.6
Diana	0.8	0.2	0.8
Gloria	0.8	0.2	0.8
Jane	1	0	1
Jessie	1	0	1
Karen	1	0	1
Susan	1	0	1

excluded middle is very easy to contradict with the standard fuzzy union operation. As in the previous Table 5 – 6, a model with part membership in A, such as 0.6 and a complement of 0.4, has the standard fuzzy union of 0.6. This result shows that x is not a member of X with full membership, and therefore breaks the law of the excluded middle. The standard fuzzy operation is illustrated in Graph 5 – 5. The shaded region

$$(A \cup B)(x) = \max\,[A(x), B(x)] \text{ .... Equation 5 – 10}$$

represents the standard union operation. When comparing the standard fuzzy union operation to the union of crisp sets, they both possess the maximum operation. This is where the similarities end. The area under the curve in Graph 5 – 5 shows all membership of the union operation. This area classifies all potential models and their complements with membership functions no less than 0.6. Another example of the union operation is to use the same example, but disregard the complement and use a new fuzzy set. This set is abbreviated and considers model beauty. This set is listed in



Graph 5 – 5 : Union of Fuzzy Sets A and B

Table 5 – 7. For this example X is a set of the models, and A and B denote fuzzy sets of those models in X that exhibit substantial height or beauty.

Table 5 – 7 : Union of Modified Fuzzy Set A and Fuzzy Set B

Name	**A = Model Height**	**B = Model beauty**	**A $\cap$ B**
Angela	0	0.5	0.5
Anne	0.2	0.7	0.7
Bobbie	0.4	0.3	0.4
Crystal	0.6	0.7	0.7
Gloria	0.8	0.4	0.8
Jane	1	0.5	1

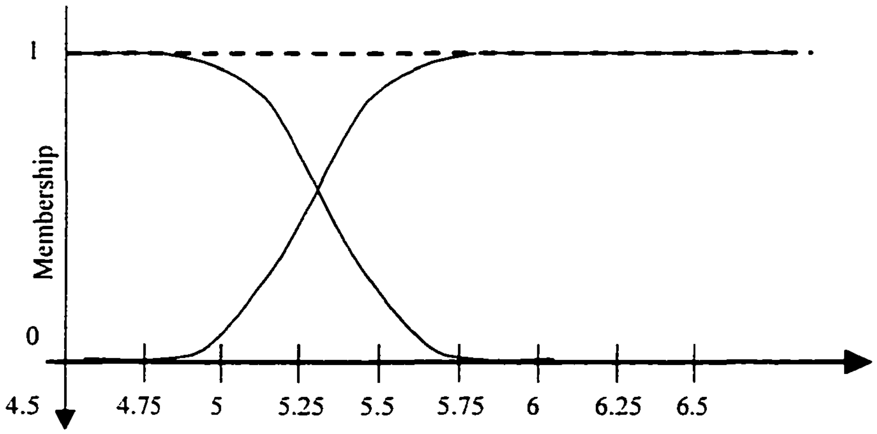
The last fuzzy set operation is the standard fuzzy intersection. This operation needs two sets just as the standard fuzzy union. The two fuzzy sets A and B are always considered to be defined in the universal set X. Intersection uses min, an abbreviation for the minimum operator, to achieve the intersection function. This function is shown in Equation 5 – 11, and is also used in the model example shown in Table 5 – 8. The law of contradiction for crisp sets is a very useful tool in determining if an intersection is valid. If a crisp set violated the law of contradiction, this would result in an invalid set type or definition. The law of contradiction for crisp sets is shown in

$(A \cap B)(x) = \min [A(x). B(x)]$..Equation 5 – 11

$(A \cap \overline{A}) = \varnothing$..Equation 5 – 12

Table 5 – 8 : Intersection of Fuzzy Set A and B

Name	**A = Model Height**	**B = Complement of A**	**A $\cap$ B**
Angela	0	1	0
Anne	0.2	0.8	0.2
Bobbie	0.4	0.6	0.4
Crystal	0.6	0.4	0.4
Diana	0.8	0.2	0.2
Gloria	0.8	0.2	0.2
Jane	1	0	0
Jessie	1	0	0
Karen	1	0	0
Susan	1	0	0



Graph 5 – 6 : Intersection of Fuzzy Sets A and B

Equation 5 – 12. This law can be easily violated with the help of Table 5 – 8. For example, consider the potential model Crystal, who is associated with a membership of A(x) = 0.6 and B(x) = 0.4. Substituting these numbers into Equation 5 – 13 the result is

$$(A \cap B)(x) = \min\,[0.6, 0.4] = 0.4 \quad \text{............Equation 5 – 13}$$

not zeroed or a null set. Graph 5 – 5 illustrates the standard fuzzy intersection function using the potential model example.

Up into now all the concentration has been on fuzzy sets. Subordinate and sometimes overlooked topic is fuzzy subsets. To give some background, any fuzzy set A defined on a finite universal set X, has a scalar cardinality shown in Equation 5 – 14.

$$|A| = \sum_{x \in X} A(x) \quad \text{............Equation 5 – 14}$$

Data from the potential model example (Table 5 – 8), is inserted into Equation 5 – 14 and yields the result in Equation 5 –15. Considering the former, any pair of fuzzy subsets defined on a finite universal set X, the degree of subsethood is found in

$$|A| = 0.2 + 0.4 + 0.6 + 0.8 + 0.8 + 4 = 6.8 \text{..........................Equation 5 – 15}$$

$$S(A,B) = \frac{1}{|A|}\left(|A| - \sum_{x \in X} \max[0, A(x) - B(x)]\right) \text{..................Equation 5 – 16}$$

Equation 5 – 16. It must also be noted that the |A| is known as the sigma count of A and the Σ in Equation 5 – 16 denotes the sum of the degrees to which the subset inequality A(x) ≤ B(x) is violated.[29]

CHAPTER VI

ADVANCED PROPERTIES OF FUZZY SET THEORY

Just like crisp sets. fuzzy sets have properties and operations in addition to than complement. union. and intersection. These other properties and operations are explored in detail in this chapter. Also. as seen from Chapter IV. fuzzy sets do not conform to the same rules as do crisp sets. There was a mirrored similarity in chapter 4. but in this chapter other such similarities can be made. The main topics of this chapter are α - cuts. the extension principle. t – norms. and t – conorms. Briefly. the α - cuts allow a horizontal action to be applied to fuzzy sets. The extension principle takes two fuzzy sets with a common factor and maps them into one set. And lastly. t – norms and t – co-norms are widely used in multi-valued logic and are used in fuzzy logic to reclassify the standard fuzzy union and intersection The remainder of this chapter shall be devoted to an in-depth discussion and development of the former subjects.

The principle of α - cuts plays a very important role in the relationship between crisp sets and fuzzy sets. This principle relies on a unique aspect of an expert. The expert factor is very common in everyday life. In every field of research. trade. or profession. the sense of an expert conveys a sense of experience, knowledge, and foresight. An expert is a person that is looked upon as achieving the pinnacle of knowledge in a certain field in which he or she is associated. To capture the knowledge of an expert, a type of computer system was developed called an expert system. Expert systems are management tools that aid in the decision making process. By accumulating the knowledge of one or many experts, these computer programs have the ability to provide advice or make recommendations to users. There are two types of expert systems that will be discussed: rule-based and case-based. Although the systems have been criticized for having no "common sense," they perform well in areas that require years of education and training for the human mind. [6]

The acceptance and use of expert systems in the commercial market has been established in three different waves. The first wave of use of expert systems began in the early 1980's. In this stage of development of expert systems, companies were not looking for these systems to become integral parts of their business. Rather, these early designers and users of expert systems were interested primarily in the research and development aspects of these new systems. The first commercial expert system, Digital Equipment Corporation's XCON, was put into use in 1981. Although it did not make a major impact on the market, it established an introduction into a new aspect of computing. The research and development completed in the early 1980's laid the groundwork for expert systems. By 1983, such expert systems building tools as VAX, OPS5, and Expert Ease were available for sale. This leads into the second major wave of expert systems adaptation. Using the expert systems builders now available on the market, some leading edge companies moved to take advantage of this new technology. Through 1985 and 1986, expert systems were in great demand as companies used their available resources to try to create an advantage for themselves. However, the systems and technology were not mature enough to make a major impact right away. Some of these early adopting companies did contribute a great deal to the maturation of expert systems. The successful early adopters of expert systems created a valuable advantage for themselves in moving toward the increase in sophisticated use of expert systems in the 1990's. The 1990's begin the third wave of acceptance for expert systems. By this time, a majority of United States companies have become interested in the adaptation of expert systems. The technology has matured, many problems have been solved by research and development teams and the prices of the programs have come down. This leads us into the present state of expert systems. Companies are building on and increasing the use of expert systems, leading the market onward towards the 21st century.

One of the goals of an expert system is for the computer to be able to make decisions without the help of human beings. In order to do this, knowledge must be represented in a way that the computer can understand and use it. Fuzzy logic's main purpose is to bridge the gap between human reasoning and computer programming. Fuzzy logic allows a computer to make decisions based on a rule set without the aid or dependence of human input. But, knowledge involves relationships between things. It is very difficult to grasp because people bring knowledge to the data through analysis. And, expert systems try to find a way for the computer to perform this analysis.

There are three different ways to express knowledge. The first way is to follow a specific set of steps to achieve a particular result. This is known as procedural knowledge and is the basis for rule-based systems. This kind of system is used almost exclusively because it is easy to understand. Knowledge is represented as statements and rules. most commonly as if-then statements. There are two types of rules that are used. definitional and heuristic. Definitional rules show the relationship between terms and heuristic rules are used when there is incomplete evidence. No program could be developed to solve all possible solutions. but a system that uses heuristic rules can make reasonable guesses based on these rules and therefore, is generally more beneficial. In addition to these two types of rules. there are two ways to group these rules. The simple way is to group all the rules together in one set and examine them all at once. The complex way is to divide the rules into subsets and examine them according to some search strategy. The variety chosen depends on what is trying to be accomplished. A second type of knowledge is declarative. This type knows that two or more terms are logically related. And the third type of knowledge uses hunches and rules of thumb learned from experience. This kind of knowledge is the hardest to represent to the computer because it doesn't have the experience.

In addition to rule-based systems, induction systems and hybrid systems can be used to represent knowledge.

Induction systems represent knowledge in a table of attributes and values and then use an induction algorithm to convert the knowledge to a decision tree. Semantic nets can be used to describe knowledge in this type of system. They consist of arcs and nodes that are usually used to describe and illustrate larger relationships.

Hybrid systems are the most complex. They combine object-oriented programming with rules. Knowledge in this context is represented in frames. a combination of descriptive and operational knowledge. This type of system is used when the description of knowledge is best thought of in terms of diagrams and models. but this form is rarely used because of its complexity and incomprehensibility to users.

One of the primary means of constructing an expert system is through the use of rules. The basis of a rule-based system is to assemble a knowledge base of rules from which the system can make a decision. As mentioned earlier. rules are composed of an if-condition-then action format. The if clause establishes a value or variable for which the information is found to exhibit. The then action section prescribes an appropriate action that should be taken. This type of expert system is very similar to some concepts in fuzzy logic. That is why the area of fuzzy expert systems have grown exponentially.

Case-based expert systems. as might be guessed. make decisions on the use of a warehouse of cases. The idea behind this system is to mirror human analogical reasoning. This process involves viewing the current situation and using your knowledge of previous experiences (cases) to find similarities. The solution or prescribed action from the previous case(s) can then be accepted or modified to the current situation. A comparison between rule-based and case-base systems is shown in Table 6-1.[21]

The concept of α - cuts will now be resumed before the digression into the realm of expert system goes any further. But it must be noted that expert systems were in wide-spread acceptance before fuzzy logic was considered a worthwhile field of study.

Table 6 – 1 : Comparison between Rule-Based and Case-Based Expert Systems

Rule-Based	**Case-Based**
Limited learning capabilities	Learning is inherent in the architecture
Reasons using IF-THEN rules	Reasons using situation-specific cases
Knowledge acquisition is time intensive	Knowledge acquisition is less complicated
Time consuming to build and maintain because knowledge is dynamic	Easier to build and maintain because cases already exist
Difficulty with problems outside original scope	Can solve problems outside original scope
Adding knowledge is complex and error-prone	Adding knowledge is adding a case
Ideal for knowledge-rich domains	Ideal for experience-rich domains

And recently, fuzzy logic has propelled the concept of expert systems to the next level with fuzzy expert systems.

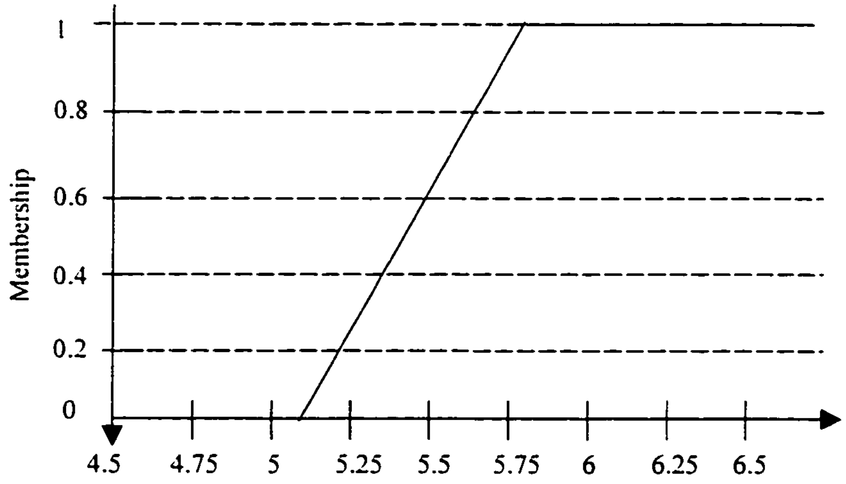
As stated before, the α - cut relies on an expert to interpret the horizontal portion of the fuzzy curve. Up to now only the vertical portion of the fuzzy set has been manipulated. There has only been concern with the degree of membership of the x in A (x) confined in the universal set X. Going back to the model example, Table 6 – 2 is an abbreviated list of tall potential models, their height, and their degree of membership. Further illustrating this example, Graph 6 – 1 plots the model heights versus the degree of membership. The corresponding membership function is depicted in Equation 6 – 1 and Equation 6 – 2. This is the point where fuzzy sets are linked to crisp sets. As an

Table 6 - 2 : Set of All Potential Models Mapped to a Fuzzy Set

Name	Height	Degree of Membership
Angela	4 feet 8 inches	0
Anne	5 feet 2 inches	0.2
Bobbie	5 feet 5 inches	0.5
Crystal	5 feet 6 inches	0.6
Diana	5 feet 7 inches	0.8
Jessie	5 feet 11 inches	1
Karen	6 feet 1 inches	1

A (x) = 0/4-8 + 0.2/5-2 + 0.6/5-6 + 0.8/5-7 + 1/5-11 + 1/6-1.........Equation 6 -1

A (x) = $0/x_1 + 0.2/\ x_2 + 0.6/\ x_3 + 0.8/\ x_4 + 1/\ x_5 + 1/\ x_6$...............Equation 6 -2



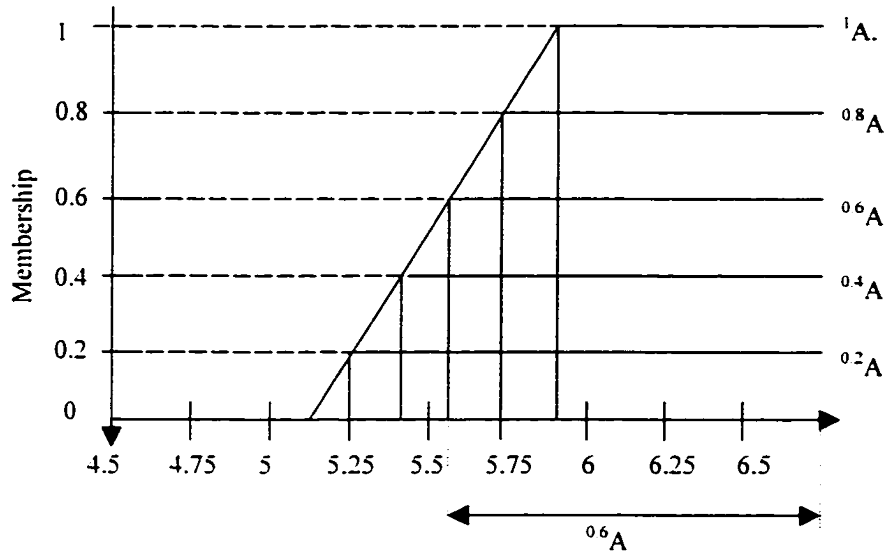
Graph 6 – 1 : Potential Models Fuzzy Set with Horizontal Membership Lines example, the closed interval $[x_2, x_3]$ maps to [0.2, 0.6] in the fuzzy set. These can be classified as subsets that consists of all elements of X whose membership degrees in the fuzzy set are restricted to some given crisp subset of [0, 1]. This restriction applied to A is known as a crisp subset $^{\alpha}A$ of the universal set X. This restriction is formally known as an alpha cut of A, which is defined in Equation 6 – 3. It must be noted that this

$$^{\alpha}A = \{x \in X \mid A(x) \geq \alpha\} \text{...Equation 6 - 3}$$

equation is valid for any $\alpha \in [0, 1]$. The alpha cut makes a distinction between various crisp subsets by the choice of an expert. There are three different properties of a fuzzy set that determine the property of the alpha cut. These properties are support, core and level. The support of a fuzzy set is shown in Equation 6 – 4. The support region of a fuzzy set is the non-zero membership in A of all elements in the universal set X. In other words the support area is any data point in A that has a greater than zero membership. In the model example this would be all potential models over 5 feet 1 inches. The next property is called the core area. The core area of a fuzzy set is shown in Equation 6 – 5. This area represents all the data in A that has a full membership in A

$$\text{supp}(A) = {}^{0+}A = \{x \in X \mid A(x) \geq 0\} \text{....................................Equation 6 – 4}$$

$$\text{core}(A) = {}^{1}A = \{x \in X \mid A(x) \geq 1\} = \{x \in X \mid A(x) = 1\} \text{...........Equation 6 – 5}$$



Graph 6 – 2 : Potential Models Fuzzy Set with Horizontal Membership Lines

defined on a universal set X. In the potential model example this would be all the women greater than or equal to the height of 5 feet 8 inches. The last property is the concept of a level set. The level set is formally defined in Equation 6 – 6. A level set is a set of numbers which represents all distinct α - cuts of A. A level set is a very important concept because this portion of the graph contains all of the alpha cuts. Considering the example of the potential model the level set would be all women with

$L(A) = \{\alpha \in [0, 1] \mid \{A(x) = \alpha \text{ for some } x \in X\}$ ....Equation 6 – 6

heights between 5 feet 1 inch and 5 feet 8 inch. All alpha cuts are contained in this region.

Another important fact about alpha cuts is that as the value of the alpha gets smaller the region represented gets larger and as the value of the alpha gets larger the region represented gets smaller. This leads to Equations 6 – 7 and 6 –8. These two equations represent a common union and intersection operation on alpha cuts or crisp subsets. Equation 6 – 7 uses the intersection operation on two fuzzy subsets to achieve the result ${}^{\alpha 2}A$. It must be noted that $\alpha 1 < \alpha 2$ and ${}^{\alpha 1}A$ is partly included in the region of

$${}^{\alpha 1}A \cap {}^{\alpha 2}A = {}^{\alpha 2}A \text{ ....Equation 6 - 7}$$

$${}^{\alpha 1}A \cup {}^{\alpha 2}A = {}^{\alpha 1}A \text{ ....Equation 6 - 8}$$

${}^{\alpha 2}A$. This can also be seen in Graph 6 – 2 where ${}^{0.6}A$ dissects the fuzzy set and includes parts of the lower alpha cuts and hence, proves Equation 6 – 7. represents the intersection of two fuzzy subsets and the smaller alpha cut is the intersection. Like intersection the union operation also has the criteria of $\alpha 1 < \alpha 2$ and ${}^{\alpha 1}A$ is partly included in the region of ${}^{\alpha 2}A$. The union of the α - cuts is performed in the same fashion as the union of crisp sets. As a result, the union operation on the alpha cuts is the exact opposite of the intersection result. For the union operation, the data from two sets become joined into one set as shown in Equation 6 – 8.

To get a better understanding of the order of alpha cuts on A, Figure 6 – 1 is presented to convey this concept. This figure shows how alpha cuts include parts of

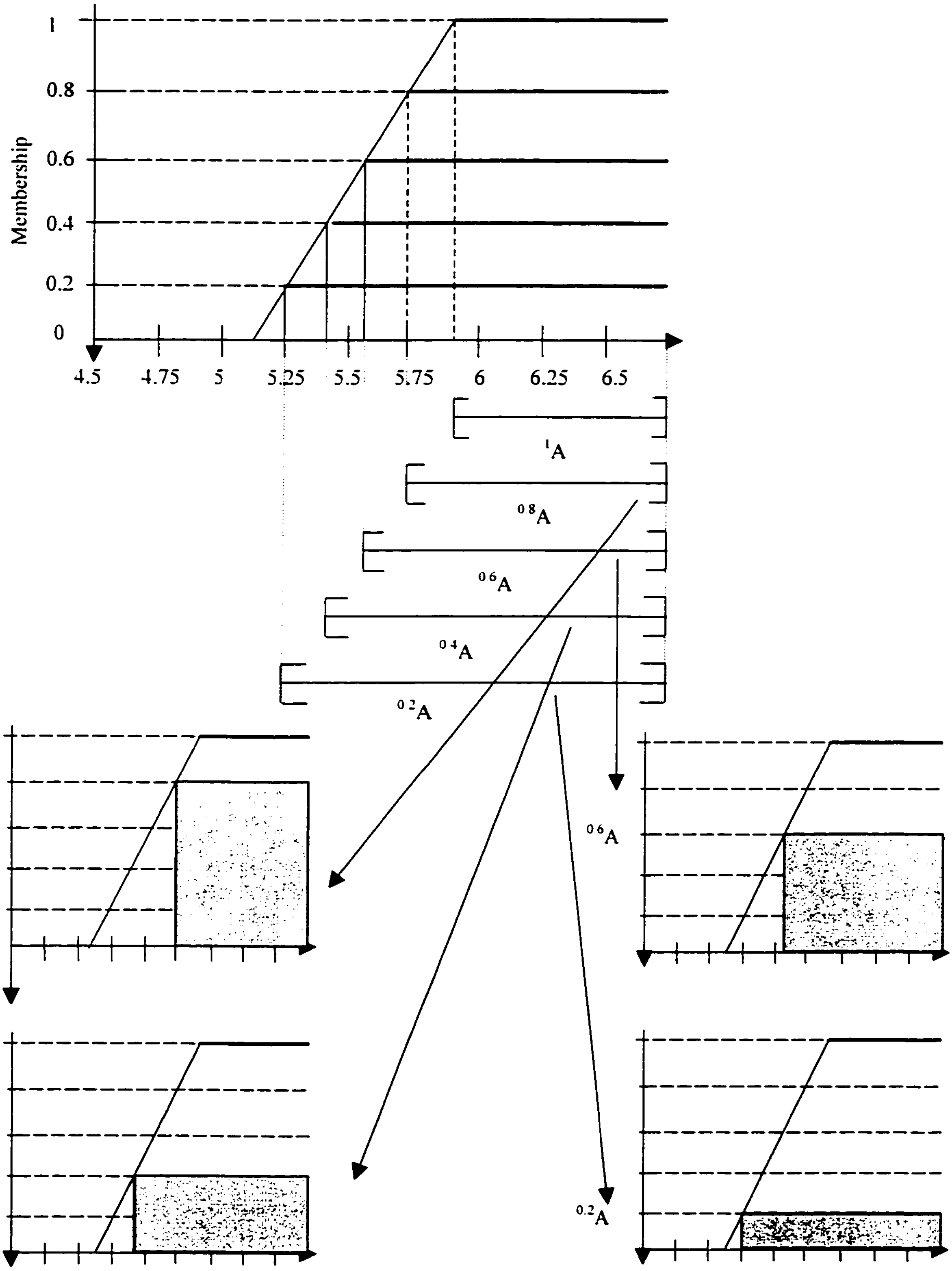


Figure 6 - 1 : Decomposition of Fuzzy Sets into Alpha Cuts

other alpha cuts. Take, for example, ${}^{0.8}A$ shown in Figure 6 - 1. This alpha cut intersects all of the other alpha cuts in A. Figure 6 - 1 also shows the decomposition of the fuzzy sets with some of the alpha cuts. To represent a one line equation for ${}^{0.8}A$, one must first represent the characteristic equation for the containing fuzzy set. Equation

$$A = 0.2/x_1 + 0.4/x_2 + 0.6/x_3 + 0.8/x_4 + 1/x_5 \text{ ....Equation 6 - 9}$$

$${}^{0.2}A = 1/x_1 + 1/x_2 + 1/x_3 + 1/x_4 + 1/x_5 \text{ ....Equation 6 - 10}$$

$${}^{0.4}A = 0/x_1 + 1/x_2 + 1/x_3 + 1/x_4 + 1/x_5$$

$${}^{0.6}A = 0/x_1 + 0/x_2 + 1/x_3 + 1/x_4 + 1/x_5$$

$${}^{0.8}A = 0/x_1 + 0/x_2 + 0/x_3 + 1/x_4 + 1/x_5$$

$${}^{1.0}A = 0/x_1 + 0/x_2 + 0/x_3 + 0/x_4 + 1/x_5$$

6 - 9 shows the characteristic equation for fuzzy set A contained in the universal set X. The next step is to show how the previous fuzzy set can be represented by its alpha cuts. The fuzzy set has five distinct alpha cuts that each has a respected characteristic equation. These characteristic equations are shown in Equation 6 - 10. Each characteristic equation has a definite membership in other fuzzy quantities. For example, the alpha cut ${}^{0.6}A$ has a definite membership of 1 in x_3, x_4, and x_5 and a membership of 0 in x_1, x_2. This means that the alpha cut ${}^{0.6}A$ contains the data represented by x_3, x_4, and x_5. From Equations 6 –7 and 6 – 8, and Figure 6 – 1, one can easily prove the union and intersection operations of alpha cuts.[7]

A term mentioned previously in this chapter is the concept of decomposition of fuzzy sets. This is an important concept because it proves that the representation of fuzzy sets and their alpha cuts are universal. This theory can be illustrated by a special case of fuzzy sets shown in Equation 6 – 11. It is now possible to convert Equation 6 – 10 with the assistance of Equation 6 – 11 to obtain a result shown in Equation 6 – 12. This result shows partial membership for different alpha cuts where before there was complete membership. This last step completes the decomposition of fuzzy set A to the special fuzzy set ${}_{\alpha}A$. It must also be noted that this definition of a special fuzzy set applies to any fuzzy set defined on a finite or infinite universal set X. Since the decomposition is complete, it would be prudent to reconstruct the original fuzzy set

$${}_{\alpha}A\,(x) = \alpha \bullet {}^{\alpha}A\,(x) \text{...Equation 6 - 11}$$

$${}_{0.2}A = 0.2/x_1 + 0.2/x_2 + 0.2/x_3 + 0.2/x_4 + 0.2/x_5 \text{ ....Equation 6 - 12}$$

$${}_{0.4}A = 0/x_1 + 0.4/x_2 + 0.4/x_3 + 0.4/x_4 + 0.4/x_5$$

$${}_{0.6}A = 0.x_1 + 0/x_2 + 0.6/x_3 + 0.6/x_4 + 0.6/x_5$$

$${}_{0.8}A = 0/x_1 + 0/x_2 + 0/x_3 + 0.8/x_4 + 0.8/x_5$$

$${}_{1.0}A = 0/x_1 + 0/x_2 + 0/x_3 + 0/x_4 + 1/x_5$$

A from the decomposed fuzzy set ${}_{\alpha}A$. This is done by the standard fuzzy union operation applied to Equation 6 – 12. It is easy to see from Equation 6 – 13 that the standard fuzzy union of the five special fuzzy sets in Equation 6 – 12 is exactly the original fuzzy set A.

$$A = {}_{0.2}A \cup {}_{0.4}A \cup {}_{0.6}A \cup {}_{0.8}A \cup {}_{1}A \text{ ....Equation 6 – 13}$$

Complement is another important operation in crisp set and fuzzy set theory. In alpha cuts, however, the concept of complementation takes on a totally different meaning. The alpha cut of a complement of A has two distinct and different definitions. The first is the alpha cut of the complement of A, denoted by ${}^{\alpha}\left(\overline{A}\right)$; the second is the complement of the alpha cut A, which is denoted by ${}^{\alpha}\overline{A}$. As shown in Equation 6 – 14, the alpha cut of the complement of the alpha cut A and the alpha cut of the complement of A are not equal. Revisiting the potential model example, an alpha cut of 0.4 is

$${}^{\alpha}\left(\overline{A}\right) \neq {}^{\alpha}\overline{A} \text{ ....Equation 6 – 14}$$

carried out on the fuzzy set A. The fuzzy set is dissected by the alpha cut, which in turn intersects the complement. At this point of intersection both the fuzzy set and its complement can be investigated. The alpha cut of the complement of A, ${}^{\alpha}\left(\overline{A}\right)$, is extruded from the fuzzy set to the orgin in Figure 6 – 2. There is a closed interval and an open interval representing this section . The other complement of the alpha cut

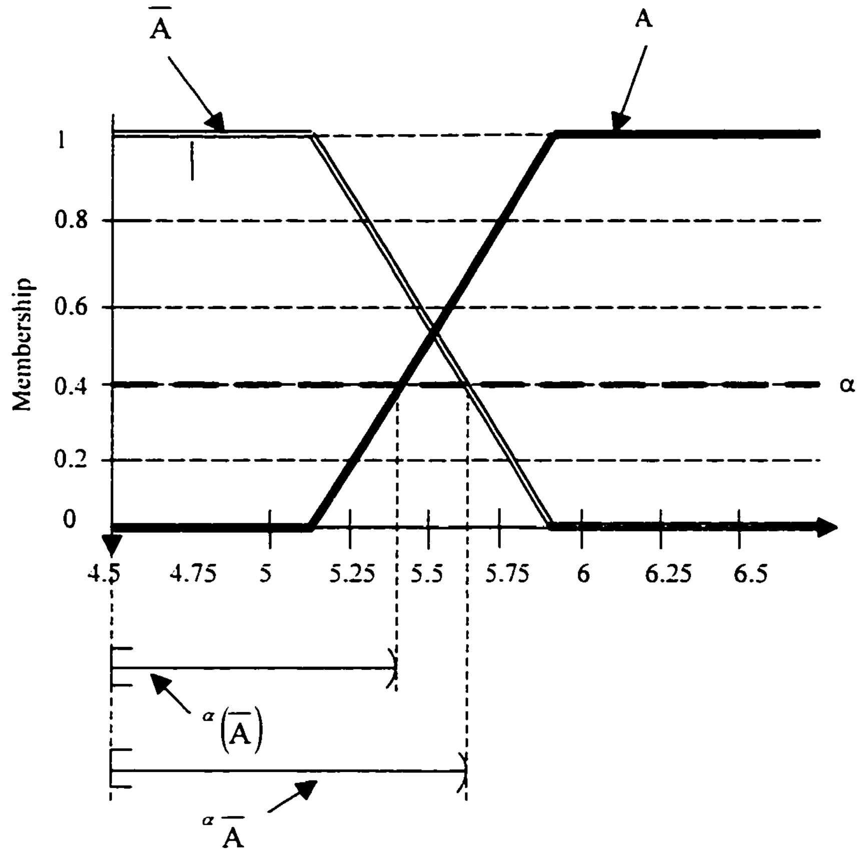


Figure 6 – 2 : Two Different Complements of an Alpha Cut of Fuzzy Set A

A, $^{\alpha}\overline{A}$, is sightly larger than the previous complement $^{\alpha}(\overline{A})$, but it depends on the orginal fuzzy set. However the two complements will never be equal no matter the fuzzy set.

These previous operations concerning alpha cuts on fuzzy sets are just the basic operations. Listed in Equations 6 – 15, 6 – 16, and 6 – 17 are other fuzzy identities with fuzzy sets A and B dissected by alpha cuts. These identities cover the addition, subtraction, and multiplication on alpha cuts and also prove that alpha cuts react the same way to distributivity as crisp sets.

$(A + B)\,\alpha = A\alpha + B\alpha$..Equation 6 - 15

$(A - B)\,\alpha = A\alpha - B\alpha$...Equation 6 – 16

$(A \bullet B)\,\alpha = A\alpha \bullet B\alpha$..Equation 6 – 17

Alpha cuts can be characterized as one of the most important concepts of fuzzy sets, and strong alpha cuts are no exception. The strong alpha cut is very similar to the regular alpha cut in definition. The only difference is that for the strong alpha cut the membership grades in the given set are greater than the specified value of alpha. In regular alpha cuts the membership grade is greater than or equal to the specified value of alpha. The strong alpha cut is shown formally in Equation 6 – 18. Also it must be

${}^{\alpha+}A = \{x \in X \mid A(x) > \alpha\}$..Equation 6 – 18

noted that when expressing the strong alpha cut, it is always included in the alpha cut of any fuzzy set and for any $\alpha \in [0, 1]$ contained in the universal set X. This statement is proven by the previous Equations 6 – 3 and 6 – 18. This concept is shown formally in Equation 6 – 19. Other identities that represents the strong alpha cut on fuzzy sets are shown in Equations 6 – 20, 6 – 21, and 6 - 22. These identities can be

${}^{\alpha+}A \in {}^{\alpha}A$ ....Equation 6 - 19

extended to include regular alpha cuts. To clarify a point in some of the identities, it is assumed that α1 > α2 and they both are defined in any fuzzy set and for any $\alpha \in [0, 1]$ contained in the universal set X.

${}^{\alpha 1+}A \cap {}^{\alpha 2+}A = {}^{\alpha 2+}A$..Equation 6 - 20

${}^{\alpha 1+}A \cup {}^{\alpha 2+}A = {}^{\alpha 1+}A$..Equation 6 - 21

${}^{\alpha+}(A \cup B) = {}^{\alpha+}A \cup {}^{\alpha+}B$ ....Equation 6 -22

${}^{\alpha+}(A \cap B) = {}^{\alpha+}A \cap {}^{\alpha+}B$ ....Equation 6 -23

${}^{\alpha}(\overline{A}) = {}^{(1-\alpha)+}(\overline{A})$..Equation 6 -24

${}^{\alpha+}(\overline{A}) \neq {}^{\alpha+}\overline{A}$ ....Equation 6 -25

The next topic of advanced property of fuzzy sets is known as the extension principle. The extension principle is an operation for fuzzifying crisp functions. This is an evolution of this dissertation because formerly this paper has examined similarities and differences between crisp sets and fuzzy sets. This led to the development of computation with fuzzy sets where there is a need to find a way to take traditional crisp set functions and fuzzify said function.

Consider the basic definition of a function shown in Equation 6 - 25. A function maps a set of data to another set of data. An equation may be as simple as a negation or complex as an nth degree polynomial. But the operation of the function does not

$f : X \rightarrow Y$ ....Equation 6 –25

change in any instance. The equation for the fuzzifying crisp set is very similar to the generic function equation . The fuzzifying equation is shown in Equation 6 – 26. This equation takes data from fuzzy set X and transposes them to fuzzy set Y. The inverse of

$f : \mathcal{F}(X) \rightarrow \mathcal{F}(Y)$ ....Equation 6 –26

$f^{-1} : \mathcal{F}(Y) \rightarrow \mathcal{F}(X)$ ....Equation 6 –27

this equation is found in Equation 6 – 27. To further explain this concept, the potential model example shall be called upon. The data set shall be modified to assure true representation of the extension principle in its entirety. The data set that has been developed throughout the previous chapters is height versus potential models. Each data point corresponds to a discrete case, in other words a young woman. Assume that some of the potential models met the height requirement and became professional models. There is another concern of the modeling agency, model pay. Now that the potential models are placed in a data set of all professional models, their pay should be different than an experienced model. Each photo shoot pays the professional models salaries ranging from 500 dollars to 4000 dollars. There is also an assumption in this example that the model with the most experience is very beautiful and highly sought after by advertising companies. Also, the modeling agency has very high standards and does not allow a model to stay with the agency if her popularity rating does not continue to increase with time. To extend this example, one could consider the experience of the

Table 6 – 3 : Model Experience and Model Pay

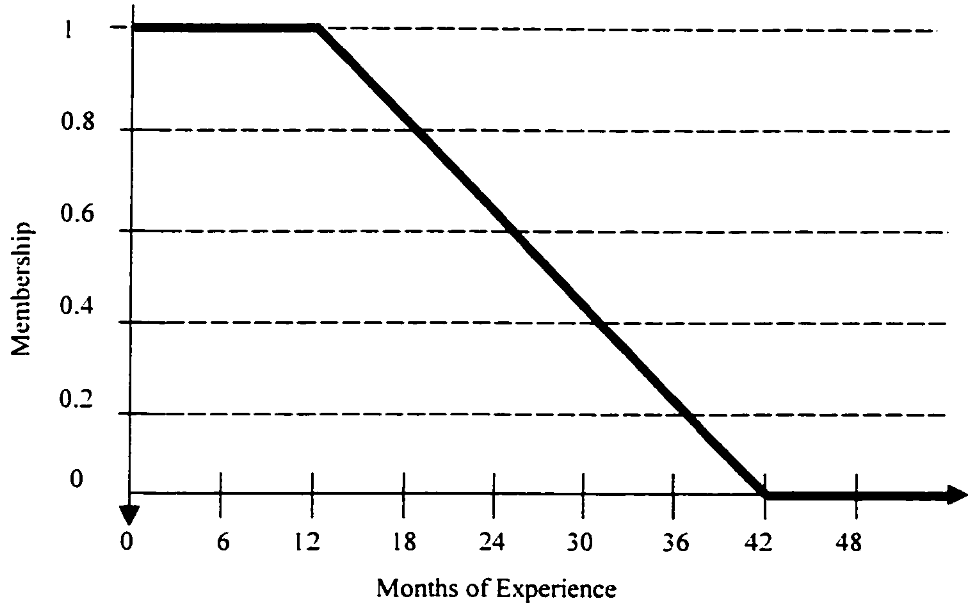
Months of Experience	**Salary per Photo Shoot**
0	500
6	1000
12	1500
18	2000
24	2500
30	3000
36	3500
42	4000

professional model in months and her corresponding salary. The data of the professional model's experience and salary are shown in Table 6 – 3. In the table one side does not necessarily correspond to the other. For example. that if a professional model has 12 months experience. her salary per photo shoot may not be 1500 dollars. The table is not reversible from salary to experience either. The reason for this is that a professional model salary may be, for example. 1500 dollars and the corresponding level of experience is 12 to 18 months. But. if the inverse of the function was considered. correlating experience to salary, the 12 month experienced model would be correlated with the 1500 dollar salary. The reason for this is that a function has one real answer only and hence one of the experience levels will be left out. Also. designating the young professional model's salary is another problem for the modeling agency. It must be noted that experience and not age of the model corresponds to young in this case.

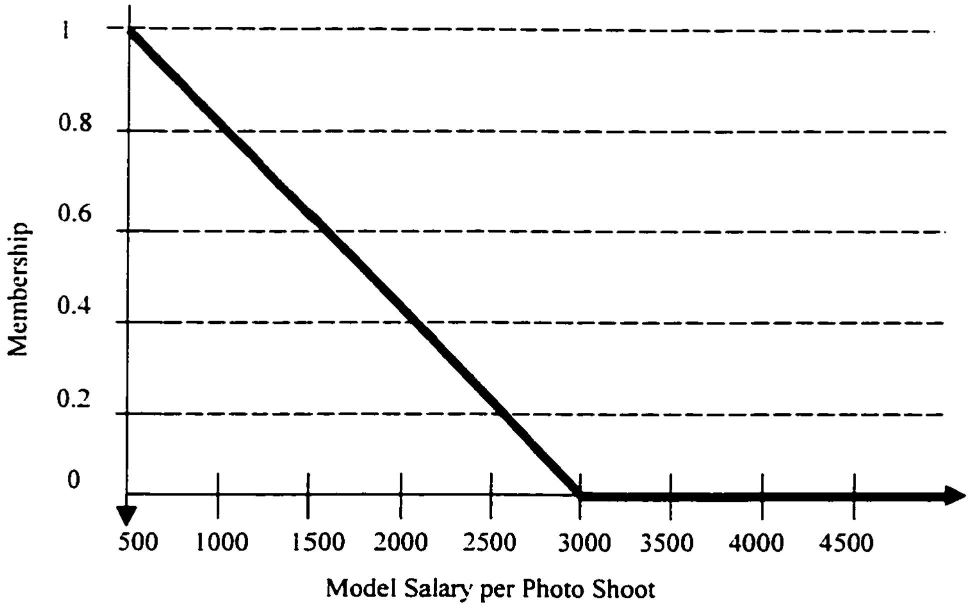
To contend with the previous problem. the extension principle is used. But first the data sets found in Table 6 – 3 have to be fuzzified. This can be done by a set of criteria found listed below:

1. Models start to gain experience around one year.
2. Models are very experienced when they have worked three and a half years.
3. Young models never get over 3000 dollars per photo shoot.
4. All young models start at 500 dollars and work their way up in pay.

From this criteria two graphs can be constructed. One graph shows the concept of inexperience of a professional model. The other graph focuses on the pay per photo



Graph 6 – 3 : Fuzzy Set of Inexperienced Professional Models



Graph 6 – 4 : Fuzzy Set of Inexperienced Professional Models Salary per Photo Shoot

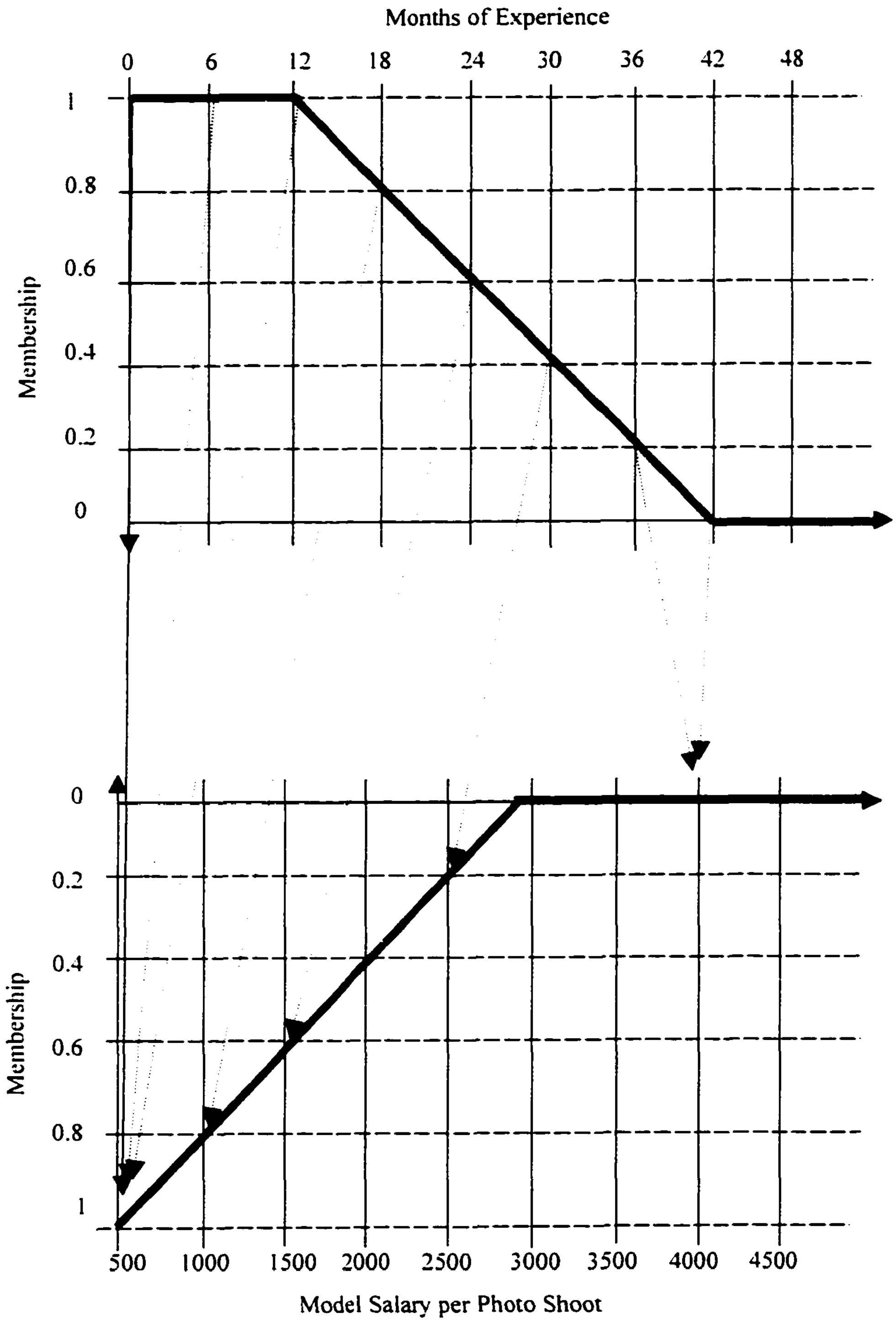


Figure 6 – 3 : Extension Principle Applied to Salary and Experience Fuzzy Sets

shoot. These two concepts are shown in Graphs 6 – 3 and 6 – 4. Instead of using experience and pay related to experience as the deciding factor, the opposite is used to show the flexibility of fuzzy sets. Also, the graphical data is shown is a one-line function in Equation 6 – 29. This was done after modifying Equation 6 – 25 and solving for the B in Equation 6 - 28. The notations of A and B are referred to the two fuzzy sets considered in this example, months of inexperience and pay per shoot, respectively.

$A(x) = B \bullet f(x)$ ....Equation 6 - 28

$B = 1/f(0) + 1/f(6) + 1/f(12) + 0.8/f(18) + 0.6/f(24) +$..........Equation 6 – 29

$0.4/f(30) + 0.2/f(36) + 0/f(42) = 1/500 + 0.8/1000 +$

$0.6/1500 + 0.4/2000 + 0.2/2500 + 0/3000 + 0/3500 +$

$0/4000$

The former two graphs are combined and mapped together to form Figure 6 – 3. This figure shows how the extension principle is used to map a fuzzy set or to connect one fuzzy set to another. These sets had a common factor involving the inexperience of a professional model, but this is not a prerequisite for the extension principle.[29]

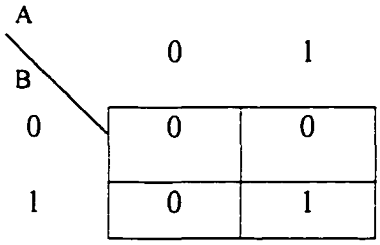
The former example is of a discrete nature, but not all fuzzy sets are discrete. Equation 6 – 30 shows a continuous function defined on the set of real numbers. The

$$[\tilde{f}(A)](y) = \sup_{x\;y=f(x)} A(x) \text{ .... Equation 6 - 30}$$

sup in the equation is known as supremum or least upper bound. The supremun is defined if X is a subset of the set $\mathbb{R}$ of real numbers. If there is a real number u such that x is greater than u for every x in X. then u is known as the least upper bound.

The last topics of this chapter are t – norms and t – conorms of fuzzy sets. which are considered the basic connectives for fuzzy logic. These are done by modeling the logical connector "and" and "or." The logical connector "and" is known as the t – norm: the "or" logical connector is considered to be the t – conorm. These two operations represent the intersection and union of fuzzy sets in logical terms. respectively. This section will introduce the concept of basic connectives used in combining knowledge that comes from fuzzy sets defined in the universal set X.

The t - norm is the intersection of two fuzzy sets A and B specified in general by a binary operation. If A and B are ordinary subsets of the universal set X. then there can be a truth evaluation of "A and B" in terms of the possible truth values 0 and 1 of A and B. Figure 6 – 4 shows the binary "and" equivalent for the t – norm or intersection action. Taking Figure 6 – 4 in consideration and adding any function *i*, one can write Equation 6 – 31 as the binary operation on the unit interval. To transform



A / B	0	1
0	0	0
1	0	1

Figure 6 – 4 : Binary Operation "And" Connector

$i : [0, 1] \times [0, 1] \rightarrow [0, 1]$ .... Equation 6 - 31

this definition from a binary operation to a set operation, the intersection operation used in crisp sets is utilized. Equation 6 – 31 considers a pair of sets consisting of the element's membership grades in sets A and B and yields the membership grade of the element in the set constituting the intersection of A and B. The function *i* in Equation 6 – 31 can be any function that qualifies as a fuzzy intersection. For *i* to qualify as a

$(A \cap B)(x) = i\,[A(x), B(x)]$ .... Equation 6 – 31

fuzzy intersection, necessary and sufficient, Axioms 6 – 1 through 6 – 7 must be satisfied. These axioms are defined on the unit interval with the common assumption that a, b, d $\in [0, 1]$.

$i\,(a, 1) = a$ .... Axiom 6 - 1

$b \leq d$ implies $i\,(a, b) \leq i\,(a, d)$ .... Axiom 6 - 2

$i\,(a, b) = i\,(b, a)$ .... Axiom 6 - 3

$i\,(a, i(b, d)) = i(i(a, b), d)$ .... Axiom 6 - 4

i is a continuous function.. Axiom 6 - 5

i (a, a) < a..Axiom 6 - 6

$a_1 < a_2$ and $b_1 < b_2$ implies . $i (a_1 < b_1) < i (a_2 < b_2)$.................... Axiom 6 - 7

The t - conorm is the union of two fuzzy sets A and B specified in general by a binary operation. This action is very similar to the t - norm or intersection operation. If A and B are ordinary subsets of the universal set X, then there can be a truth evaluation of "A or B" in terms of the possible truth values 0 and 1 of A and B. Equation 6 – 32 shows the binary "And" equivalent for the t – conorm or union action. To transform

$u : [0, 1] \times [0, 1] \rightarrow [0, 1]$..Equation 6 - 33

this definition from a binary operation to a set operation, the union operation used in crisp sets is utilized just like the intersection operation is used in t - norms. Equation 6 – 33 considers a pair of sets consisting of the element's membership grades in sets A and B and yields the membership grade of the element in the set constituting the union of A and B. The function u in Equation 6 – 31 can be any function that qualifies as a fuzzy union. For u to qualify as a fuzzy union, as a minimum, Axioms 6 – 8 through

$(A \cup B)(x) = u [A(x), B(x)]$.. Equation 6 – 34

6 – 14 must be satisfied. These axioms are also defined on the unit interval with the common assumption that a, b, d $\in[0, 1]$.[15]

u (a, 0) = a ....Axiom 6 - 8

b $\leq$ d implies u (a, b) $\leq$ u (a, d)...Axiom 6 - 9

u (a, b) = u (b, a)... Axiom 6 - 10

u (a, u(b, d)) = u(u(a, b), d)... Axiom 6 - 11

u is a continuous function... Axiom 6 - 12

u (a, a) > a... Axiom 6 - 13

$a_1 < a_2$ and $b_1 < b_2$ implies u ($a_1 < b_1$) < u ($a_2 < b_2$).................... Axiom 6 - 14

CHAPTER VII

FUZZY MATHEMATICS

The basics of conventional mathematics are the manipulation of numbers. This manipulation may be addition. subtraction. multiplication. or division of these numbers. Fuzzy mathematics is very similar to conventional mathematics in this area. But where numbers are manipulated in conventional mathematics. fuzzy numbers are manipulated in fuzzy mathematics. This is where the similarity breaks down between the two types of mathematics.

It is apparent how to represent a number in conventional mathematics. but it is not that apparent in how to define a fuzzy number. As the concept of fuzzy logic is rooted in vagueness. so is fuzzy numbers. Fuzzy numbers are not that uncommon to a person in every day life. At one time or another people say. it's about 2 o'clock. I have donated approximately 2.000 dollars to charity. or she's about 6 feet tall. Fuzzy numbers have attributes of conventional numbers but with the addition of a linguistic modifier. These modifiers are very important to the concept of fuzzy numbers. A linguistic modifier can make a big difference in the range of a fuzzy number. For example. one can say. she is almost 6 feet tall or she is about 6 feet tall. The modifier "almost" relays the concept of very close to a specific height. The modifier "about" dredges up the feelings of someone being 3 to 4 inches from the desired height. The linguist modifier allows many fuzzy numbers with the same specific underlying numerical value to have drastically different fuzzy values. The concept of a fuzzy number is formally shown in Equation 7 – 1. This equation states that a fuzzy number A is expressed by a membership function. Different types of fuzzy numbers are shown in Figures 7 – 1 through 7 – 3. The first fuzzy number (Figure 7 – 1) has a triangular shape. This fuzzy number is found extensively in control applications where fuzzy logic is utilized. Also, this type of control theory is known as fuzzy control and will be discussed in coming chapters. This fuzzy number is define by four points on the x-axis, a, b, c, and d. These points represent real numbers, and f(x) and g(x) stand for functions defined in a closed interval. It can be gathered from the presentation of this fuzzy number that Equation 7 – 1 does not represent all fuzzy numbers but it does convey the

A: ˙ → [0. 1]...Equation 7 – 1

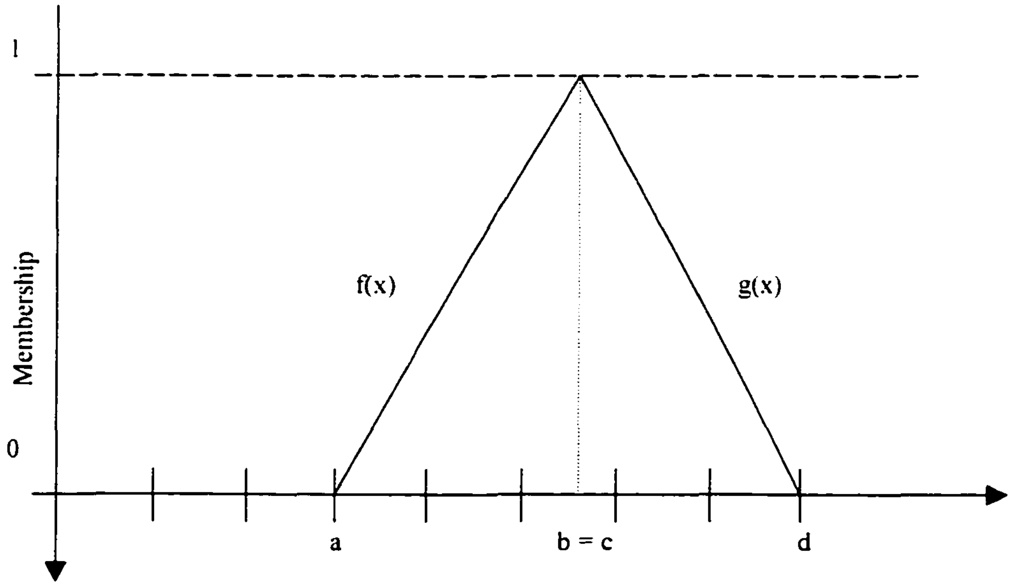


Figure 7 –1 : Triangular Fuzzy Number

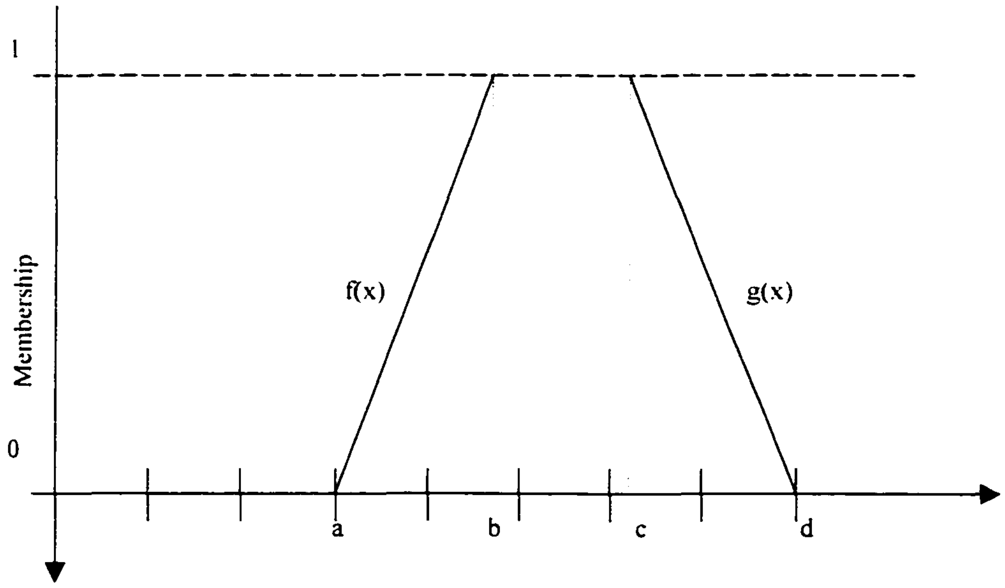


Figure 7 –2 : Trapezoidal Fuzzy Number

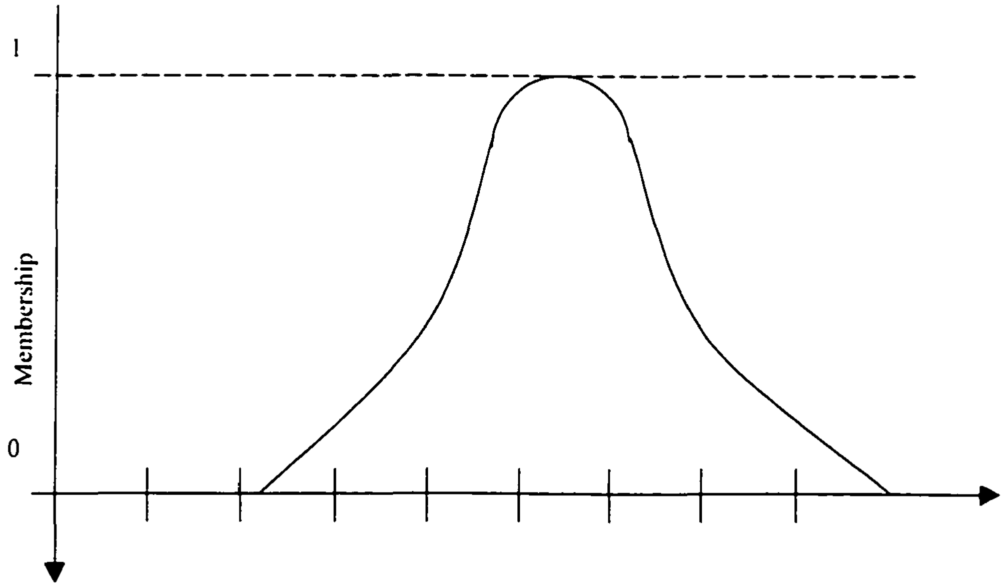


Figure 7 –3 : Bell Shaped Fuzzy Number

concept of the membership around a real number. The real number that is the focus of this fuzzy number is b and c. It must be noted that b and c are the same point, hence the same real number. To define this type and other types of fuzzy numbers, interval analysis must be applied. The concept of interval analysis of a function is an old and proven area in mathematics. A membership function that conforms to the conception of a set of numbers that are around a given real number or an interval of real numbers is shown in Equation 7 – 2. The functions f(x) and g(x) in Equation 7 – 2 are

$$A(x) = \begin{cases} f(x) & \text{for } x \in [a, b] \\ 1 & \text{for } x \in [b, c] \\ g(x) & \text{for } x \in [c, d] \\ 0 & \text{for } x < a \text{ and } x > d \end{cases} \qquad \text{.....Equation } 7-2$$

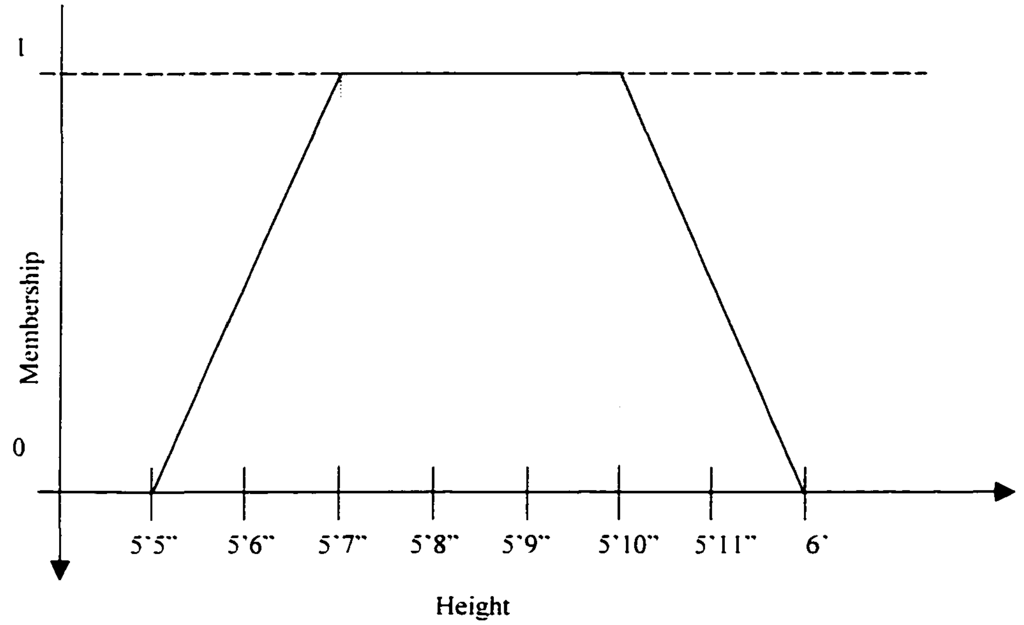
continuous which increase to 1 at point b and decrease from 1 at point c, respectively. The second fuzzy number is of a trapezoidal shape and is shown in Figure 7 – 2. This fuzzy number is sometimes called a fuzzy interval. The two names are frequently used interchangeably in various fuzzy logic texts. This dissertation shall take the standpoint of classifying fuzzy numbers and fuzzy intervals as fuzzy numbers. The trapezoidal fuzzy number is defined by four distinct points a, b, c, and d. The two points b and c display an interval within an interval. This flat plateau has full and continuous membership between b and c. The interval defines a place where this fuzzy number is in fact a set of real numbers.[27]

The model example will be revisited to further explain this concept. The idea of fuzzy numbers allows this example to approach real world conditions because, when picking a group of people or beautiful women out of a larger an all inclusive group, some criteria must be established. Height is one criterion that has been considered in this example so far. But the membership function was all inclusive on one end (short models) and restrictive on the other extreme (tall models). The trapezoidal fuzzy number allows the models height to be all inclusive or in full membership over a range of heights. For example, the modeling agency could say any girl between 5 feet 7 inches and 5 feet 10 inches should be picked, and some girls around these heights will be considered. This fuzzy number considering the models is shown is Graph 7 – 1. This is how decisions are made in the real world. Vagueness is becoming more and more prevalent in this example because of the complexity of the English language. But fuzzy numbers account for this lacking of precision with intervals of real numbers. The linguistic modifier used in this example is "around". The dilemma is how far from the interval does "around" represent? This is a very elementary concept because when someone uses a linguist modifier they have a crude, maybe even very precise, idea of the including area of acknowledgement. In this example the area around the interval is represented by 2 inches, one inch on both sides of the core number, which is also shown in Graph 7 – 2.[13]

Before venturing any further into fuzzy numbers some statements must be made about their construction. All true fuzzy numbers must conform to the following:

1. Fuzzy numbers are normal fuzzy sets.

2. The alpha cuts of every fuzzy number are closed intervals of real numbers.
3. The support of every real number is the open interval of real numbers.
4. Fuzzy numbers are convex fuzzy sets.



Graph 7 –1 : Trapezoidal Shaped Fuzzy Number of Model Height Parameters

These properties are very essential to be able to perform arithmetic operations on fuzzy numbers. The first criterion means that the fuzzy set must have a height of one. If the fuzzy set does not have a height of one it is called a subnormal fuzzy set. This concept of a normal fuzzy set suggests that the core of the fuzzy set not be empty as well. The second statement conveys the concept of alpha cuts being closed intervals of real numbers. This is a very important statement because the closed interval concept allows interval arithmetic to be performed on fuzzy numbers. Each alpha cut represents a small piece or slice of the fuzzy number. If this slice is defined on a closed interval, it is considered to be a conventional interval of real numbers. And hence, conventional interval mathematics can be performed on said intervals. The third criterion states that the support of the real number is the open interval of real numbers. This simply means that the realm were the fuzzy number is defined, between zero and one, contain all real numbers. The support is the portion of Graph 7 - 1 that is between heights 5 feet 6 inches and 6 feet. Since this is an open interval the end points are included in the representation. The last criterion states that fuzzy numbers are convex fuzzy sets. Fuzzy sets can be convex or show no conformity to convexion. A convex fuzzy set is intersected by an alpha cut twice. If there are multiple intersections on the fuzzy set it is considered to be non-convex. This concept of convexity and a fuzzy set is shown in Figure 7 - 4. The triangular and trapezoidal fuzzy numbers are very common, even popular, but they are not the only types of fuzzy numbers. Shown in Figure 7 - 5 are other types of fuzzy numbers similar to triangular and trapezoidal shapes. Another type of fuzzy number is one that enjoys smooth transition from zero to one and back to zero.

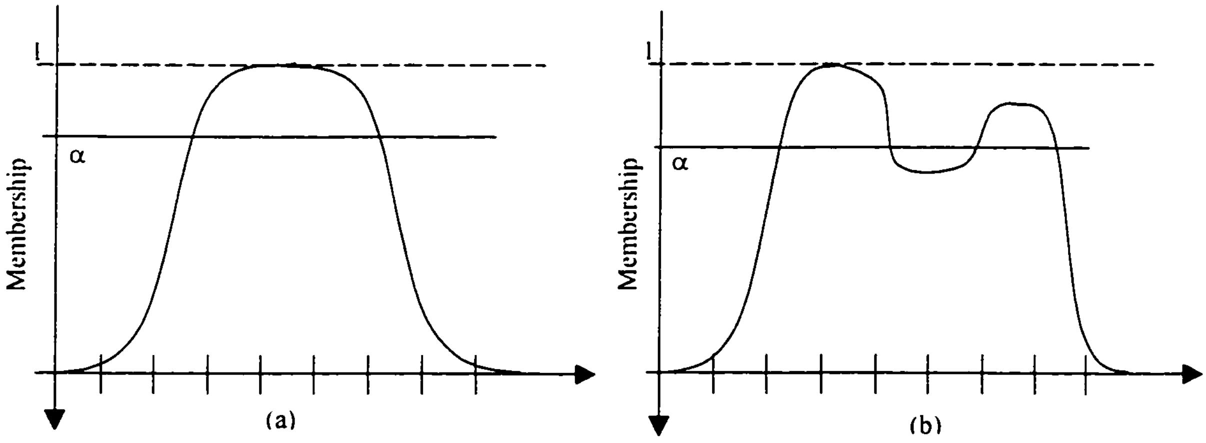


Figure 7 - 4 : (a) Convex Fuzzy Set, (b) Non-Convex Fuzzy Set

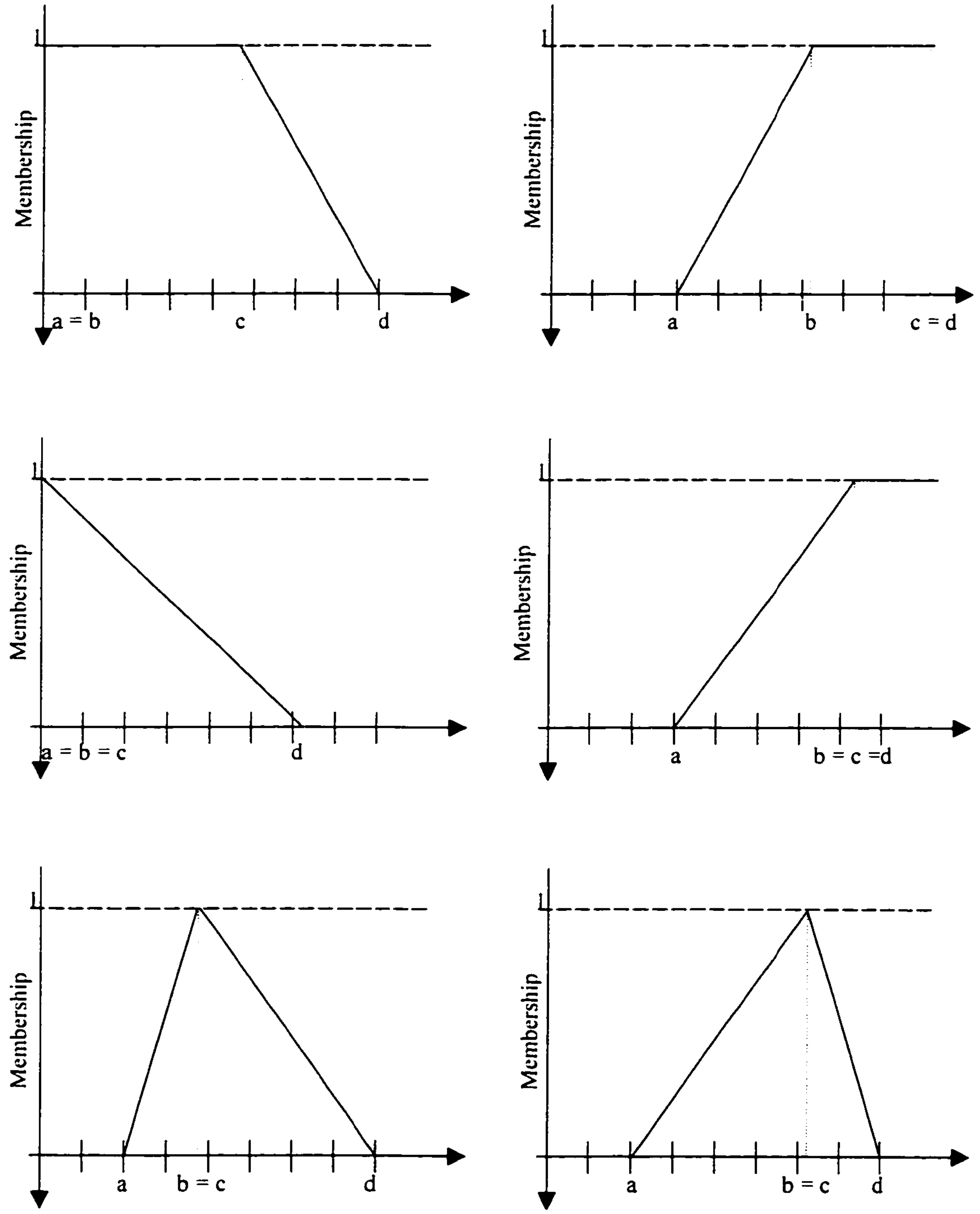


Figure 7 - 5 : Other Types of Triangular and Trapezoidal Shaped Fuzzy Numbers

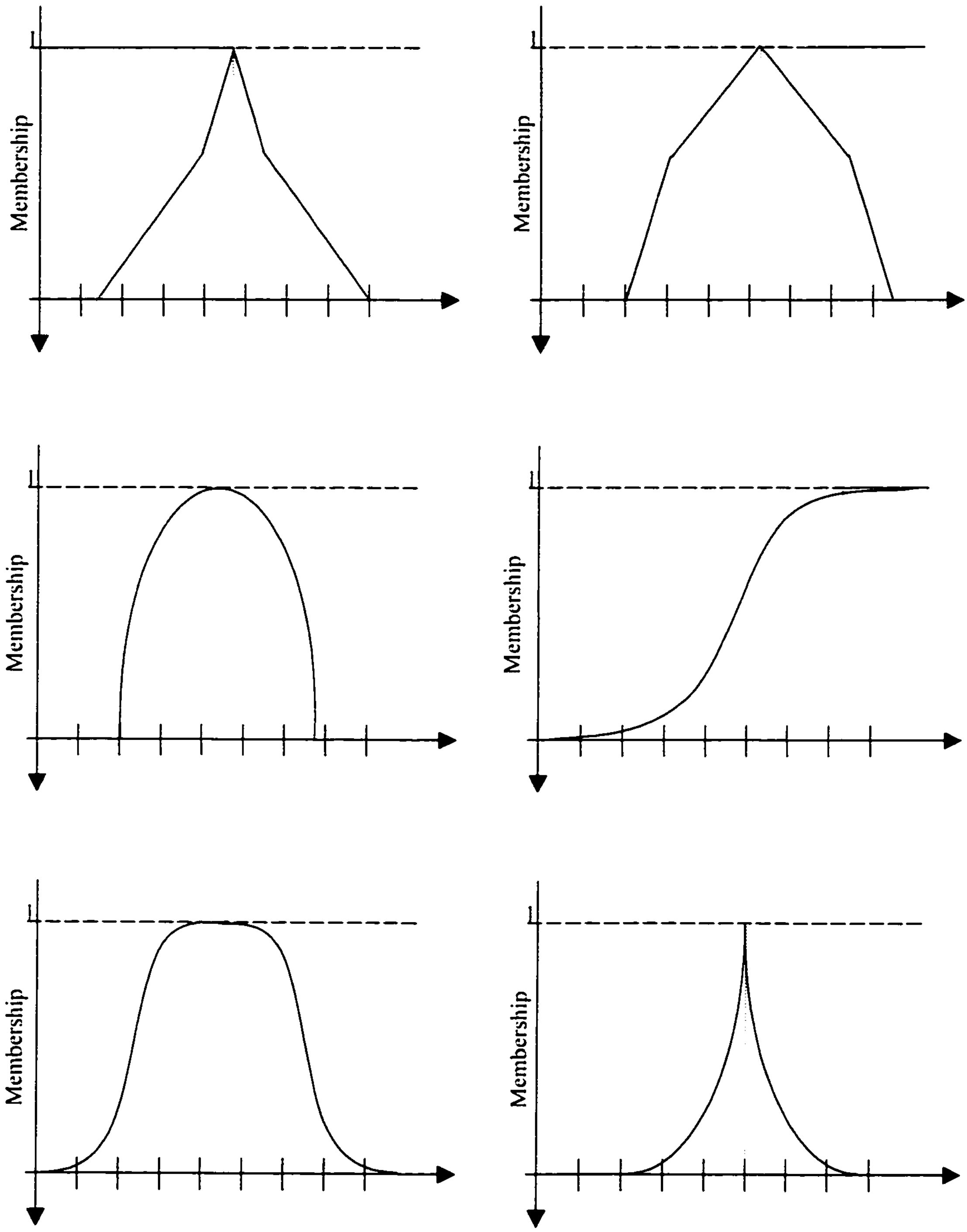


Figure 7 - 6 : Other Types of Gaussian and Exponential Shaped Fuzzy Numbers

The fuzzy number pictured in Figure 7 - 3 is called a bell shaped or a Gaussian fuzzy number. This fuzzy number is very useful in representing averages of large sets. Also. the Gaussian function has no sharp or sudden inclusive points that convey full membership. Other fuzzy numbers of this type and other popular types are shown in Figure 7 - 6.[30]

Another concept used in fuzzy numbers is called a linguistic variable. This takes the idea of a linguistic modifier one step further. These are variables whose states are fuzzy numbers. The occurrence of a linguistic variable happens because fuzzy numbers represent linguist concepts. such as very short. short. average. and so on. The linguistic variable allows manipulation on a base variable. This base variable is a collection of real numbers within a specific range. Base variables fall into two different categories:

1. Physical variable
2. Numerical variable

The physical variable is a description of a real world event. This event must be measurable by some physical means. Pressure. temperature. wind velocity. and light intensity are all physical variables. Examples of numerical variables are salary. efficiency. and interest rates. These variables do not represent a measurable real world event. One may contest that a salary is measurable with money but the basic idea of a salary is not measurable physically.

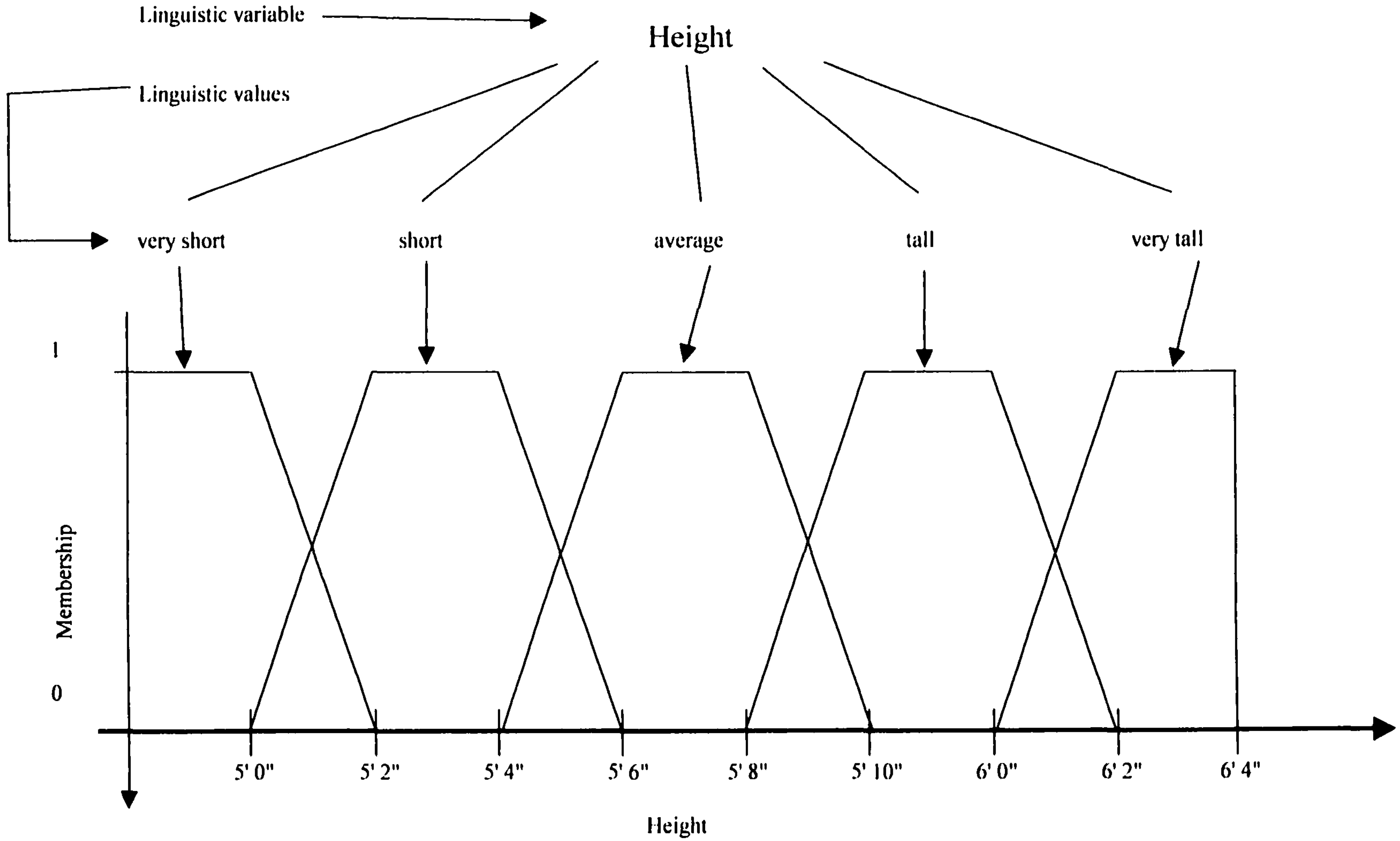


Figure 7 - 7 : Linguistic Variable of Model Example

The linguistic variable can be very useful in the model example. The linguistic variable allows a greater degree of selection and complexity to the concept of height. This is done by assigning height as the base variable and very short. short. medium. tall. and very tall as the linguistic modifiers. The model example utilizing the concept of linguistic variables has several membership functions. which convey a complete aspect of height. This is shown in Figure 7 – 7.

It must be noted that every linguistic variable is characterized by five distinct elements. The five terms representing these elements are *v*. T. X. *g*. and *m*. and each term is listed below with its definition.

1. *v* is the name of the linguistic variable
2. The set of linguistic terms of v that refers to a base variable is known as T.
3. The universal set where all variables exist is denoted by X.
4. *g* is the syntactic rule for generating linguistic terms.
5. *m* is a semantic rule that assigns to each linguistic term its meaning.

The statements above characterize a linguistic variable and can be applied to the model example. In Figure 7 – 7 a complete range of fuzzy numbers were used to develop the concept of height. The base variable in this example is defined on the x-axis. Five different fuzzy numbers subdivide the numerical range of different model heights. Three of the fuzzy numbers are trapezoidal and the other two can be classified as a degenerate case of the trapezoidal shape. As seen in Figure 7 – 7 these fuzzy numbers overlap each other. This overlap is not a combination of the fuzzy numbers but it is a distinct period of contradiction. The reason for this contradiction is that a height can be associated to more than one membership value corresponding to more than one fuzzy number. At first glance it is not apparent on which membership value the height corresponds. This will be defined by conditional rules. which are covered in the next chapter. It must also be noted that any of the other types of fuzzy numbers could have been used in place of ones found in Figure 7 – 7.[20]

Interval analysis is an area in mathematics that allows the description of uncertainty about the actual value of a numerical variable. Fuzzy numbers. with the assistance of alpha cuts. are classified as real number intervals. With this distinction interval mathematical operations can be applied. Typical arithmetic operations on intervals include addition. subtraction. multiplication and division. Using the model example. consider two intervals defined between 4 feet to 5 feet and 5 feet to 6 feet in height. These intervals represent no contribution to the example at hand. But it does supply two sets of intervals for arithmetic interval operations. The operations on these intervals are shown in Equations 7 – 3 to 7 - 15 and diagrams 7 – 8 to 7 – 11. In the interval division example 0 is assumed not to be one of the elements c or d.

$[4, 5]$ and $[5, 6]$ .... Equation 7 – 3

$[a, b] + [c, d] = [a + c, b + d]$ .... Equation 7 – 4

$[4, 5] + [5, 6] = [4 + 5, 5 + 6]$ .... Equation 7 – 5

$[9, 11]$ .... Equation 7 – 6

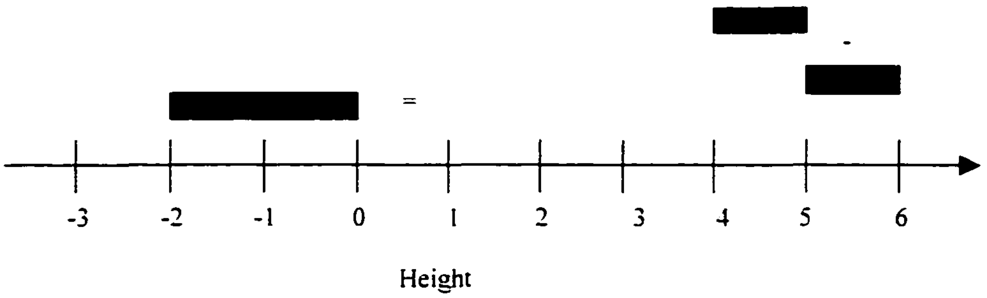


Figure 7 – 8 : Interval Addition

[a, b] - [c, d] = [a - d, b - c]...Equation 7 – 7

[4, 5] - [5, 6] = [4 - 6, 5 - 5]...Equation 7 – 8

[-2, 0]...Equation 7 – 9

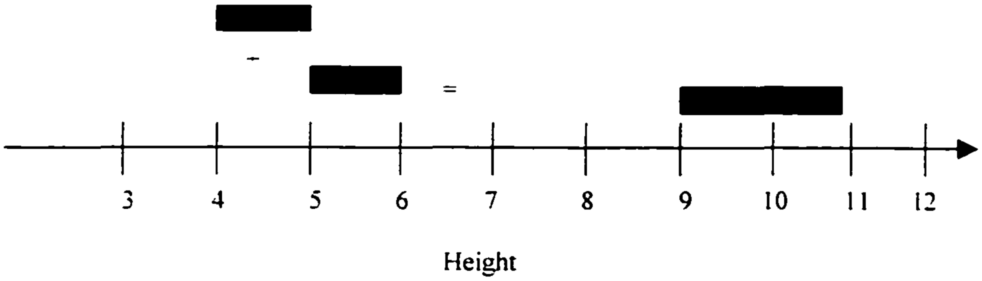


Figure 7 – 9 : Interval Subtraction

[a, b] • [c, d] = [min (ac, ad, bc, bd), max(ac, ad, bc, db)]..........Equation 7 – 10

[4, 5] • [5, 6] = [min(4 • 5), (4 • 6), (5 • 5), (5 • 6)........................Equation 7 – 11

max(4 • 5), (4 • 6), (5 • 5), (5 • 6)]

[20, 30]..Equation 7 – 12

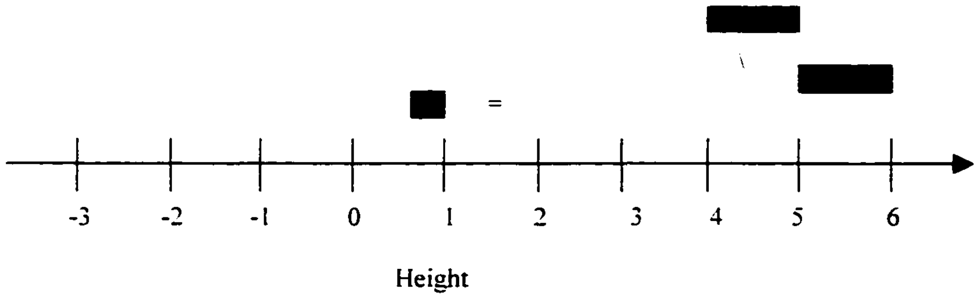


Figure 7 – 10 : Interval Multiplication

[a, b] / [c, d] = [a, b] • [1 / d, 1 / c]...Equation 7 – 13

= [min (a / c, a / d, b / c, b / d),

max (a / c, a / d, b / c, b / d)]

[4, 5] / [5, 6] = [min(4 / 5), (4 / 6), (5 / 5), (5 / 6)...........................Equation 7 – 14

max(4 / 5), (4 / 6), (5 / 5), (5 / 6)]

[0.67, 1]..Equation 7 – 15

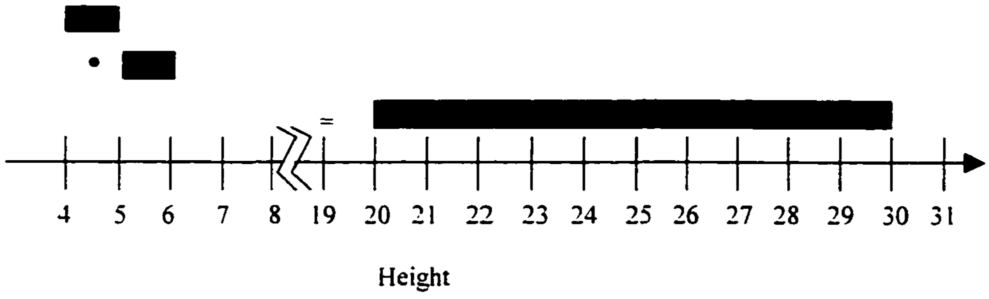
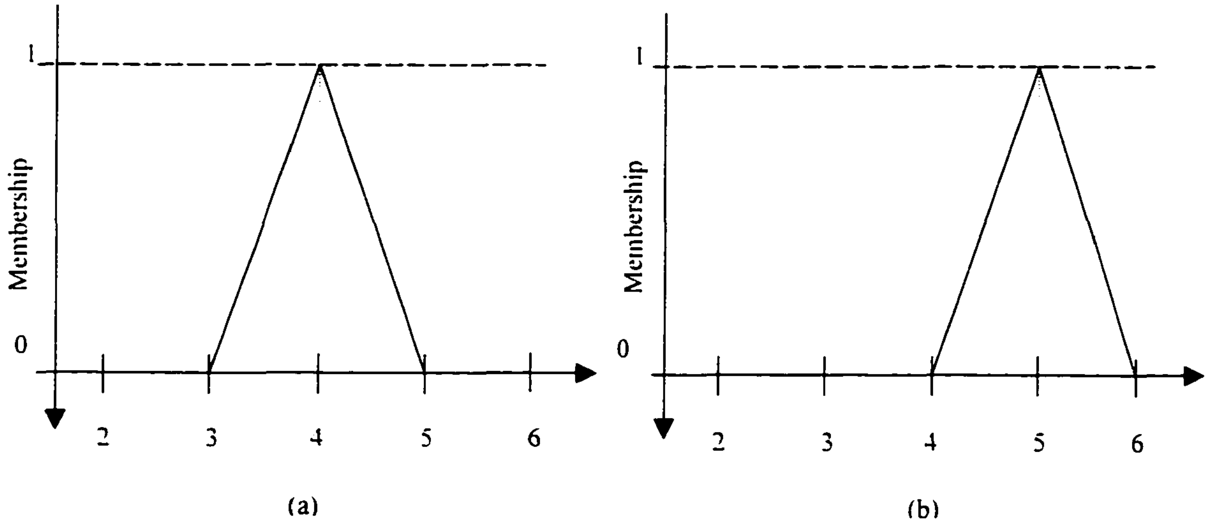
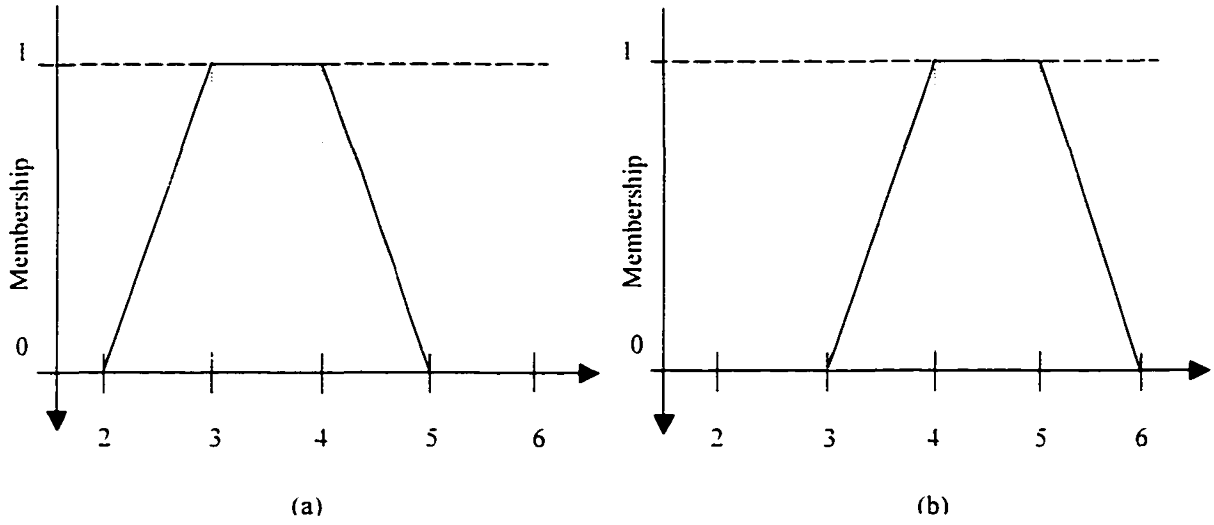


Figure 7 – 11 : Interval Division

Interval mathematics is an excellent tool for fuzzy numbers because of its ease of implementation. But there are some drawbacks when the interval operations are applied to fuzzy numbers. The addition and subtraction interval operations are well suited to trapezoidal and triangular fuzzy numbers, although this is a one-dimensional solution to interval mathematical problems. Fuzzy numbers have a degree of membership, and this solution falls apart brought the two-dimensional realm concerning different types of fuzzy numbers. However, interval addition and subtraction can be applied to the four fuzzy numbers shown in Graphs 7 – 2 and 7 – 3.



Graph 7 - 2 : (a) Triangular Shaped Fuzzy Number with a Defining Interval [3, 5]
(b) Triangular Shaped Fuzzy Number with a Defining Interval [4, 6]



Graph 7 - 3 : (a) Trapezoidal Shaped Fuzzy Number with a Defining Interval [2, 5]
(b) Trapezoidal Shaped Fuzzy Number with a Defining Interval [3,6]

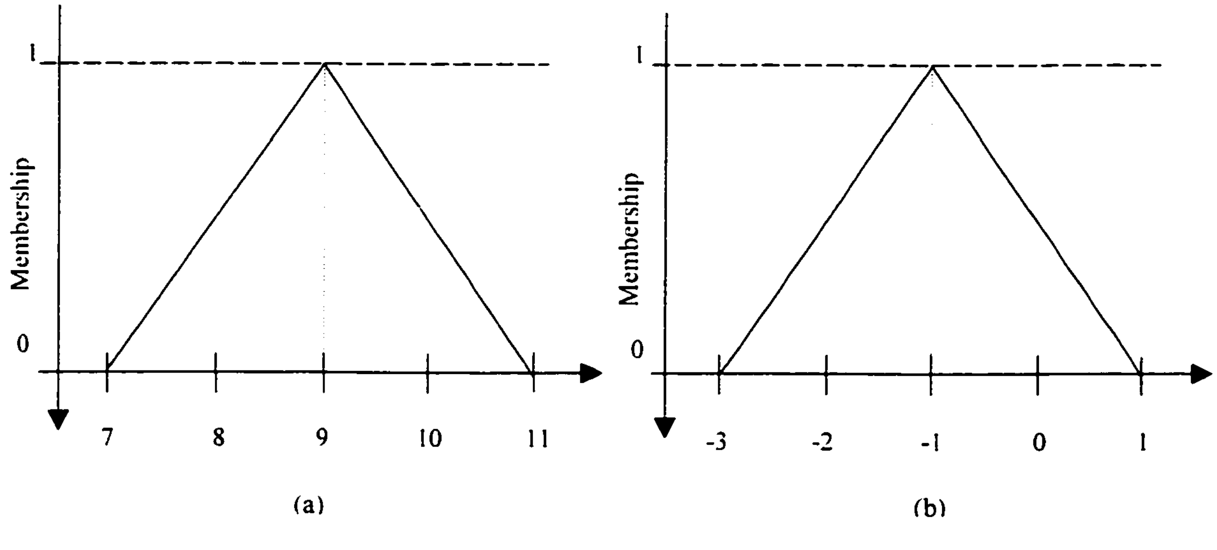
These fuzzy numbers are of triangular and trapezoidal shapes and can be operated on by interval addition and subtraction. Graphs 7 – 4 and 7 – 5 are the results of the addition and subtraction of the fuzzy numbers found in Graphs 7 – 3 and 7 – 4. It must also be noted that multiplication and division are not applicable to fuzzy numbers in the simplified form. Therefore, alpha cuts must be utilized to perform arithmetic operations on fuzzy numbers. To formulate any of the four basic arithmetic operations on arbitrary fuzzy numbers, one represents the numbers by their alpha cuts and employ interval arithmetic to the alpha cuts. In order to develop this concept, one first must consider two fuzzy numbers, A and B, and let * denote any of the four interval arithmetic operations. Then, for each interval the alpha cut of A * B is defined in terms of the alpha cuts of A and B. This is formally defined by Equation 7 – 16. In the case

$$^{\alpha}(A * B) = {}^{\alpha}A * {}^{\alpha}B \text{..Equation 7 – 16}$$

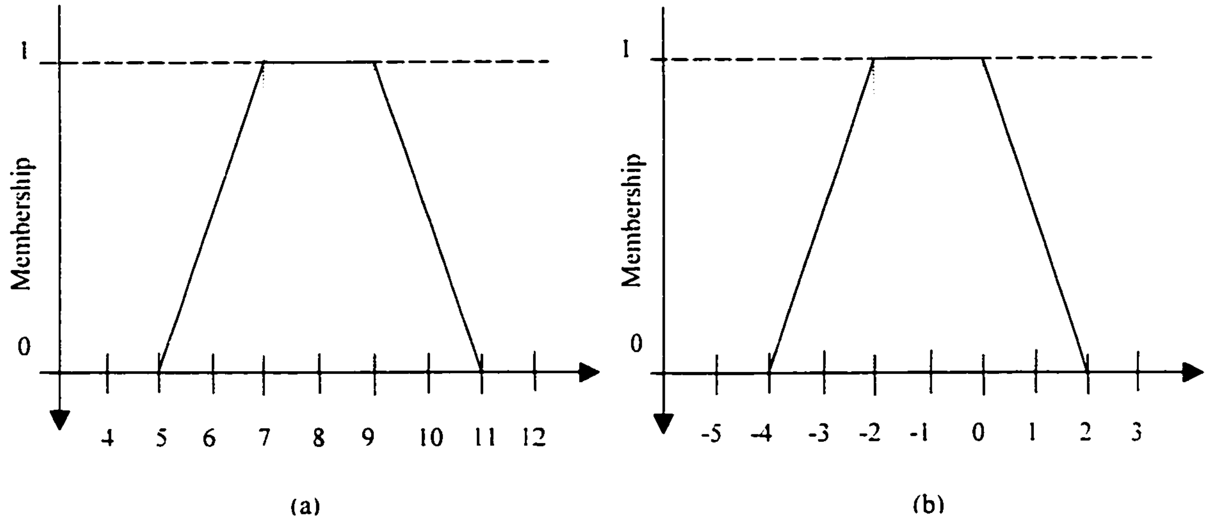
of division, B cannot contain zero in its interval. Once alpha cuts $^{\alpha}(A * B)$ are determined, the resulting fuzzy number A * B is expressed by Equation 7 – 17, where

$$A * B = \bigcup_{\alpha \in [0,1]} {}_{\alpha}(A * B) \ldots\ldots\ldots\ldots \text{Equation } 7-17$$

$\cup$ is the standard fuzzy union of a special fuzzy set defined for each $x \in X$. Considering the former, all arithmetic operations can be applied to the two triangular shaped fuzzy numbers found in Graph 7 – 2. The formulas for these two fuzzy numbers are shown in Equations 7 – 18 and 7 – 19. But first the intervals must be defined to satisfy the requirements of alpha cuts. These requirements were covered in Chapter VI.



Graph 7 - 4 : (a) Addition of Two Triangular Shaped Fuzzy Numbers with Defining Intervals [3, 5] and [4, 6]
(b) Subtraction of Two Triangular Shaped Fuzzy Numbers with Defining Intervals [3, 5] and [4, 6]



Graph 7 - 5 : (a) Subtraction of Two Trapezoidal Shaped Fuzzy Numbers with Defining Intervals [2, 5] and [3, 6]
(b) Subtraction of Two Trapezoidal Shaped Fuzzy Numbers with Defining Intervals [2, 5] and [3, 6]

The fuzzy number defined by A(x) is dissected by an alpha cut and is shown in Figure 7 – 12. This is very important because this alpha cut forms two end points which can be classified as an interval. This interval is of general format and the fuzzy number

$$A(x) = \begin{cases} 0 & \text{for } x < 3 \text{ and } x > 5 \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Equation } 7-18 \\ (x - 3) & \text{for } 3 < x \leq 4 \\ (5 - x) & \text{for } 4 < x < 5 \end{cases}$$

$$B(x) = \begin{cases} 0 & \text{for } x < 4 \text{ and } x > 6 \ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Equation } 7-19 \\ (x - 4) & \text{for } 4 < x \leq 5 \\ (6 - x) & \text{for } 5 < x < 6 \end{cases}$$

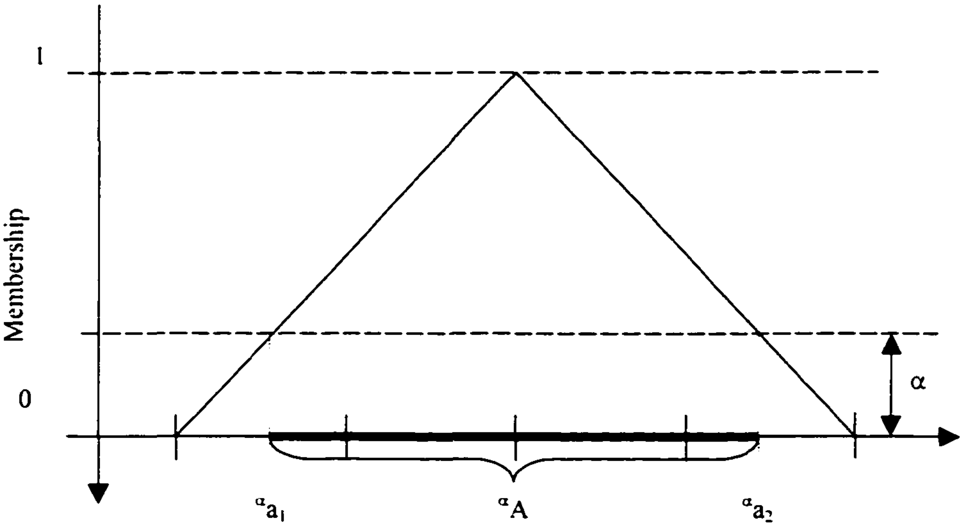


Figure 7 - 12 : Alpha Cut Applied to a Fuzzy Number

B(x) would be very similar to Figure to 7 -12, so a graphical example is not pursued.

The interval defined in Figure 7 - 12 is more precisely defined by its endpoints in Equations 7 - 20 and 7 - 21. These equations are referenced back to Graph 7 - 2 where fuzzy numbers are defined by A(x) and B(x). The endpoints of the intervals

$^{\alpha}A = [^{\alpha}a_1, {}^{\alpha}a_2]$ .... Equation 7 – 20

$^{\alpha}B = [^{\alpha}b_1, {}^{\alpha}b_2]$ .... Equation 7 – 21 defined by the alpha cuts are substituted into the formula of both functions (Equations 7 - 18 and 7 - 19), which are shown in Equations 7 - 22 through 7 - 25. These previous equations can be considered as transforming the formulas from the x domain to the alpha domain. But since this is a direct translation this terminology would be considered redundant.

$A\,({}^{\alpha}a_1) = ({}^{\alpha}a_1 - 3) = \alpha$ ....Equation 7 – 22

$A\,({}^{\alpha}a_2) = (5 - {}^{\alpha}a_2) = \alpha$ ....Equation 7 – 23

$B\,({}^{\alpha}b_1) = ({}^{\alpha}b_1 - 4) = \alpha$ ....Equation 7 – 24

$B\,({}^{\alpha}b_2) = (6 - {}^{\alpha}b_2) = \alpha$ ....Equation 7 – 25

The reason for these substitutions is to find the value of the endpoint associated with alpha cuts. Equations 7 - 22 through 7 - 25 display the intervals that

${}^{\alpha}a_1 = 3 + \alpha$ ....Equation 7 – 26

${}^{\alpha}a_2 = 5 - \alpha$ ....Equation 7 – 27

${}^{\alpha}b_1 = 4 + \alpha$ ....Equation 7 – 28

$^{\alpha}b_2 = 6 - \alpha$ ....Equation 7 – 29

are associated with an endpoint. These equations are set equal to the alpha cut that is transposed to the x axis. Solving for the endpoint variable is done for each part of the two fuzzy numbers and is shown in Equations 7 - 26 through 7 - 29. From these equations, $^{\alpha}a_1$, $^{\alpha}a_2$, $^{\alpha}b_1$, and $^{\alpha}b_2$ can be substituted into Equations 7 - 20 and 7 - 21 to achieve Equations 7 - 30 and 7 - 31. Equation 7 - 32 is the representation of both alpha

$^{\alpha}A = [3 + \alpha, 5 - \alpha]$ ....Equation 7 – 30

$^{\alpha}B = [4 + \alpha, 6 - \alpha]$ ....Equation 7 – 31

$^{\alpha}(A * B) = [3 + \alpha, 5 - \alpha] * [4 + \alpha, 6 - \alpha]$ ....Equation 7 – 32

cut endpoints or alpha intervals of the corresponding fuzzy numbers. The * operation represents all arithmetic operations for the two endpoints considered.

The first operation explained is addition. Using the Equation 7 - 4 on the four endpoints, Equation 7 – 33 is generated. From this equation simple interval arithmetic is applied and the result is Equation 7 – 34, followed by Equation 7 – 35. Equation 7 – 36 finds out where these endpoints are valid. Each equation is set equal to x and 0 and 1 are substituted in for alpha. The result of Equation 7 – 36 is 7 – 37, which represents

$^{\alpha}(A + B) = [3 + \alpha, 5 - \alpha] + [4 + \alpha, 6 - \alpha]$ ....Equation 7 – 33

$^{\alpha}(A + B) = [3 + \alpha + 4 + \alpha, 5 - \alpha + 6 - \alpha]$ ....Equation 7 – 34

$^{\alpha}(A + B) = [7 + 2\alpha, 11 - 2\alpha]$ ....Equation 7 – 35

$7 + 2\alpha = x$ and $11 - 2\alpha = x$ ....Equation 7 – 36

closed and open-ended intervals for which these equations satisfy the definition of alpha cuts. Equation 7 – 38 is the solution of 7 – 36 when alpha is solved for and the equation

$x \in (7, 9]$ and $x \in [9, 11)$..Equation 7 – 37

is solely comprised of the element x. Taking all the previous information into consideration the equation of the new fuzzy number is shown in Equation 7 - 39. The equation is plotted in Graph 7 – 7 along with the fuzzy numbers A and B. Subtraction,

$\alpha = (x - 7)/2$ and $\alpha = (11 - x)/2$....................................Equation 7 – 38

multiplication, and division are all done in the same fashion. Subtraction is represented by Equations 7 – 40 through 7 – 46 and Graph 7 – 7. Multiplication is shown by Equations 7 – 47 through 7 – 53 and Graph 7 – 8. And finally division is represented by Equations 7 – 54 through 7 – 59 and Graph 7 – 9.

$$A + B = \begin{cases} 0 & \ldots\text{ for } x < 7 \text{ and } x > 11 \ldots\ldots \text{ Equation } 7-39 \\ (x - 7)/2 & \text{for } 7 \le x \le 9 \\ (11 - x)/2 & \text{for } 9 \le x \le 11 \end{cases}$$

$${}^{\alpha}(A - B) = [3 + \alpha, 5 - \alpha] - [4 + \alpha, 6 - \alpha] \ldots\ldots \text{Equation } 7-40$$

$${}^{\alpha}(A - B) = [3 + \alpha - 6 + \alpha,\ 5 - \alpha - 4 - \alpha] \ldots\ldots \text{Equation } 7-41$$

$${}^{\alpha}(A - B) = [-3 + 2\alpha, -1 - 2\alpha] \ldots\ldots \text{Equation } 7-42$$

$$-3 + 2\alpha = x \text{ and } 1 - 2\alpha = x \ldots\ldots \text{Equation } 7-43$$

$$x \in (-3,-1] \text{ and } x \in [-1, 1) \ldots\ldots \text{Equation } 7-44$$

$$\alpha = (x + 3)/2 \quad \text{and} \quad \alpha = (1 - x)/2 \ldots\ldots \text{Equation } 7-45$$

$$A - B = \begin{cases} 0 & \text{for } x < -3 \text{ and } x > 1 \ldots\ldots \text{ Equation } 7-46 \\ (x + 3)/2 & \text{for } -3 \le x \le -1 \\ (1 - x)/2 & \text{for } -1 \le x \le 1 \end{cases}$$

$${}^{\alpha}(A \bullet B) = [\min ((3 + \alpha)(4 + \alpha), (3 + \alpha)(6 - \alpha),(5 - \alpha) \ldots\ldots \text{ Equation } 7-47$$

$$(4 + \alpha), (5 - \alpha)(6 - \alpha)), \max (3 + \alpha)(4 + \alpha),$$

$$(3 + \alpha)(6 - \alpha),(5 - \alpha)(4 + \alpha),\alpha), (5 - \alpha)(6 - \alpha)]$$

$$^{\alpha}(\,A \bullet B\,) = [\min(\alpha^2 + 7\alpha + 12, -\alpha^2 + 3\alpha + 18, \alpha^2 - 11\alpha + 30 \ldots\ldots \text{Equation 7 – 48}$$

$$\alpha^2 + \alpha + 20), \max(\alpha^2 + 7\alpha + 12, -\alpha^2 + 3\alpha + 18,$$

$$\alpha^2 - 11\alpha + 30, \alpha^2 + \alpha + 20)]$$

$$^{\alpha}(\,A \bullet B\,) = [\alpha^2 + 7\alpha + 12, \alpha^2 - 11\alpha + 30] \ldots\ldots\ldots\ldots\ldots\ldots \text{Equation 7 – 49}$$

$$\alpha^2 + 7\alpha + 12 = x \text{ and } \alpha^2 - 11\alpha + 30 = x \ldots\ldots\ldots\ldots\ldots\ldots \text{Equation 7 – 50}$$

$$x \in (12, 20] \text{ and } x \in [20, 30) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Equation 7 – 51}$$

$$\alpha = (x + 0.25)^{1/2} - 3.5 \text{ and } -(x + 0.25)^{1/2} + 5.5 \ldots\ldots\ldots\ldots \text{Equation 7 – 52}$$

$$A \bullet B = \begin{cases} 0 & \text{for } x < 12 \text{ and } x > 30 \ldots\ldots \text{ Equation 7 – 53} \\ (x + 0.25)^{1/2} - 3.5 & \text{for } 12 \le x \le 20 \\ -(x + 0.25)^{1/2} + 5.5 & \text{for } 20 \le x \le 30 \end{cases}$$

$$^{\alpha}(\,A / B\,) = [\min((3 + \alpha) / (4 + \alpha), (3 + \alpha) / (6 - \alpha).(5 - \alpha) / \ldots\ldots \text{Equation 7 – 54}$$

$$(4 + \alpha), (5 - \alpha) / (6 - \alpha)), \max(3 + \alpha) / (4 + \alpha),$$

$$(3 + \alpha) / (6 - \alpha).(5 - \alpha) / (4 + \alpha), (5 - \alpha) / (6 - \alpha)]$$

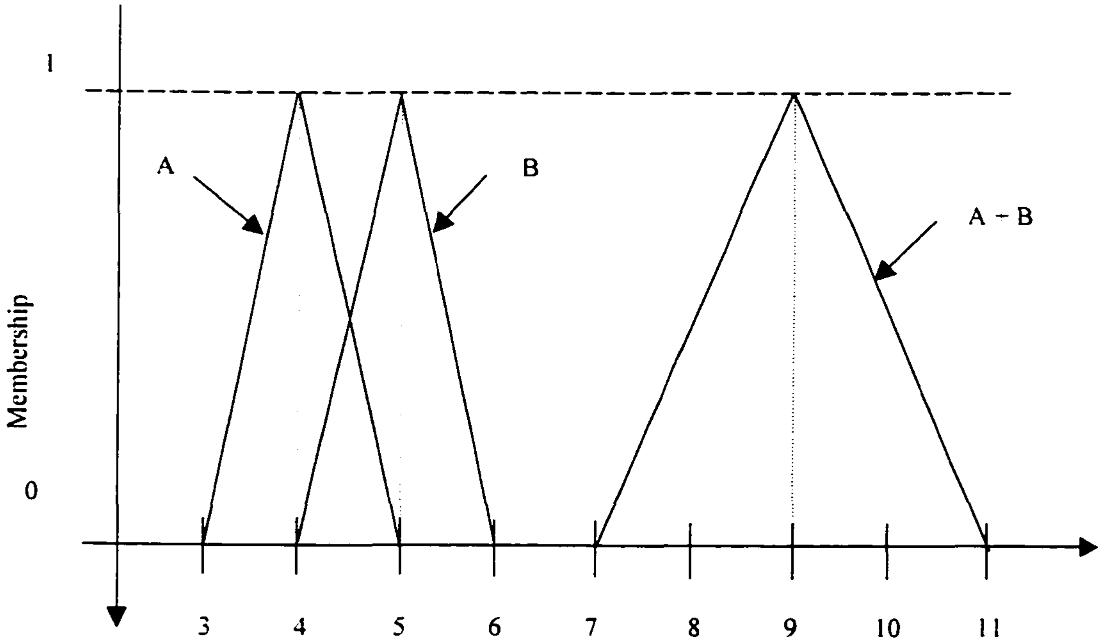
$^{\alpha}(A / B) = (3 + \alpha) / (6 - \alpha), (5 - \alpha) / (4 + \alpha)$.......................... Equation 7 – 55

$(3 + \alpha) / (6 - \alpha) = x$ and $(5 - \alpha) / (4 + \alpha) = x$...................... Equation 7 – 56

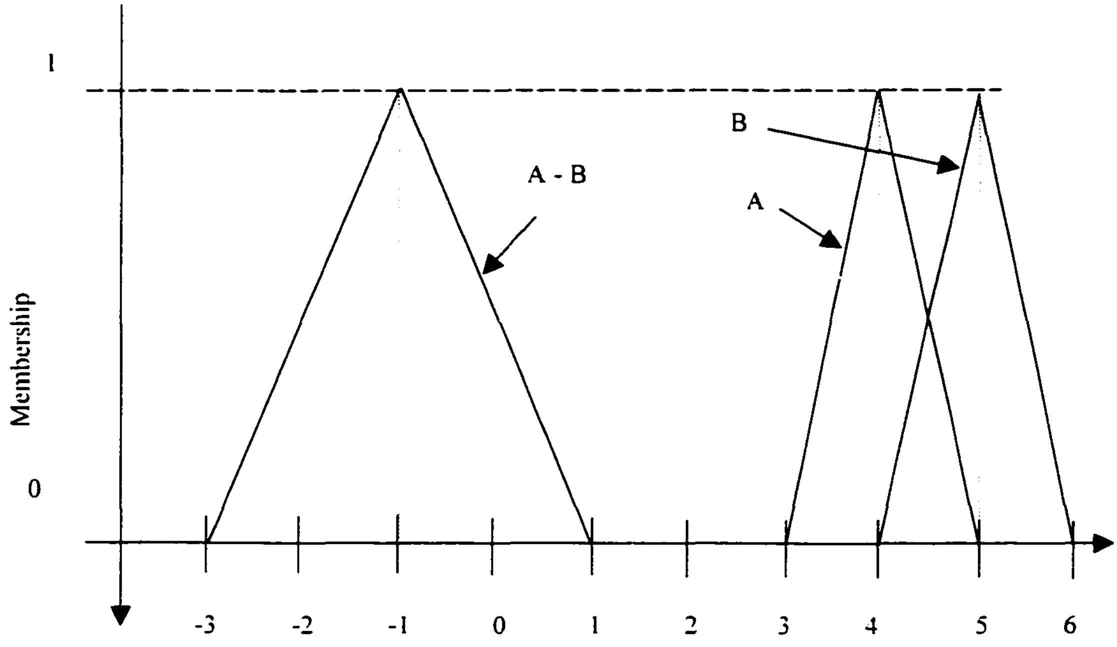
$x \in (1/2, 5/4]$ and $x \in [5/4, 5/4)$.................................... Equation 7 – 57

$\alpha = (6x - 3) / (1 + x)$ and $\alpha = (5 - 4x) / (x + 1)$......................Equation 7 – 58

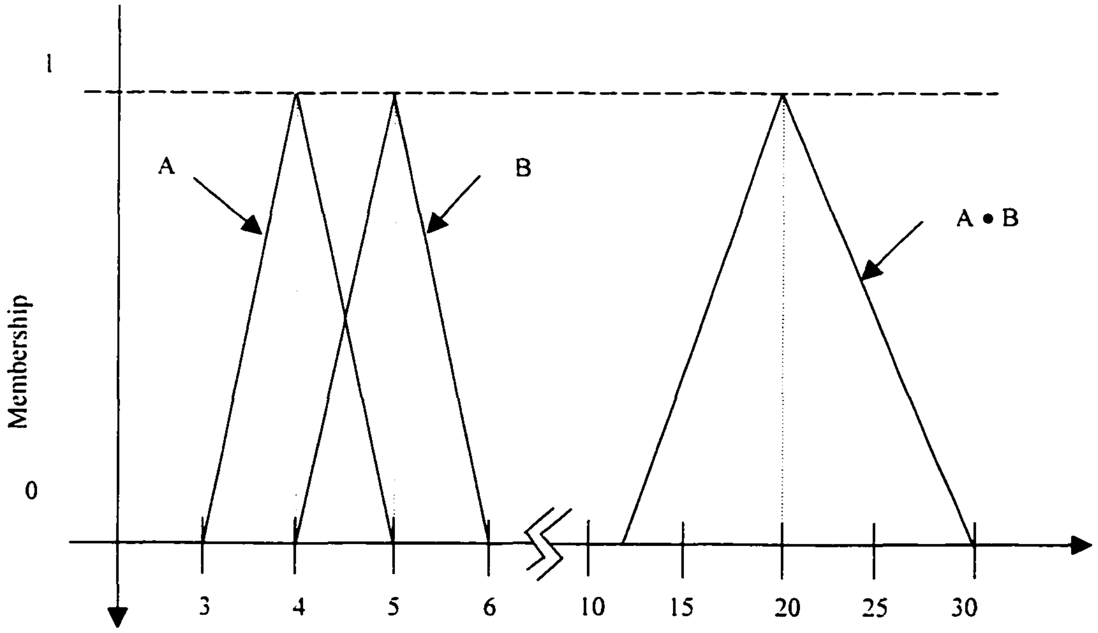
$$A / B = \begin{cases} 0 & \text{for } x < 1/2 \text{ and } x > 5/4 \\ (-3 + 6x) / (1 + x) & \text{for } 3/6 \leq x \leq 4/5 \\ (5 - 4x) / (x + 1) & \text{for } 4/5 \leq x \leq 5/4 \end{cases}$$ .....Equation 7 – 59



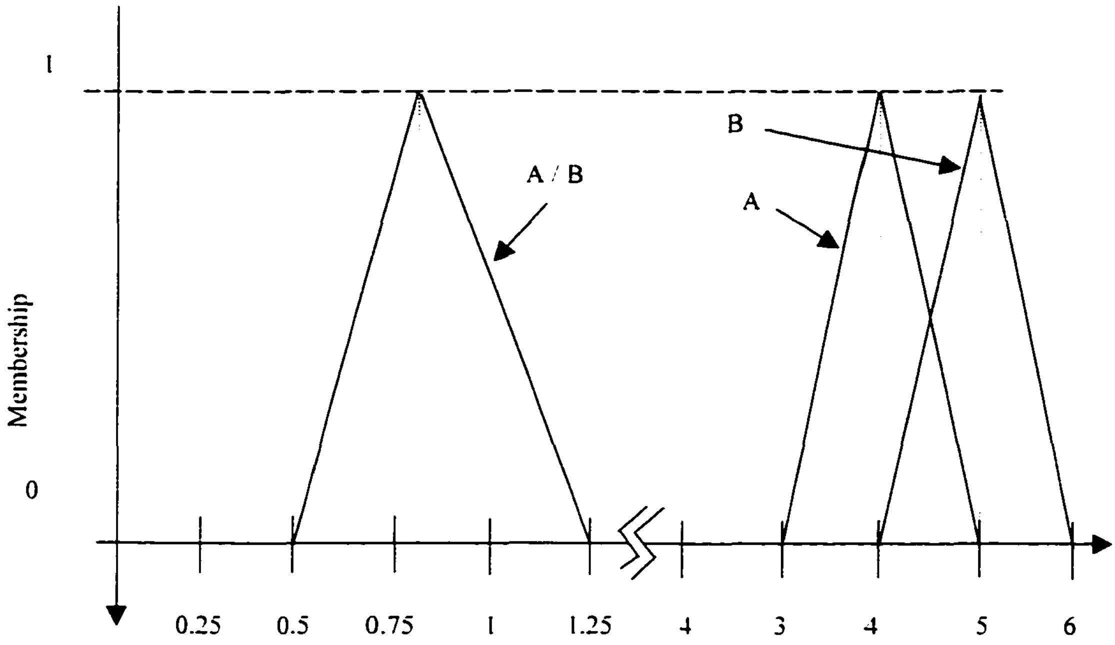
Graph 7 - 6 : Addition of Two Triangular Shaped Fuzzy Numbers using Alpha Technique



Graph 7 - 7 : Subtraction of Two Triangular Shaped Fuzzy Numbers using Alpha Technique



Graph 7 - 8 : Multiplication of Two Triangular Shaped Fuzzy Numbers using Alpha Technique



Graph 7 - 9 : Division of Two Triangular Shaped Fuzzy Numbers using Alpha Technique

CHAPTER VIII

FUZZY LOGIC

The chapters preceding this one have dealt with ideas of set theory, fuzzy set theory and fuzzy mathematics. These topics lead to the development of fuzzy logic. Fuzzy logic is the evolution from two-valued logic to continuous logic. There are other logic fields that fall between the previous two types of logic. These other logic fields are three-valued logic and n valued logic.

Lukasiewicz described three-valued logic in the early 1900's. The third value he proposed can best be translated as the term "possible," and he assigned it a numeric value between True and False. This value assigned is a medium representing not totally true and not totally false. This term takes a numerical value of one half (1/2). Eventually he proposed an entire notation and axiomatic system from which he hoped to derive modern mathematics. The connectives for this three valued logic is shown in Table 8 – 1. This table shows one of several connectives for three-valued logic. Lukasiewicz was not the only person to conceive the notion of three-valued logic. There were others, such as Bochvar, Kleene, Heyting, and Reichenbach to conceive a system of three-valued logic. It must be noted that each of these three valued logic connectives are different. Therefore, this is one reason three valued logic has never truly gained great acceptance in mathematics.

Later. Lukasiewicz explored four-valued logics. five-valued logics. and then declared that in principle there was nothing to prevent the derivation of an infinite-valued logic. Lukasiewicz felt that three- and infinite-valued logic's were the most intriguing, but he ultimately settled on a four-valued logic because it seemed to be the most easily adaptable to Aristotelian logic.

Table 8 – 1 : Lukasiewicz Connectives of Three Valued Logic

a	b	Lukasiewicz			
		∧	∨	⇒	⇔
0	0	0	0	1	1
0	1/2	0	1/2	1	1/2
0	1	0	1	1	0
1/2	0	0	1/2	1/2	1/2
1/2	1/2	1/2	1/2	1	1
1/2	1	1/2	1	1	1/2
1	0	0	1	0	0
1	1/2	1/2	1	1/2	1/2
1	1	1	1	1	1

N-valued logic or many-valued logic is defined by n number of truth values that a proposition may have in some particular logic. N valued logic is used to describe many different types of logic, such as the five-valued logic. The states of the five-valued logic are 0, ¼, ½, ¾, and 1. These states show a very true or false of the subject being studied. This logic set is finite and truly lacks the continuous truth and false membership as fuzzy logic enjoys. Fuzzy logic is very similar to n-valued logic, but with one exception: fuzzy logic is infinite valued and continuous.

The notion central to fuzzy logic systems is that truth-values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0, 1], with zero representing absolute falseness and one representing absolute truth. For example, consider the statement:

"Jane is old."

If Jane's age is 75, the corresponding truth value could be 0.80. The statement could be translated into set terminology as follows:

"Jane is a member of the set of old people."

This statement would be rendered symbolically with fuzzy sets as:

mOLD(Jane) = 0.80

where m is the membership function, operating in this case on the fuzzy set of old people, which returns a value between 0 and 1.

The previous example brings up several concepts into consideration, such as fuzzy propositions, fuzzy quantifiers, and linguistically hedges. All of these concepts build on the linguistic philosophy that characterizes fuzzy logic from other types of logic. The fuzzy propositions, are the first concept and are divided into four different areas:

- Unconditional and unqualified propositions
- Unconditional and qualified propositions
- Conditional and unqualified propositions
- Conditional and qualified propositions

The unconditional and unqualified propositions are assertions that are not in the traditional if-then format. These propositions are asserted to be simply true with the truth values not qualified by any modifying expression. One example of this type of proposition would be:

The girl who stands 6 feet is tall.

This proposition is formally defined in Equation 8 – 1. P stands for the unconditional

p : X is A..Equation 8 - 1 and unqualified preposition, X is a variable, and A is some property, or predicated attributed to the variable. Applying the previous to the statement above, the parts of this proposition are as follows: "the girl who stands 6 feet" is X (the variable) and "is tall" is A (the property of the variable).

The second propositions discussed are unconditional and qualified propositions. These are propositions making the unconditional assertion that another proposition has a qualified truth-value. This type of proposition is formally defined by Equation 8 – 2.

p : 'X is A' is S..Equation 8 - 2

In this equation p, X, and A mean the same as stated for the unconditional and unqualified propositions in Equation 8 – 1. S is what makes Equation 8 – 1 different from Equation 8 – 2. The S stands for a fuzzy truth quantifier, which is a linguistic expression that adds a modifier to the claim of simple truth. The propositions of this form are also known as truth qualified. An example of the fuzzy truth quantifier is as follows:

p(6) : 'The girl who stands 6 feet is tall' is S

The girl's height of 6 feet is shown by the vertical line in Figure 8 – 1, which dissects the three falsity lines. These three lines represent falsity, hence the line with the lowest value of the corresponding height conveys the highest truth value. The lines can

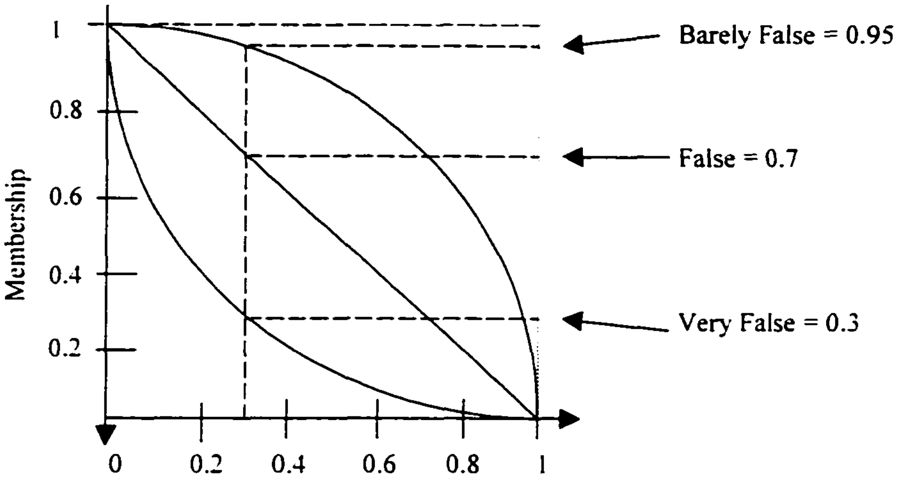


Figure 8 – 1 : Illustration of the Truth Quantifier

represent truth and the curves would start at the origin and end at value [1, 1]. This type of figure is sometimes called a unity truth graph. For S to validate the example of the truth quantifier, S would have to be of value 0.3. This value of true for the girl height is expressed in Equation 8 – 3. The level of membership of 0.3 must not be

$T_s\ (p_6) = S(0.3)$ .... Equation 8 - 3

confused with the falsity value found on the y-axis. This value of S corresponds with the vertical line that intersects all three falsity lines.

The third proposition is the conditional and unqualified proposition. This proposition is constructed from the if-then conditional statement. A formal definition of this proposition is shown in Equation 8 – 4. These types of propositions are also known

p : 'If X is A, then y is B' is true.......................................Equation 8 - 4

as fuzzy implications and contain simple fuzzy propositions as antecedent and consequent. One example of this type of proposition is as follows:

'If the girl is tall, then the boy is short' is true.

This is why these types of propositions are unqualified. The truth-value of this last example is of questionable validity. The girl could be tall but the boy does not have to be considered short. This if-then format is handled well in fuzzy logic, where the girl is a degree of being tall and the boy is of a degree of being short. In classical logic this is a true or false statement, but fuzzy logic allows degrees of truth and false.

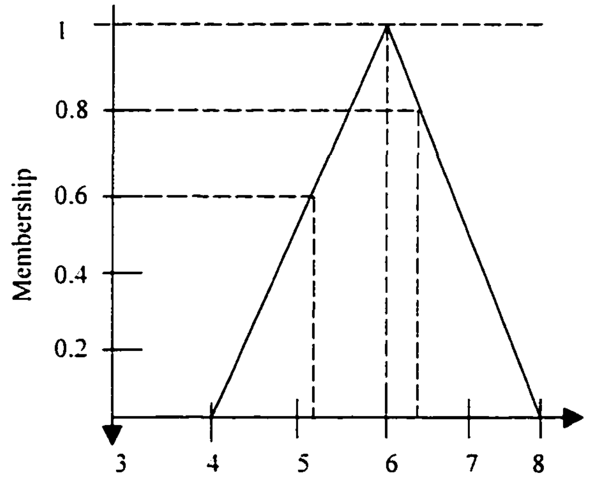
The last proposition is called the conditional and qualified proposition. This type of proposition is a combination between unconditional and qualified proposition and conditional and unqualified propositions. Equation 8 – 5 formally

p : 'If X is A, then y is B' is true.......................................Equation 8 – 5

defines the conditional and qualified proposition. This type of proposition is similar to other types of propositions and the methods used to deal with the other three can be applied here as well.

The second important area of fuzzy logic is known as fuzzy quantifiers. Fuzzy quantifiers are fuzzy numbers that take part in fuzzy propositions. The fuzzy quantifier is very useful because of the imprecision of the English language. To solidify this idea of the definition of this type of fuzzy quantifier, the concept of fuzzy sets and fuzzy numbers must be revisited in Chapters V, VI and VII.

There are two kinds of fuzzy quantifiers. The first fuzzy quantifier is the absolute quantifier, which is expressed by fuzzy numbers defined on the set of real numbers or on a set of integers. This type of quantifier characterizes linguistic terms such as almost a gallon, about 6 feet, and at least a mile. An example of the fuzzy quantifier approximating a number is shown in Graph 8 – 1. This figure demonstrates the importance of fuzzy numbers to the concept of the fuzzy quantifier. Graph 8 – 1



Graph 8 – 1 : Fuzzy Set of about 6 Feet in a Qualified Fuzzy Proposition displays the concept of 'about 6 feet,' and from this one can consider any membership of 0.5 and over categorized to be almost 6 feet tall. Or more precisely, the membership functions of over 0.5 being considered true to very true can be almost 6 feet tall.

Examples of some controversial and uncontroversial propositions are as follows:

Controversial

- 10 girls in the group are tall.
- 50% of cats destroy furniture.
- 200 students wait to the last minute to study for a test.

Uncontroversial

- About 10 girls in the group are tall.
- At least about 50% of cats destroy furniture.
- Much more than 100 students wait to the last minute to study for a test.

The difference from the controversial and the uncontroversial phrases is the presence of the fuzzy quantifier. The fuzzy quantifier does not allow broad sweeping accusations or statements. These statements are not all-inclusive or exclusive. The phases are very popular in real life, and are extremely necessary in the political field.

The second type of fuzzy quantifier is the existential quantifier. The existential quantifier is defined on [0, 1] and characterize linguistic terms such as almost all, about half. most and so on. This type of quantifier is also called the relative quantifier. Some examples of this type of fuzzy quantifier as follows:

- About half of the girls are tall.
- Almost all of the dogs in the neighborhood bark.
- Most of the graduates will get jobs.

Another important feature of fuzzy logic is the ability to define "hedges." or modifiers of fuzzy values. These operations are provided in an effort to maintain close ties to natural language. and to allow for the generation of fuzzy statements through mathematical calculations. As such. the initial definition of hedges and operations upon them will be quite a subjective process and may vary from one project to another. Nonetheless. the system ultimately derived operates with the same formality as classic logic. The simplest example is in which one transforms the statement "Jane is tall" to "Jane is very tall." The hedge "very" is usually defined as follows:

m"very"A(x) = mA(x)^2

Thus. if mTALL(Jane) = 0.8. then mVERYTALL(Jane) = 0.64 . Other common hedges are "more or less" [typically SQRT(mA(x))]. "somewhat." "rather." "sort of." and so on. Again. their definition is entirely subjective. but their operation is consistent: they serve to transform membership/truth values in a systematic manner according to standard mathematical functions.[29]

All of these concepts discussed are inherent to fuzzy rules. Fuzzy rules or the fuzzy rule base is considered the core of the whole fuzzy system. The fuzzy system is built from three distinct parts: inputs. fuzzy inference. and outputs. The inputs are crisp raw data. which are represented by a membership function. This membership function can be of any shape and is representative of a fuzzy number. There can be one or many inputs depending on the fuzzy system. The first step in a fuzzy system. which transforms all real world data to a degree of membership. is called fuzzification. The second step is the heart of fuzzy systems. fuzzy logic. This intermediate step is where the fuzzy rules for the system are found. The rules give the system its definition and customize the system to the discretion of the user. A process called the fuzzy inference evaluates these rules. One of the most popular fuzzy inference processes is known as Mamdani's fuzzy inference method. It's the most commonly seen method in fuzzy methodology. Mamdani's method was among the first control systems built using fuzzy set theory. Ebrahim Mamdani proposed it in 1975 as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. Mamdani's effort was based on Lotfi Zadeh's 1973 paper on fuzzy algorithms for complex systems and decision processes.

Mamdani-style inference expects the output membership functions to be fuzzy sets. After the aggregation process. there is a fuzzy set for each output variable that needs defuzzification. It's possible. and in many cases much more efficient. to use a single spike as the output membership function rather than a distributed fuzzy set. This is sometimes known as a *singleton* output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required to find the centroid of a two-dimensional shape. Rather than integrating across a continuously varying two-dimensional shape to find the centroid, the weighted average of a few data points can be found. Sugeno systems support this kind of behavior.

The other popular inference method is the Sugeno, or Takagi-Sugeno-Kang method of fuzzy inference first introduced in 1985. It is similar to the Mamdani method in many respects. In fact the first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same.

A typical fuzzy rule in a *zero-order Sugeno fuzzy model* has the form

if x is A and y is B then $z = k$

where A and B are fuzzy sets in the antecedent, while k is a crisply defined constant in the consequent. When the output of each rule is a constant like this, the similarity with Mamdani's method is striking. The only distinctions are that all output membership functions are singleton spikes, and the implication and aggregation methods are fixed and can not be edited. The implication method is simply multiplication and the aggregation operator just includes all of the singletons.[30]

The last step in the fuzzy system is the outputs. This step relates fuzzy numbers and fuzzy variables to crisp numbers. The outputs can be of any number and is the end result of the fuzzy system. The outputs can be classified as a reverse of the inputs when fuzzy systems are considered.

To bring all of these concepts together. the model example is employed for the final time. The example has inputs of model height and model experience in months with an output of model pay per photo shoot. This is not an impossible task for conventional programming. But first one must find a model for the system that associates model height and experience with model pay. This would be a somewhat complex program and the vagueness of height and experience makes the task even more difficult. A program named MATLAB is implemented to overcome such a programming obstacle.

MATLAB is an integrated technical computing environment that combines numeric computation. advanced graphics and visualization. and a high-level programming language. The extensive and powerful numeric computing methods and graphics allows testing and exploring alternative ideas easily. while the integrated development environment makes it easy to produce fast. practical results.

The name MATLAB stands for matrix laboratory. MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. which together represent the state of the art in software for matrix computation. Today MATLAB is used in a variety of application areas including signal and image processing. control system design. financial engineering. and medical research. The open architecture makes it easy to use MATLAB and companion products to explore data and create custom tools that provide early insights and competitive advantages.

MATLAB also features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB. toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment in order to solve particular classes of problems. Researched and developed by experts in their fields. toolboxes allows a user to learn. apply. and compare best-of-class techniques. allowing him to evaluate different approaches without writing the code.

Areas in which toolboxes are available include signal processing. control systems design. dynamic systems simulation. fuzzy logic. systems identification. neural networks. and others.

The Fuzzy Logic Toolbox features a simple point-and-click interface that guides a user effortlessly through the steps of fuzzy design. from setup to diagnosis. It provides built-in support for the latest fuzzy logic methods. such as fuzzy clustering and adaptive neuro-fuzzy learning. The Toolbox's interactive graphics allows a user to instantly visualize and fine tune system behavior.[25]

The fuzzy toolbox is constructed in a GUI format. which allows the user to view different parts of the design at the same time. The FIS (Fuzzy Inference System) editor is the main window which all other windows are linked. In this window all inputs and outputs are created and named. This window is shown in Figure 8 – 2. Once the inputs are created and named. each input and output must be defined to fit the particular system. Definitions for the model example inputs and output are shown in Figures 8 – 3 through 8 – 5. The inputs are made of triangular and trapezoidal functions. From these membership functions rules for the system must be defined to satisfy the preferences of the user. The rules are in a verbal if-then format. which allows fast manipulation and creation. The rules can also be defined in other formats such as symbolic and indexed. The symbolic doesn't consider the if-then part of the verbose format; it is simply implied. The indexed version. which is shown in Listing 8 – 1. is the numerical equivalent of the rules in the way the computer interprets said rules. The rules for the model example are shown in Figure 8 – 6. The 1 followed by every rule is known as the rule weight. This weight may vary from 0 to 1 and the default is a weight of 1. The rule weight allows the user to prioritize rules of importance from rule that are not that critical to system performance. After all membership functions for input and outputs and rules are defined. a program listing is generated. This program listing is what the computer uses to obtain all calculations for the system. The program listing for the model example is shown in Listing 8 – 1.

To this point all the work has been done for the creation of the fuzzy system concerning the model example. The fuzzy system does not only simplify the creation but the modification of the program as well. When conventional programs are constructed. many lines are invoked to get the output the user wants. Years in the future. when the program has become deficient and needs to be reworked. the task of relearning what transpired very long ago is now at hand. But with a fuzzy logic program this is not a problem because the code is in plain and simple English statements. Conventional programs can be commented to make it easier for later modification. but this technique does not compare to the fuzzy logic rule base.

MATLAB outputs several figures when the rules and membership functions are defined correctly. The first figures which MATLAB outputs is the model height. model experience. and model pay. These figures are shown in Figures 8 – 7 through 8 – 9. These figures are the same as the membership function plots located in the membership function editor found in Figures 8 – 3 through 8 – 5. The next output generated by MATLAB is the rule viewer. The rule viewer allows the user to input different input values and receives a graphical solution. There are three different inputs selected randomly to convey the range and robustness of the fuzzy system. The solutions for these three different inputs are shown in Figures 8 – 10 through 8 – 12. The final two figures that the fuzzy toolbox outputs are the system plot and the surface view. These two outputs are pictured in Figures 8 – 13 and 8 – 14, respectively. The system plot gives the shape of the input and output membership functions and displays the system in a block diagram format. The surface view generates a three dimensional picture of the system. This picture can be manipulated by rotation and scaling. The picture is very useful when there are several inputs and the user needs to see the relations the inputs have to the output.

As mentioned previously in this chapter. there are two different techniques of defuzzification that can be used in the fuzzy logic toolbox. The Mamdani style was previously used in the model example. But to get a greater understanding of the capabilities of the fuzzy logic toolbox. the Sugeno style is now utilized for the model example. The FIS editor for this style is shown in Figure 8 – 15. There is no difference between the Sugeno and the Mamdani style when inputs are concerned. But there is no output membership functions defined by the user. There are only levels of membership referred to the output. which can be defined by the user. The output membership of the model example is shown in Figure 8 – 16. The rules that define the system are the same as the Mamdani style. Since the output and the style are different. so is the program listing. The fuzzy logic program is shown in Listing 8 – 2. Comparing the rule view for the two different styles is done by inputting the same height and experience values. The output for the rule view considering the Sugeno method is shown in Figures 8 – 17 through 8 – 19. A comparison of the two styles is found in a compressed form in Table 8 – 2. This table lists the inputs and outputs that are found in the rule view for the two styles. The surface view is the last figure associated with the Sugeno method. While this figure is very similar to Figure 8 – 14 there is one difference. The surface view in Figure 8 – 14 is coarser that the one in 8 – 20. Also. the ranges of the model pay are greater in the Segeno method. This is attributed to the output membership values. A

Table 8 – 2: Comparison between Mamdani and Sugeno Styles for Model Example

Model Height, Experience	Model Pay (Mamdani)	Model Pay (Sugeno)
4.6. 24	668	108
5.8. 21	1.580	2400
6. 37	3.490	4500

similar output could be obtained by the Mamdani method by output membership manipulation. Other advantages of each method are as follows:

Advantages of Sugeno's method

- Computational efficiency
- Guaranteed continuity of the output surface
- Better suited to mathematical analysis

Advantages of Mamdani's method

- More intuitive
- Wide spread acceptance
- Better suited to human input

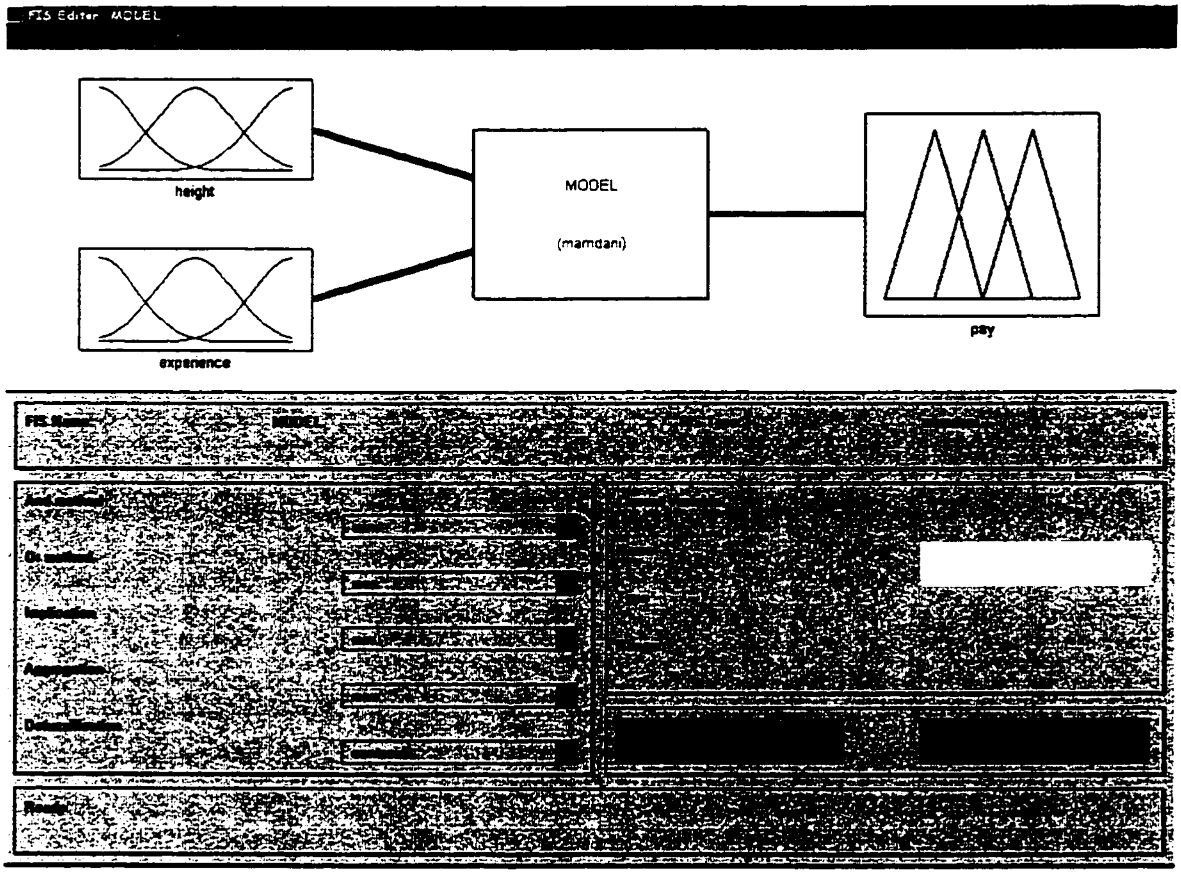
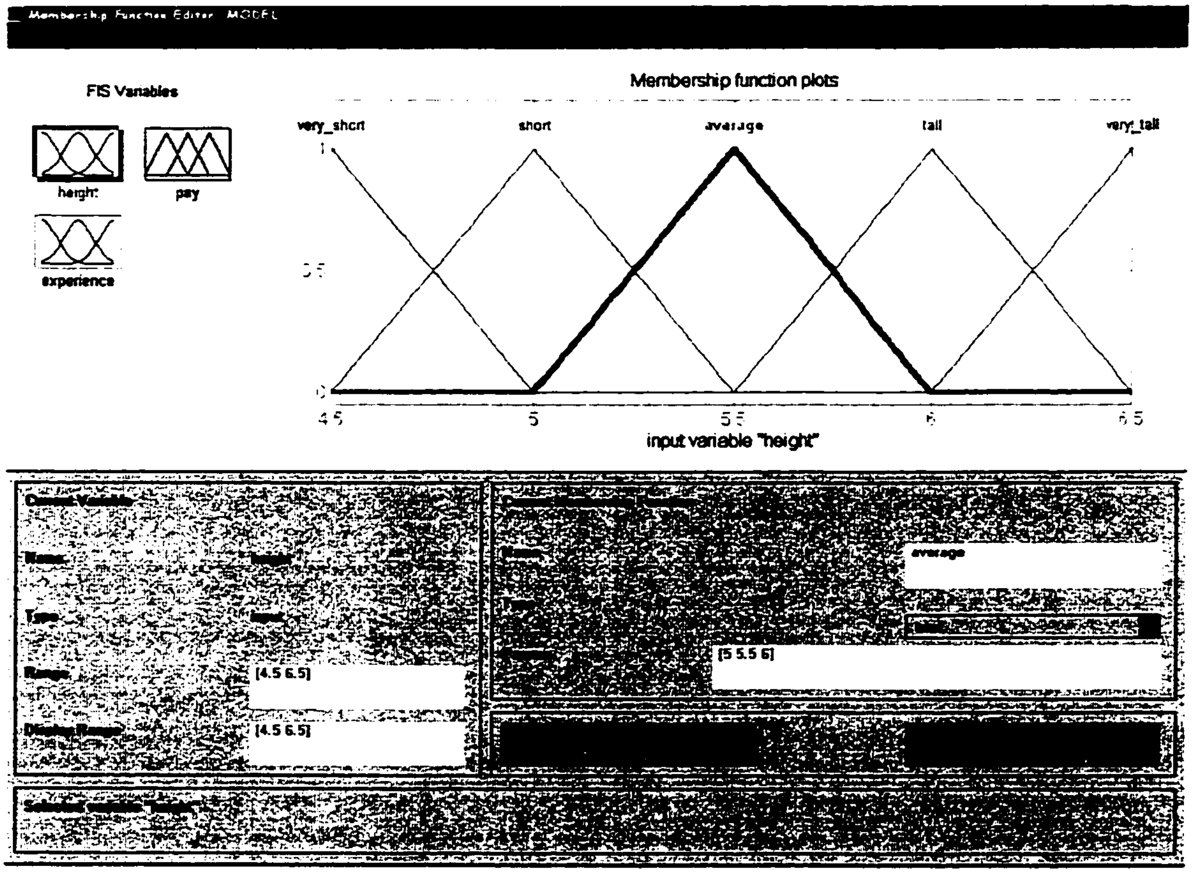


Figure 8 - 2 : FIS Editor for Model Example



Current Variable

Name: height

Type: input

Range: [4.5 6.5]

Display Range: [4.5 6.5]

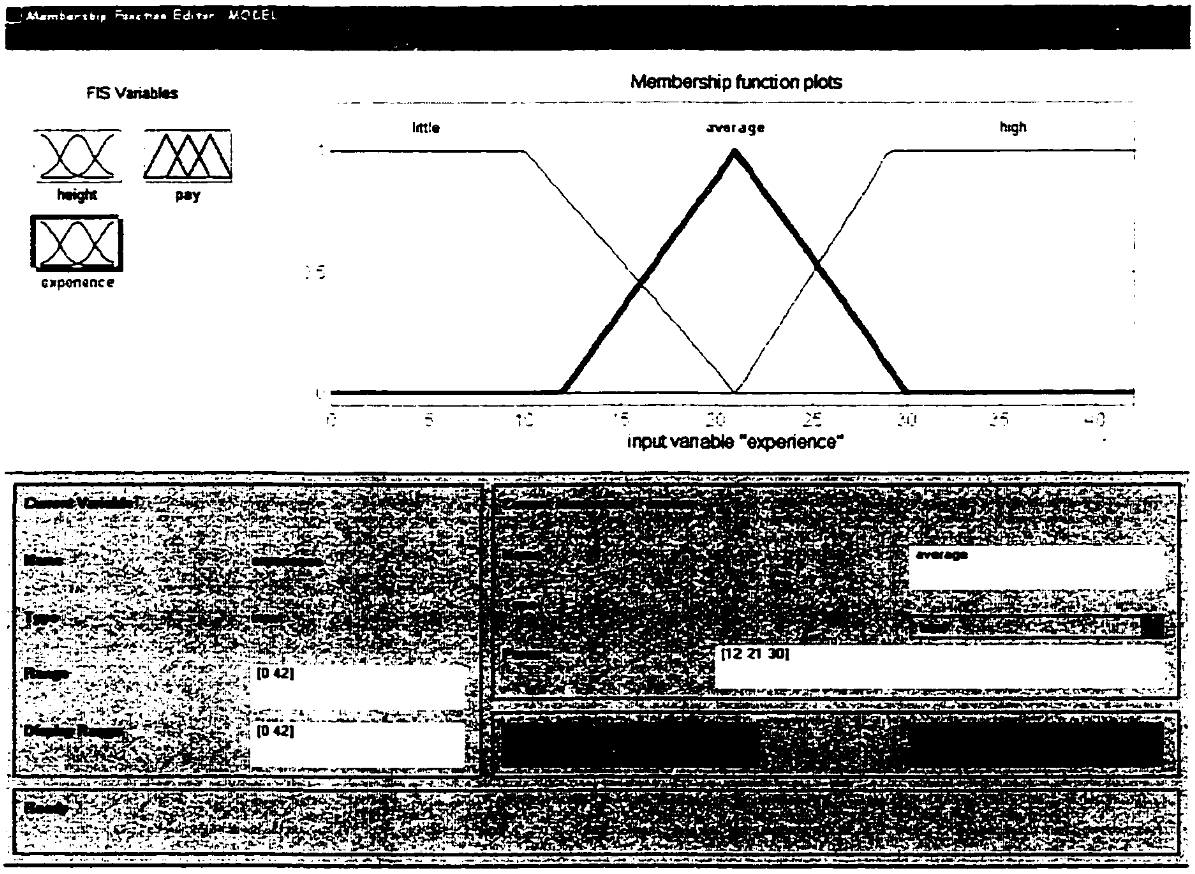
Current Membership Function

Name: average

Params: [5 5.5 6]

Selected variable "height"

Figure 8 - 3 : Membership Function Editor for Height Input of Model Example


Membership Function Editor: MODEL
FIS Variables
height
pay
experience
Membership function plots
little
average
high
1
0.5
0
0
5
10
15
20
25
30
35
40
input variable "experience"

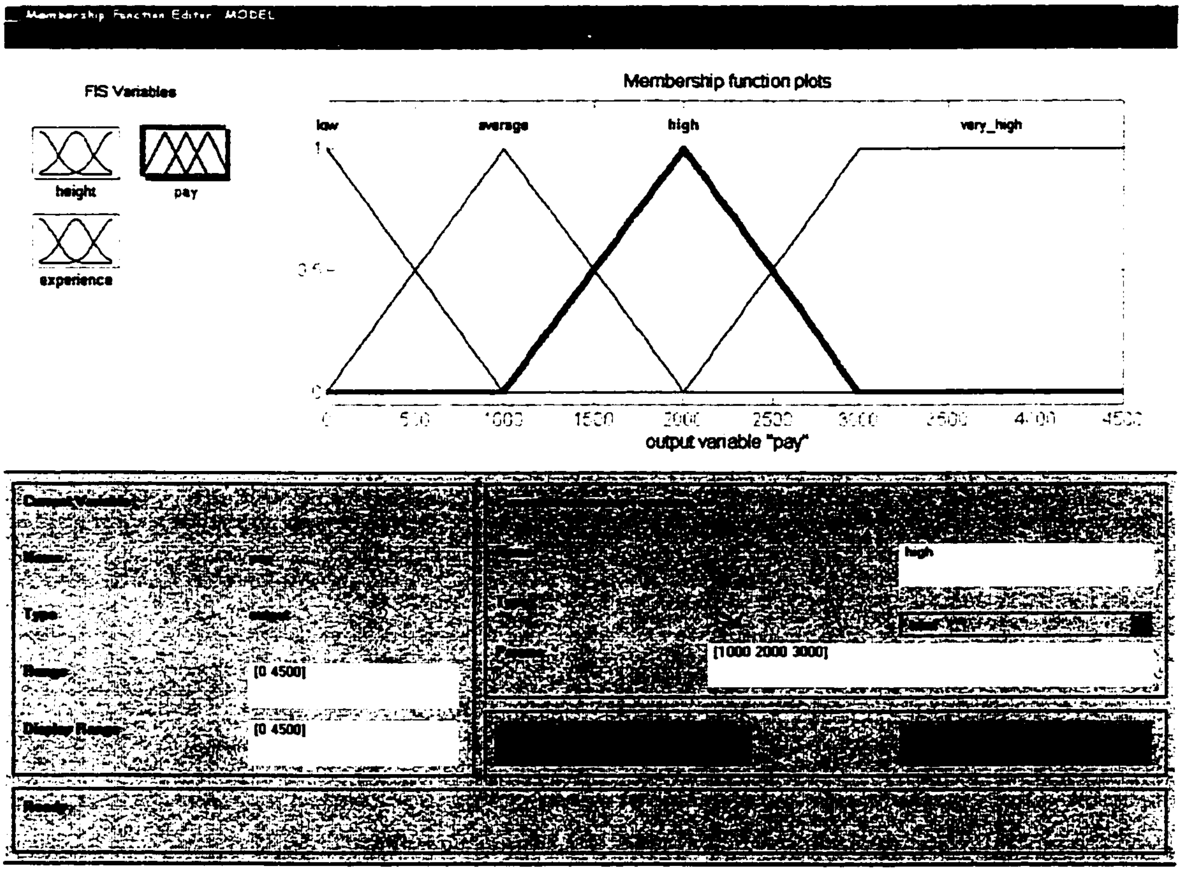
[0 42]

[0 42]

average

[12 21 30]

Figure 8 - 4 : Membership Function Editor for Experience Input of Model Example



Membership Function Editor: MODEL

Current Variable

Name: pay

Type: output

Range: [0 4500]

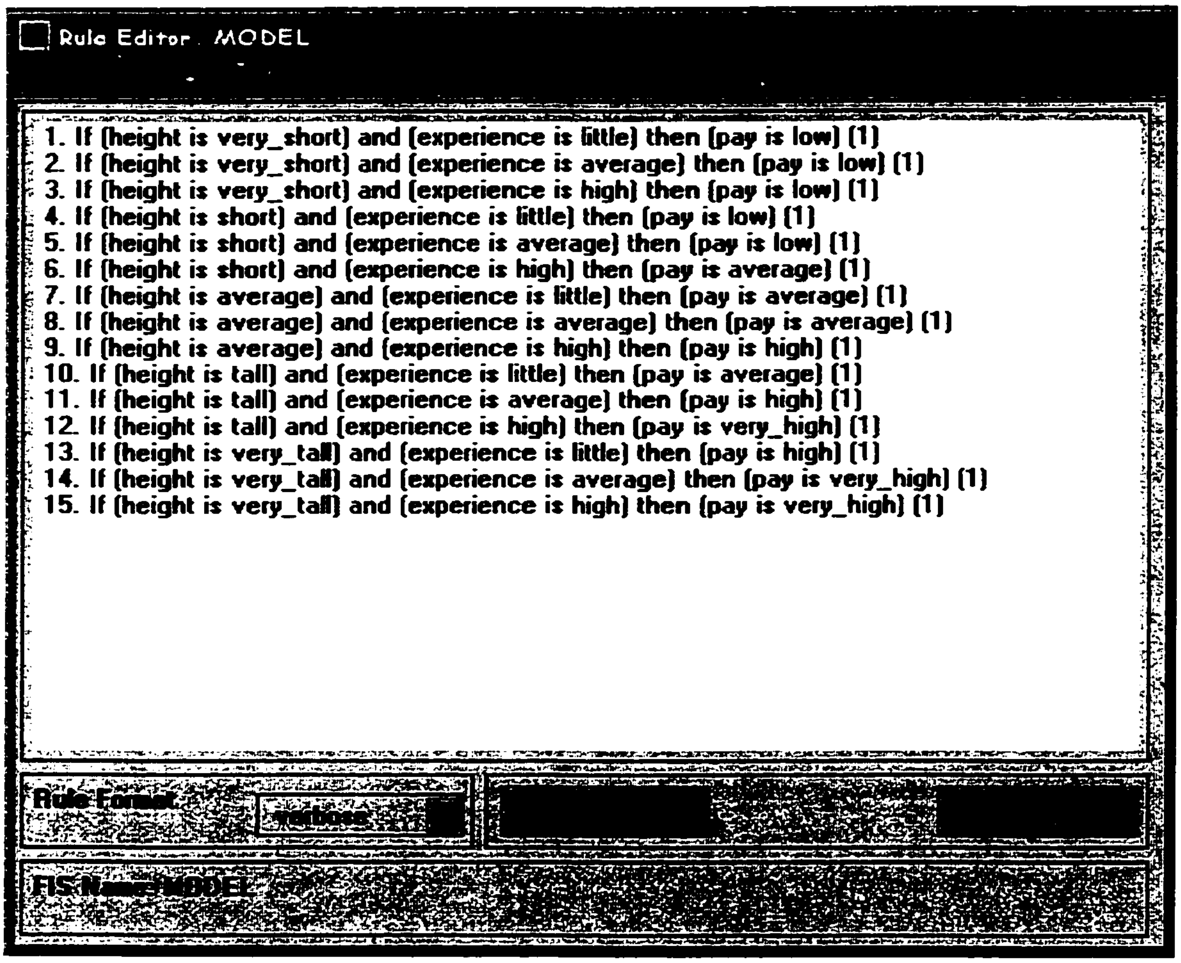
Display Range: [0 4500]

Name: high

Params: [1000 2000 3000]

Ready

Figure 8 - 5 : Membership Function Editor for Pay Output of Model Example



1. If (height is very_short) and (experience is little) then (pay is low) (1)
2. If (height is very_short) and (experience is average) then (pay is low) (1)
3. If (height is very_short) and (experience is high) then (pay is low) (1)
4. If (height is short) and (experience is little) then (pay is low) (1)
5. If (height is short) and (experience is average) then (pay is low) (1)
6. If (height is short) and (experience is high) then (pay is average) (1)
7. If (height is average) and (experience is little) then (pay is average) (1)
8. If (height is average) and (experience is average) then (pay is average) (1)
9. If (height is average) and (experience is high) then (pay is high) (1)
10. If (height is tall) and (experience is little) then (pay is average) (1)
11. If (height is tall) and (experience is average) then (pay is high) (1)
12. If (height is tall) and (experience is high) then (pay is very_high) (1)
13. If (height is very_tall) and (experience is little) then (pay is high) (1)
14. If (height is very_tall) and (experience is average) then (pay is very_high) (1)
15. If (height is very_tall) and (experience is high) then (pay is very_high) (1)

Figure 8 - 6 : Rule Editor for Model Example

```
[System]
Name='MODEL'
Type='mamdani'
NumInputs=2
NumOutputs=1
NumRules=15
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'
[Input1]
Name='height'
Range=[4.5 6.5]
NumMFs=5
MF1='short':'trimf',[4.5 5 5.5]
MF2='average':'trimf',[5 5.5 6]
MF3='tall':'trimf',[5.5 6 6.5]
MF4='very_short':'trapmf',[4.5 4.5 4.5 5]
MF5='very_tall':'trimf',[6 6.5 8.5]
[Input2]
Name='experience'
Range=[0 42]
NumMFs=3
MF1='little':'trapmf',[0 0 10 21]
MF2='average':'trimf',[12 21 30]
MF3='high':'trapmf',[21 29 42 42]
[Output1]
Name='pay'
Range=[0 4500]
NumMFs=4
MF1='low':'trimf',[0 0 1000]
MF2='average':'trimf',[0 1000 2000]
MF3='high':'trimf',[1000 2000 3000]
MF4='very_high':'trapmf',[2000 3000 4500 4500]
[Rules]
4 1, 1 (1) : 1
4 2, 1 (1) : 1
4 3, 1 (1) : 1
1 1, 1 (1) : 1
1 2, 1 (1) : 1
1 3, 2 (1) : 1
2 1, 2 (1) : 1
2 2, 2 (1) : 1
2 3, 3 (1) : 1
3 1, 2 (1) : 1
3 2, 3 (1) : 1
3 3, 4 (1) : 1
5 1, 3 (1) : 1
5 2, 4 (1) : 1
5 3, 4 (1) : 1
```

Listing 8 - 1 : Program Listing for Model Example

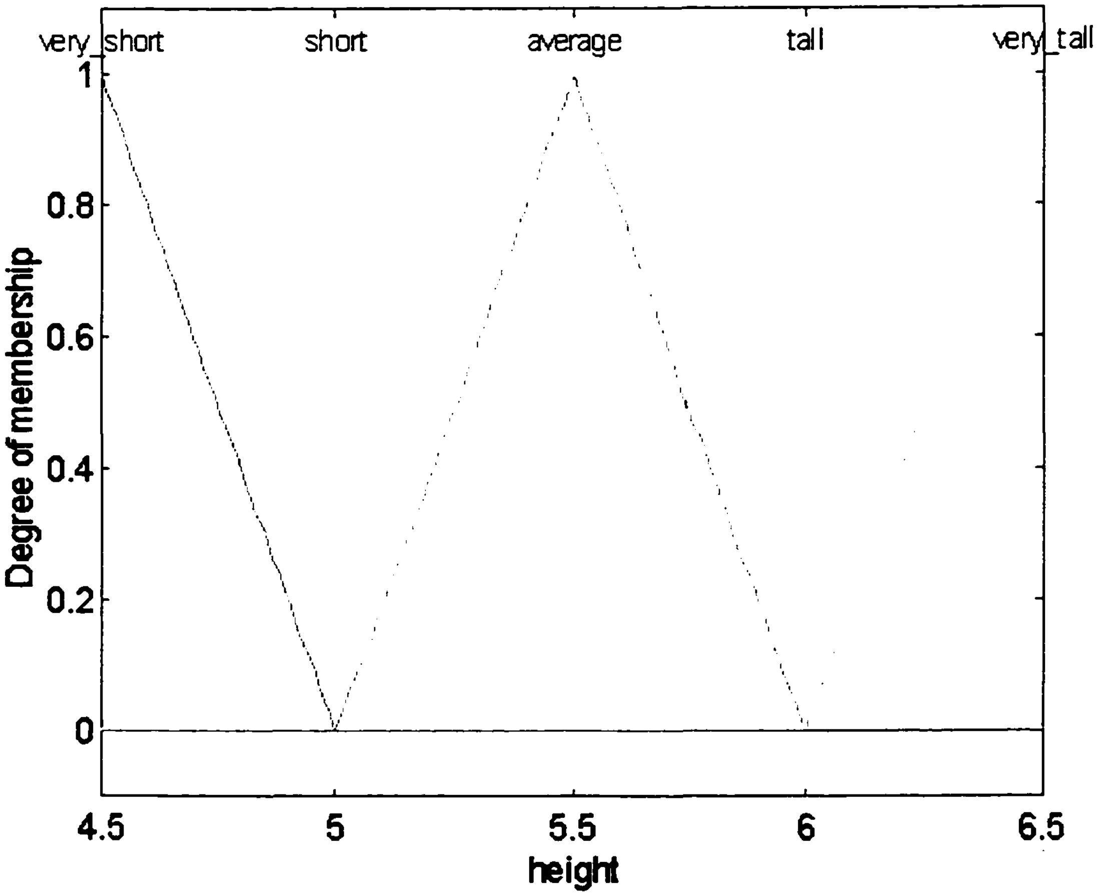


Figure 8 - 7 : MATLAB Output for Model Height Input

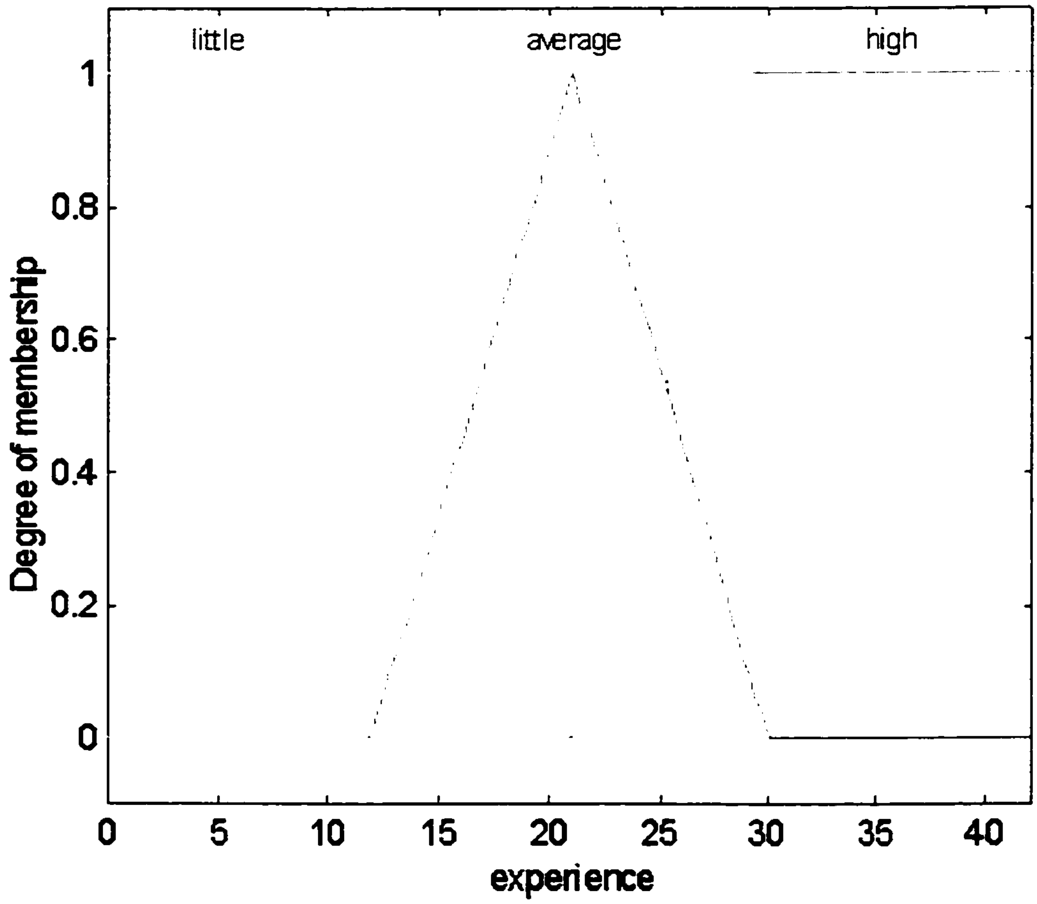


Figure 8 - 8 : MATLAB Output for Model Experience Input

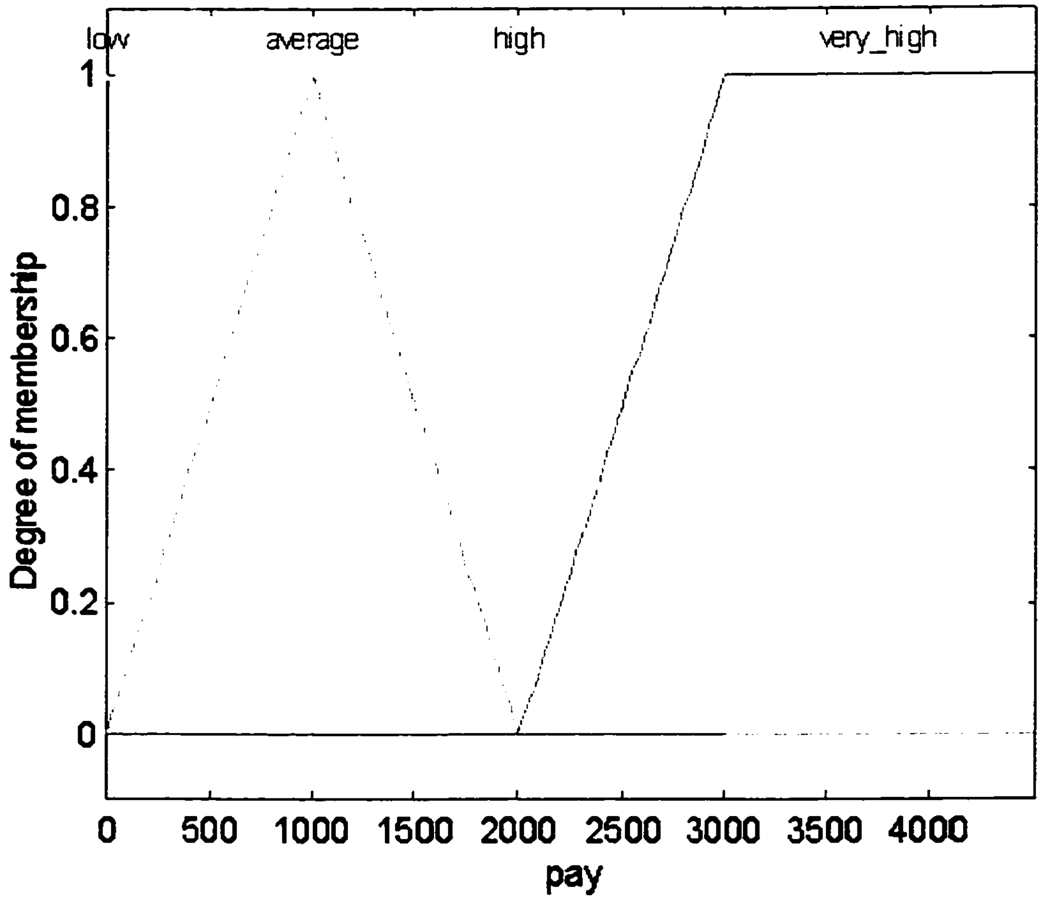
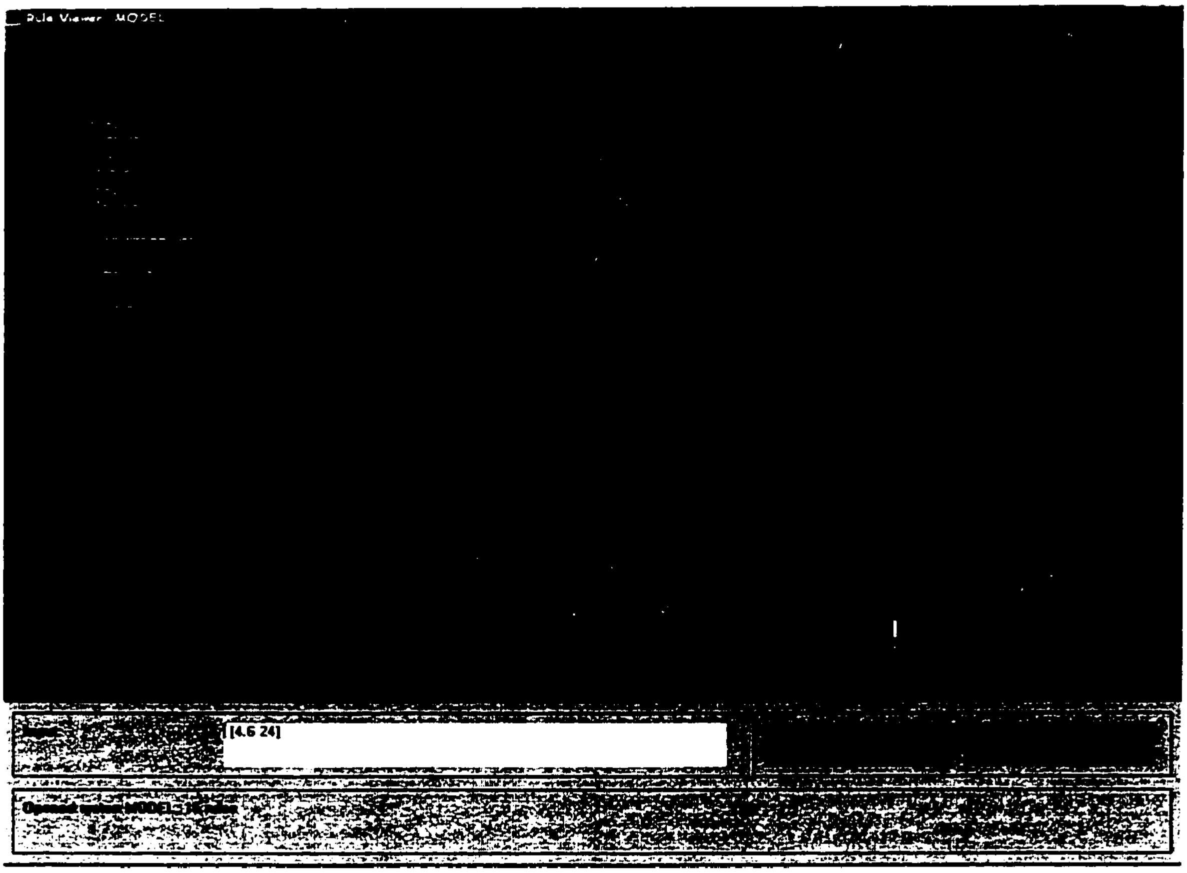


Figure 8 - 9 : MATLAB Output for Model Pay Output



[4.6 24]

Figure 8 - 10 : Rule Viewer for Model Example

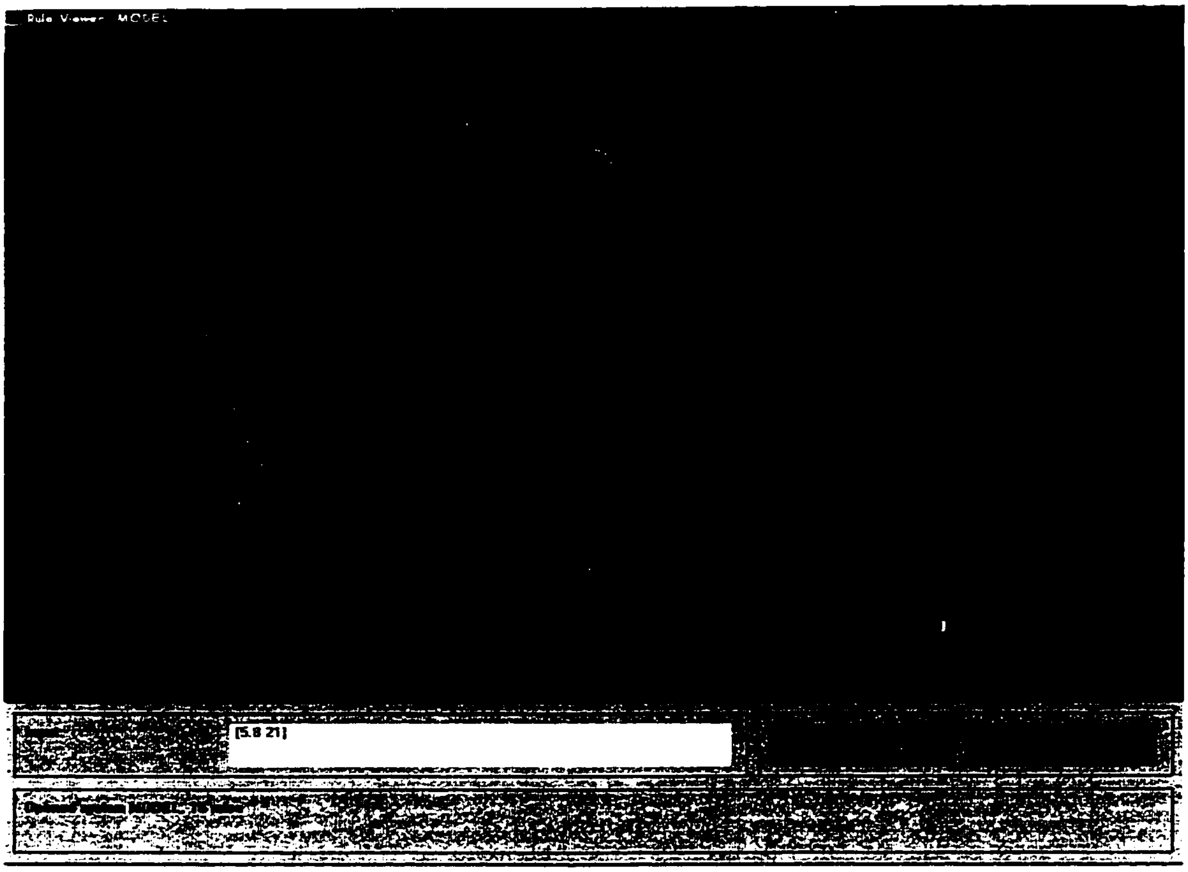


Figure 8 - 11 : Rule Viewer for Model Example

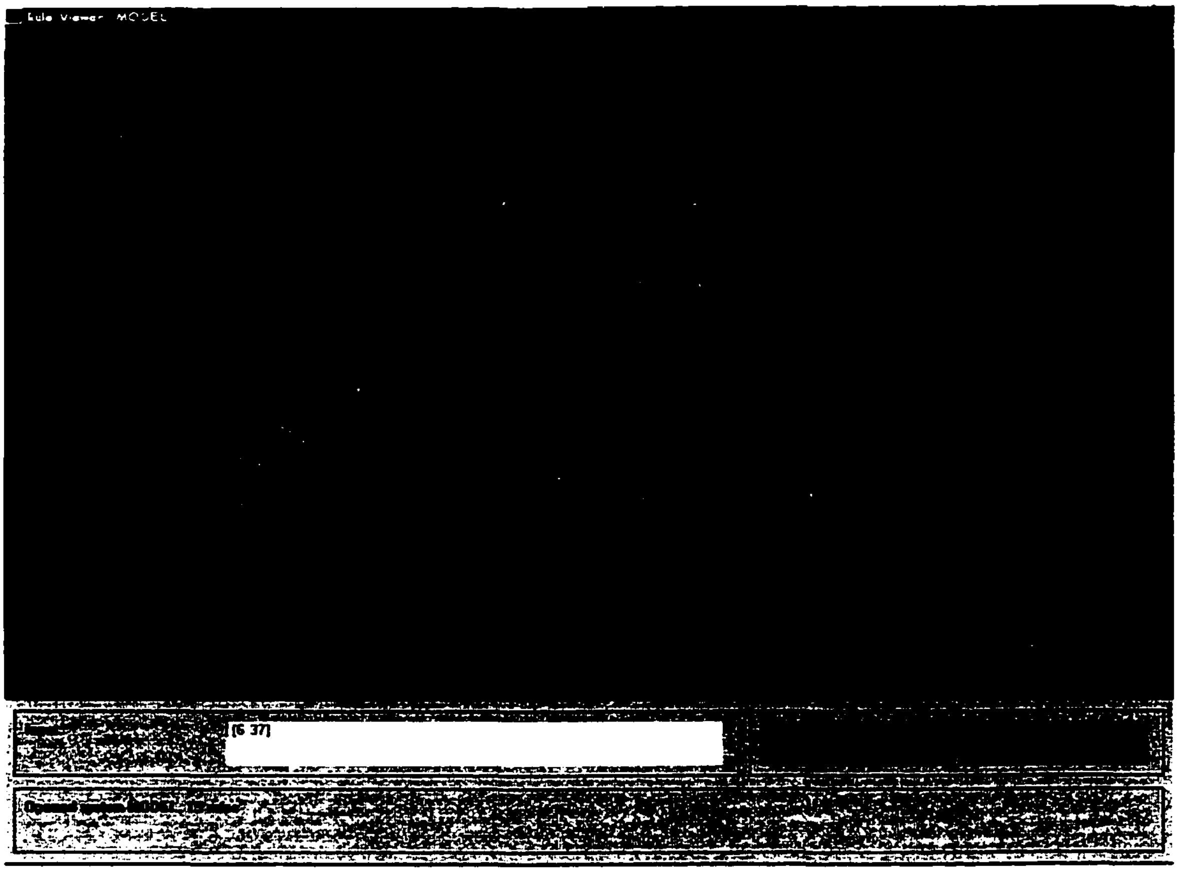


Figure 8 - 12 : Rule Viewer for Model Example

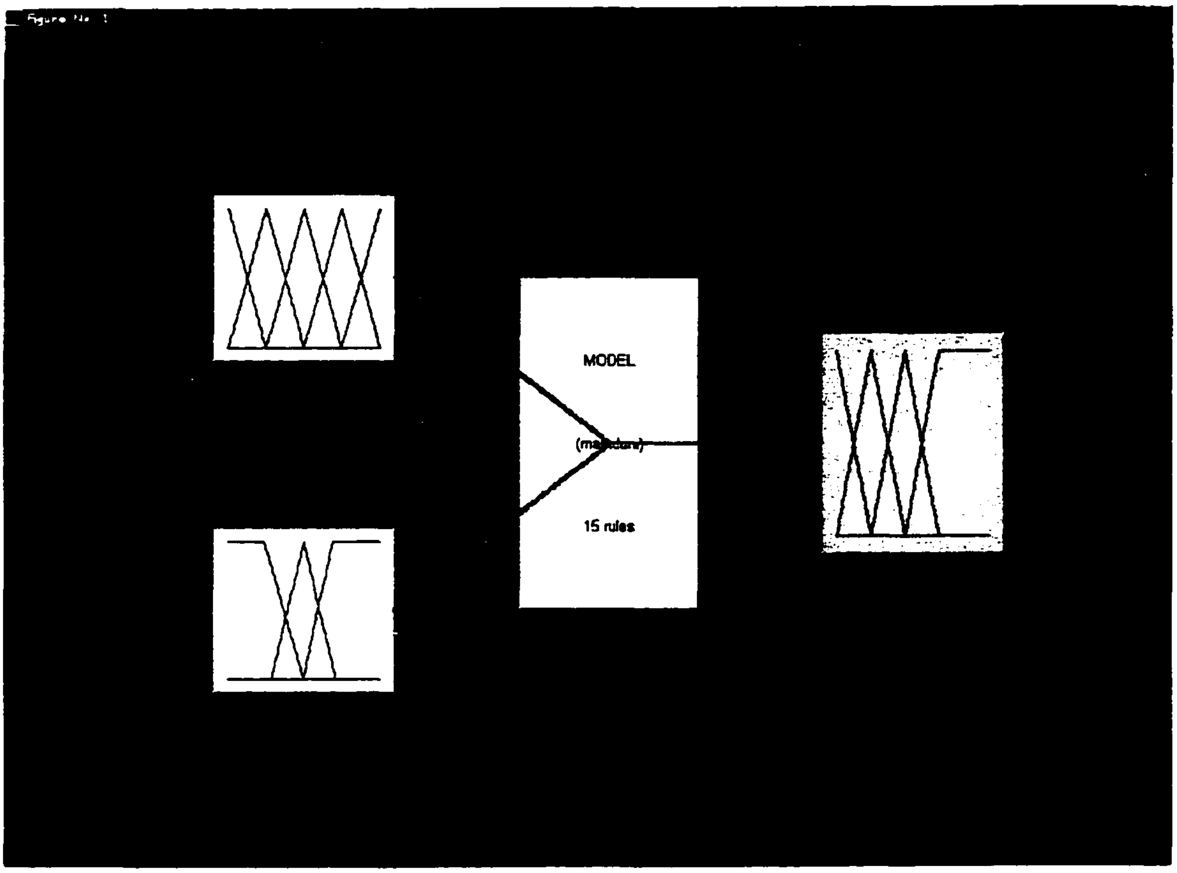


Figure 8 - 13 : System Plot for Model Example

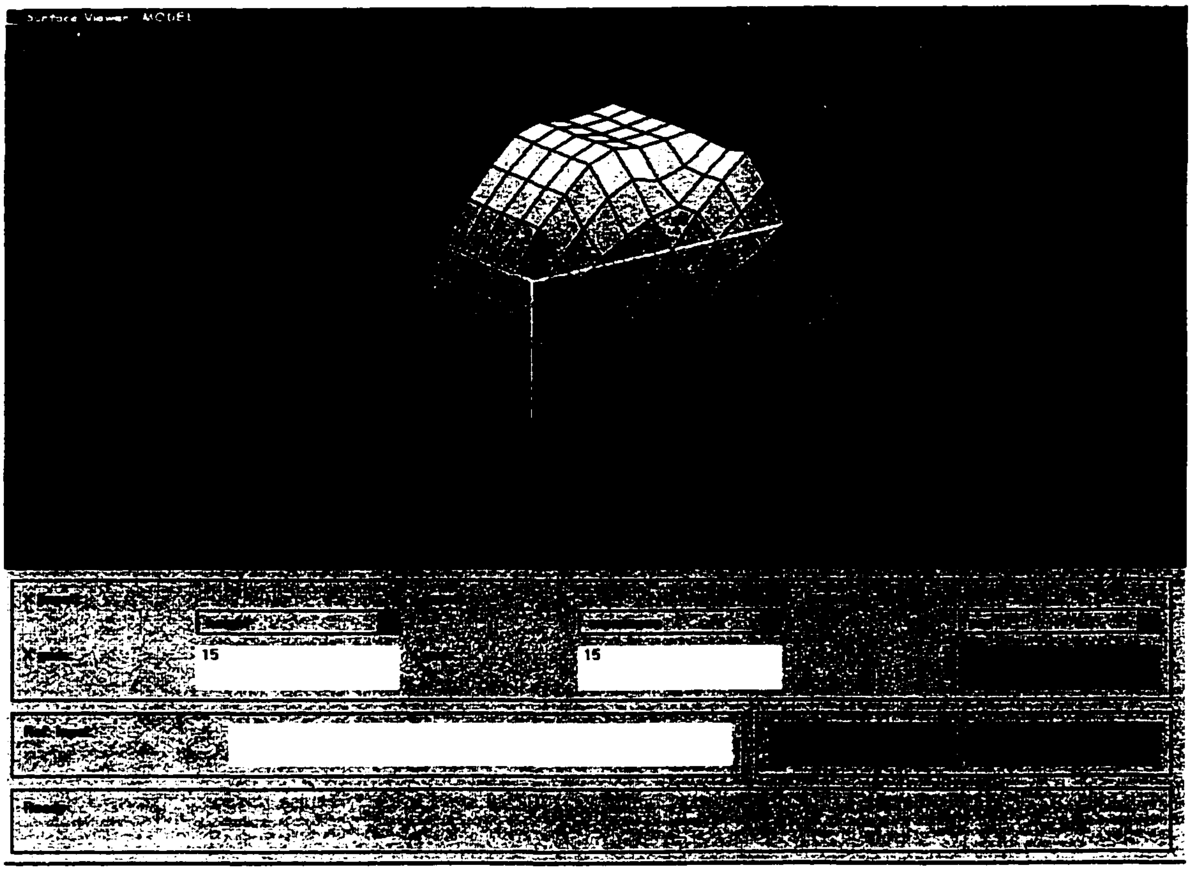


Figure 8 - 14 : Surface View for Model Example

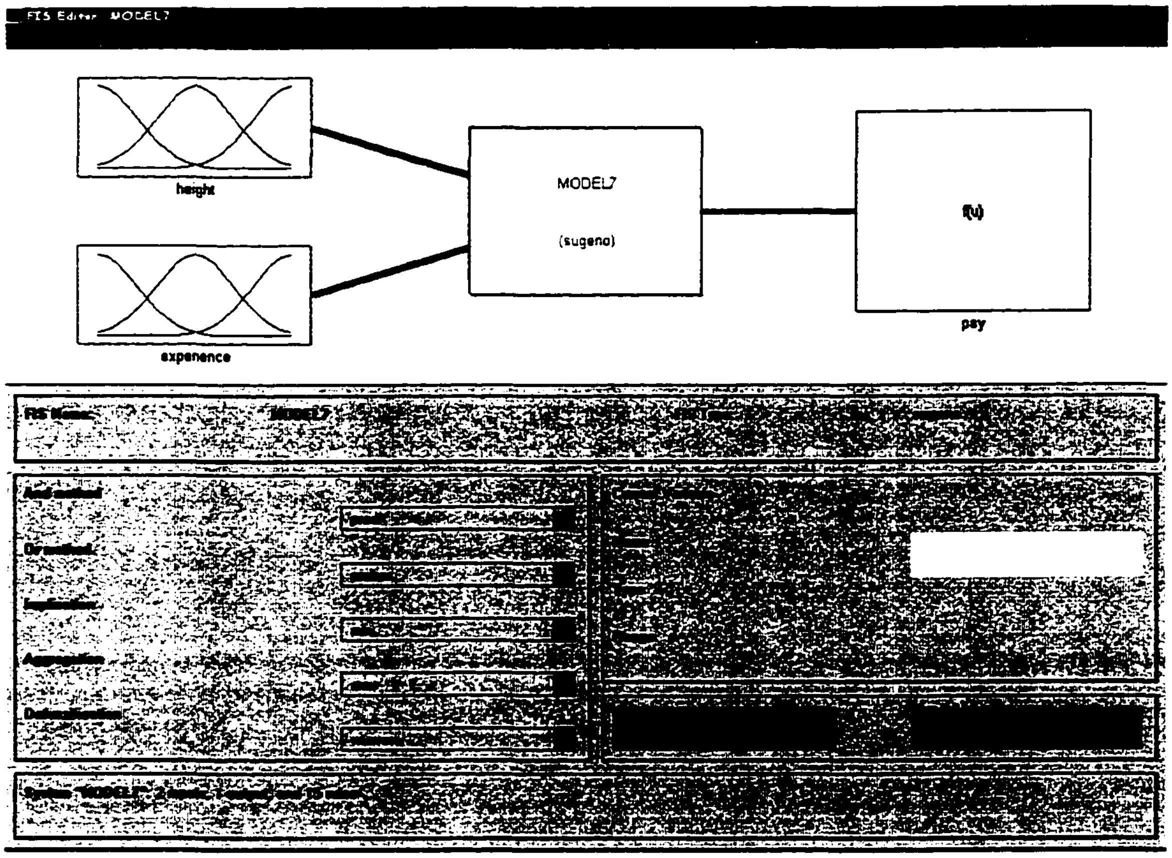
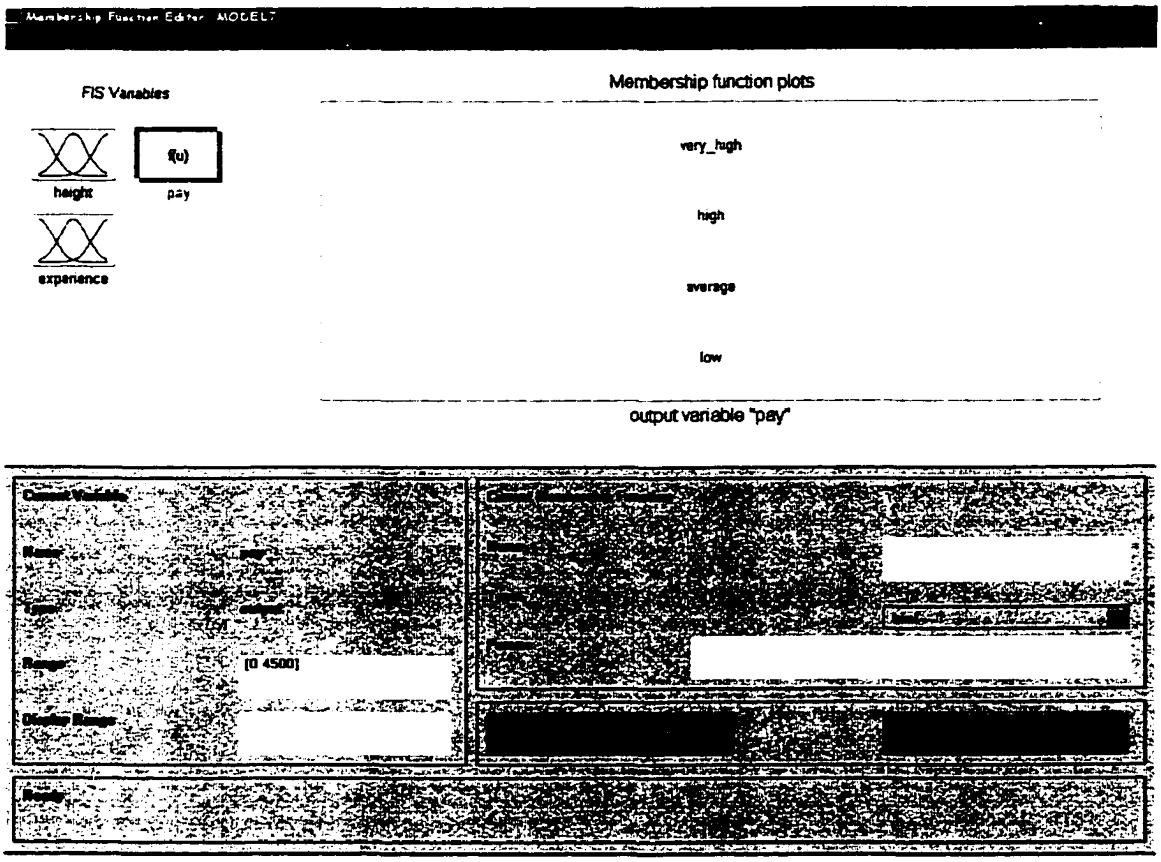


Figure 8 - 15 : FIS Editor for Model Example (Sugeno Style)



Current Variable

Name pay

Type output

Range [0 4500]

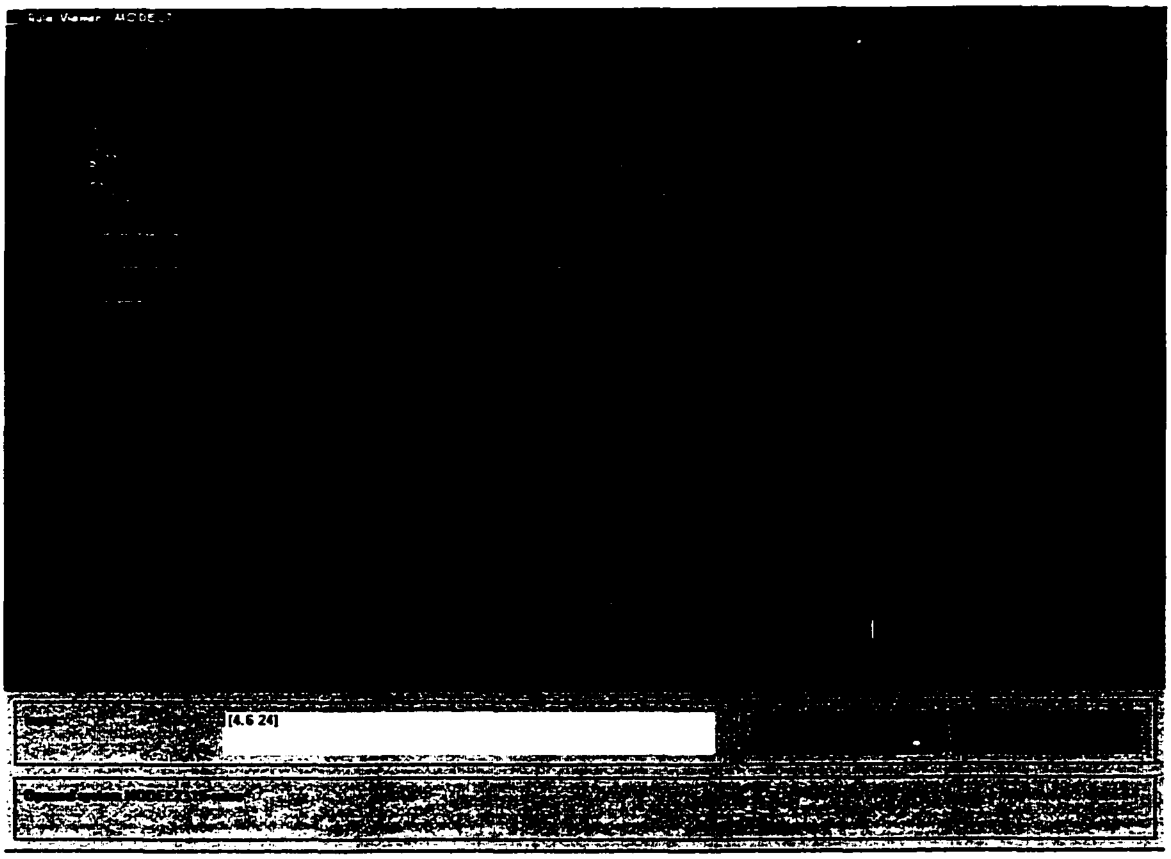
Display Range

Ready

Figure 8 - 16 : Membership Function Editor for Pay Output of Model Example (Sugeno Style)

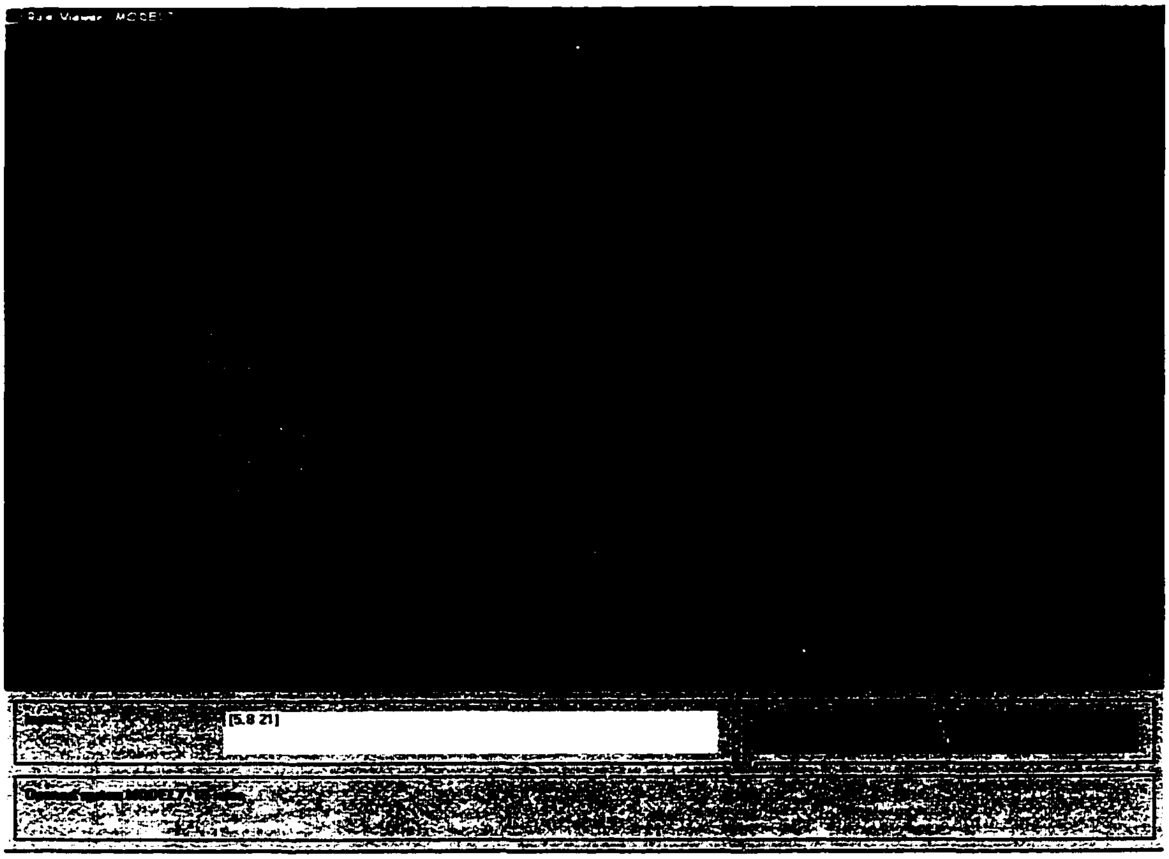
```
[System]
Name='MODEL7'
Type='sugeno'
NumInputs=2
NumOutputs=1
NumRules=15
AndMethod='prod'
OrMethod='probor'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='wtaver'
[Input1]
Name='height'
Range=[4.5 6.5]
NumMFs=5
MF1='short':'trimf',[4.5 5 5.5]
MF2='average':'trimf',[5 5.5 6]
MF3='tall':'trimf',[5.5 6 6.5]
MF4='very_short':'trapmf',[4.5 4.5 4.5 5]
MF5='very_tall':'trimf',[6 6.5 8.5]
[Input2]
Name='experience'
Range=[0 42]
NumMFs=3
MF1='little':'trapmf',[0 0 10 21]
MF2='average':'trimf',[12 21 30]
MF3='high':'trapmf',[21 29 42 42]
[Output1]
Name='pay'
Range=[0 4500]
NumMFs=4
MF1='low':'constant',0
MF2='average':'constant',1500
MF3='high':'constant',3000
MF4='very_high':'constant',4500
[Rules]
4 1, 1 (1) : 1
4 2, 1 (1) : 1
4 3, 1 (1) : 1
1 1, 1 (1) : 1
1 2, 1 (1) : 1
1 3, 2 (1) : 1
2 1, 2 (1) : 1
2 2, 2 (1) : 1
2 3, 3 (1) : 1
3 1, 2 (1) : 1
3 2, 3 (1) : 1
3 3, 4 (1) : 1
5 1, 3 (1) : 1
5 2, 4 (1) : 1
5 3, 4 (1) : 1
```

Listing 8 - 2 : Program Listing for Model Example (Sugeno style)



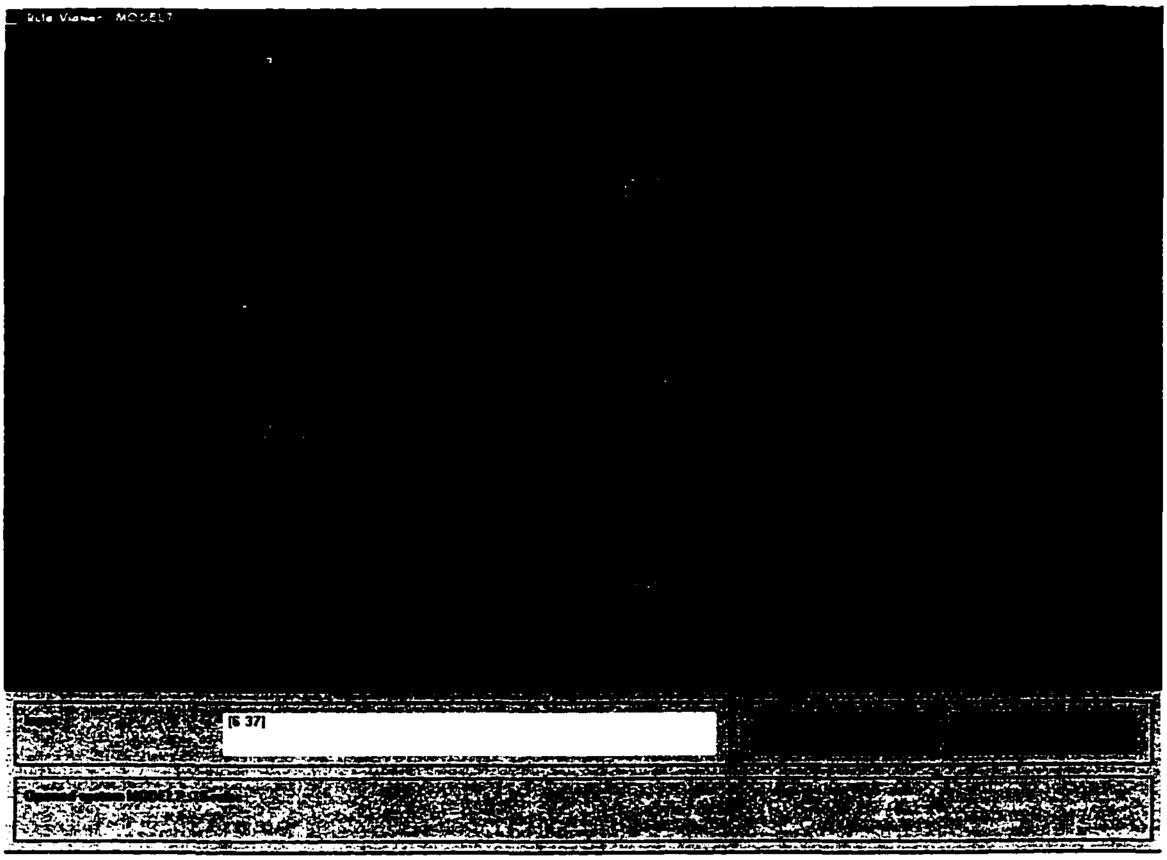
[4.6 24]

Figure 8 - 17 : Rule View for Model Example (Sugeno Style)



[5.8 21]

Figure 8 - 18 : Rule View for Model Example (Sugeno Style)



Rule Viewer: MODEL7

[6 37]

Figure 8 - 19 : Rule View for Model Example (Sugeno Style)

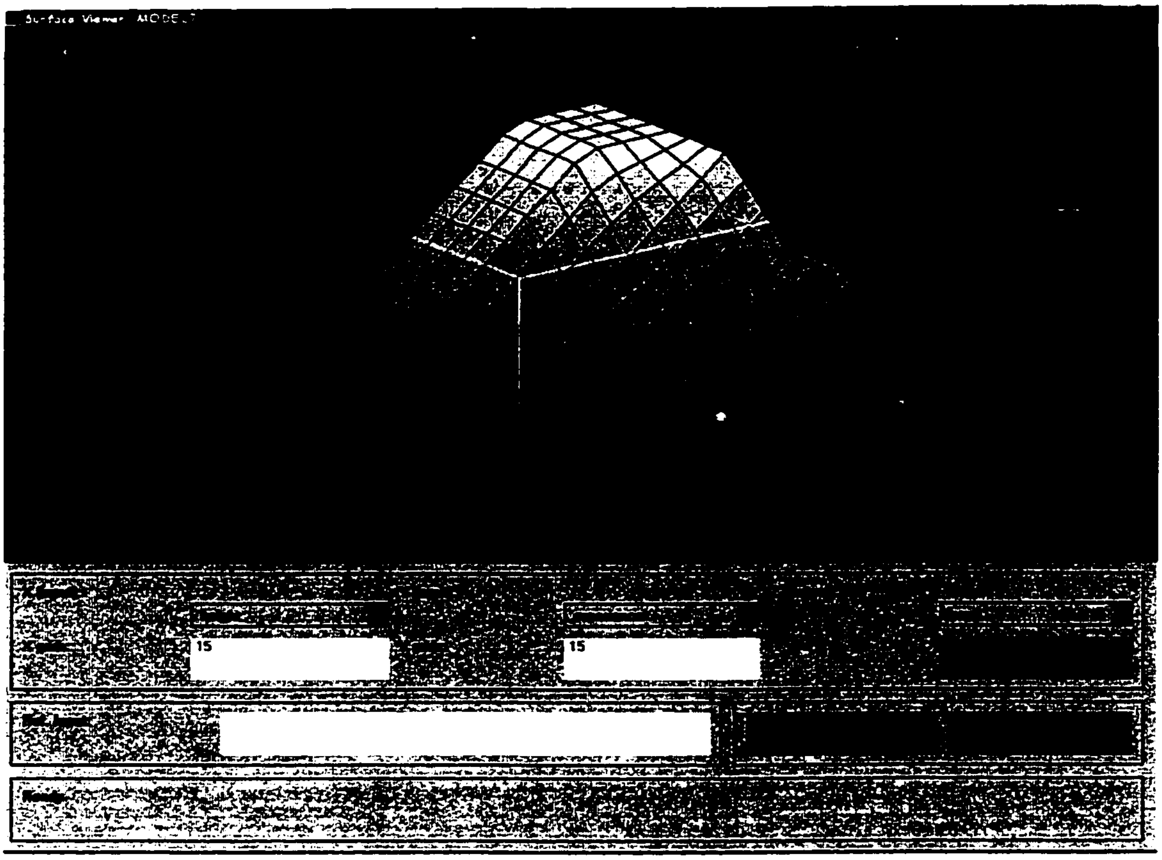


Figure 8 - 20 : Surface View for Model Example (Sugeno Style)

CHAPTER IX

OVERVIEW OF FUZZY CONTROL AND NEUROFUZZY SYSTEMS

Fuzzy systems are exploited in many different areas in engineering and science. These areas have been furthered in many ways by the introduction of fuzzy logic. Some of these areas that have been revolutionized by fuzzy logic are control systems, neural networks, expert systems, and others. Two of these areas shall be investigated more closely in this chapter. These areas have taken on new names, fuzzy control and neurofuzzy systems.

Fuzzy control is a way to transform knowledge into control laws. Traditional control is an arrangement of physical components connected or related in such a manner as to command, direct, or regulate itself or another system. [10] In engineering, control is usually restricted to only apply to those systems whose major function is to dynamically or actively command, direct or regulate. Control systems are made up of two main defining divisions that identify with the system, input and output. The input is the stimulus or excitation applied to a control system from an external energy source, usually in order to produce a specified response from the control system. The output is the actual response obtained from a control system. It may or may not be equal to the specified response implied by the input. From these inputs and output, the control system is formed. There are two types of control systems, open loop and closed loop. The open loop control system is one in which the control action is independent of the output. A closed loop system is one that the control action is somehow dependent on the output. A main characteristic of a closed loop system is feedback. Feedback is a property of a closed loop control system, which permits the output to be compared with the input of the system so that the appropriate control action may be taken. This appropriate action is dictated by some function for the output and input. The open and closed loop control systems are shown in Figures 9 – 1 and 9 – 2, respectively.

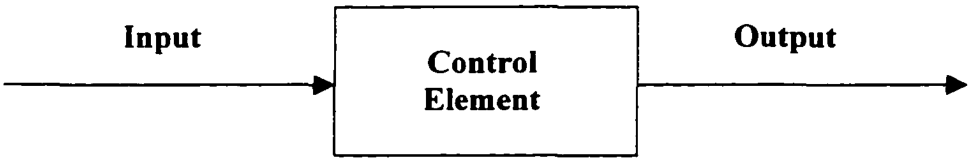


Figure 9 – 1 : Open Loop Control System

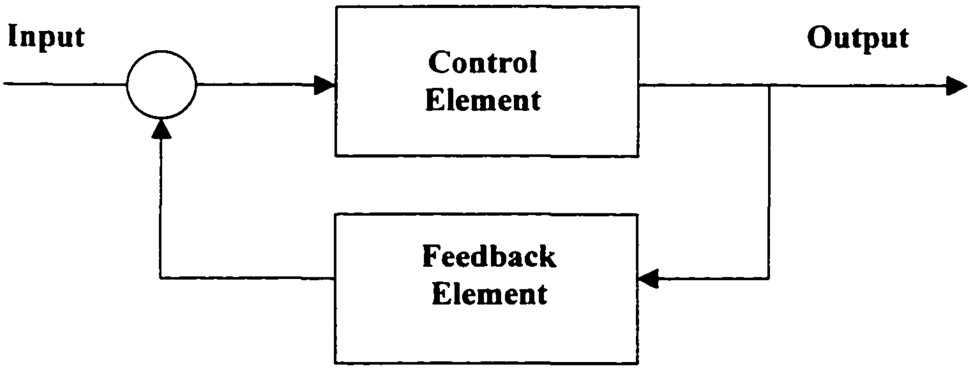


Figure 9 – 2 : Closed Loop Control System

This is by no means a detailed look at control theory, but a basic introduction of control systems. The drawback of conventional control systems that allow fuzzy control to flourish must also be mentioned. This drawback is contained in the system control element found in Figures 9 – 1 and 9 – 2 and is known as the mathematical model. Mathematical models, in the form of system equations, are employed to contend with detailed relationships within the system control element. In theory, every system may be categorized by a system equation.[8]

In the application of fuzzy control, the following stages can be distinguished: matching of data with rule premises (includes fuzzification), determination of degrees of fulfillment for the rules, aggregation of results of individual rules, and defuzzification to obtain a numerical controller output. A fuzzy controller as such is a mapping; it can be shown that the mapping is characterized by tuples in a hyperspace and each tuple represents a fuzzy rule. The fuzzy reasoning performs an interpolation between these tuples in that hyperspace, resulting in a (non)linear input-output mapping.

If the nonlinearity of the mapping should be defined by the fuzzy rules, the following choices should be made: product for conjunction, summation for disjunction and aggregation, fuzzy-mean defuzzification, fuzzy partitions on the input universes and a triangular norm for the implication.

Several adaptive fuzzy controllers have been studied, such as self-organizing controllers, fuzzy associative memories, fuzzy neural networks and fuzzy supervisors.

Fuzzy control can be regarded as only a small part of a much broader framework of approximate reasoning and possibility theory. Approximate reasoning provides a method for modeling human classification and reasoning. There is a lack of practical applicability due to severe calculational effort and memory requirements to perform reasoning according to the theory of approximate reasoning.

In many cases, a fuzzy controller can be simplified to a look-up table and an interpolation method to provide the fuzzy inference. Hence, these simplifications can reduce the fuzzy aspect of fuzzy control to a user-interfacing concept during the design stage. The architecture of a fuzzy controller is shown in Figure 9 – 3.

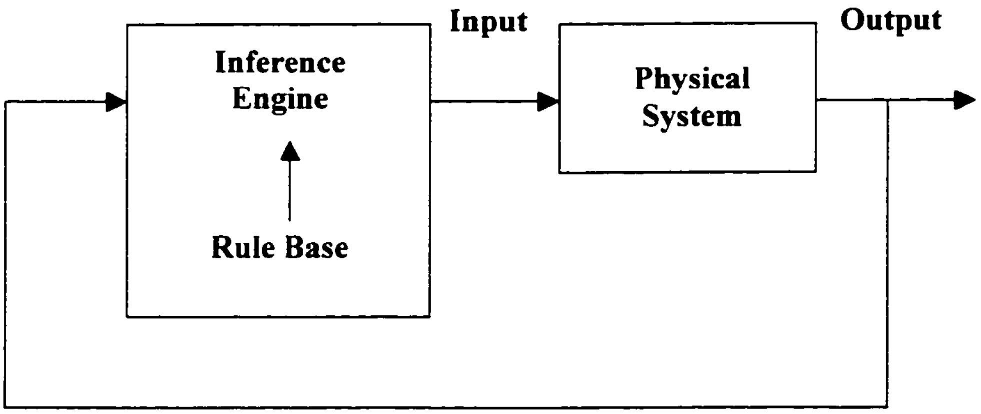


Figure 9 – 3 : Architecture of a Fuzzy Controller

The purpose of control is to influence the behavior of a system by changing input or inputs to that system according to a rule or set of rules that model how the system operates. The system being controlled may be mechanical, electrical, chemical, or any combination of these.

Again, classic control theory uses a mathematical model to define a relationship that transforms the desired state (requested) and observed state (measured) of the system into an input or inputs that will alter the future state of that system.

The most common example of a control model is the PID (proportional-integral-derivative) controller. This takes the output of the system and compares it with the desired state of the system. It adjusts the input value based on the difference between the two values according to Equation 9 - 1, where A, B and C are constants, e is the

$$\text{output} = A.e + B.\text{INT}(e)dt + C.de/dt \text{.................................Equation 9 - 1}$$

error term, INT(e)dt is the integral of the error over time and de/dt is the change in the error term.

The major drawback of this system is that it usually assumes that the system being modeled is linear, or at least behaves in some fashion that is a monotonic function. As the complexity of the system increases it becomes more difficult to formulate that mathematical model.[31]

Fuzzy control replaces, in Figures 9 –1 and 9 – 2 shown above, the role of the control element (mathematical model) and replaces it with another that is built from a number of smaller rules that in general only describe a small section of the whole system. The process of inference binds them together to produce the desired outputs. That is, a fuzzy model has replaced the mathematical one. The inputs and outputs of the system have remained unchanged.

To this point, the fuzzification of inputs, fuzzy rule base, and fuzzy inference engines have been discussed in detail. But there is another very import aspect that has just been mentioned in previous chapters. The concept is defuzzification; it is very important in fuzzy controllers and, hence, fuzzy control. Figure 9 – 4 displays defuzzification in a fuzzy controller and how it relates to the control system.

The method of choosing the correct defuzzification method is extremely critical in fuzzy control. There are three common methods used for defuzzification. These methods are center of area (centroid), mean of maximum, and center of maximum.

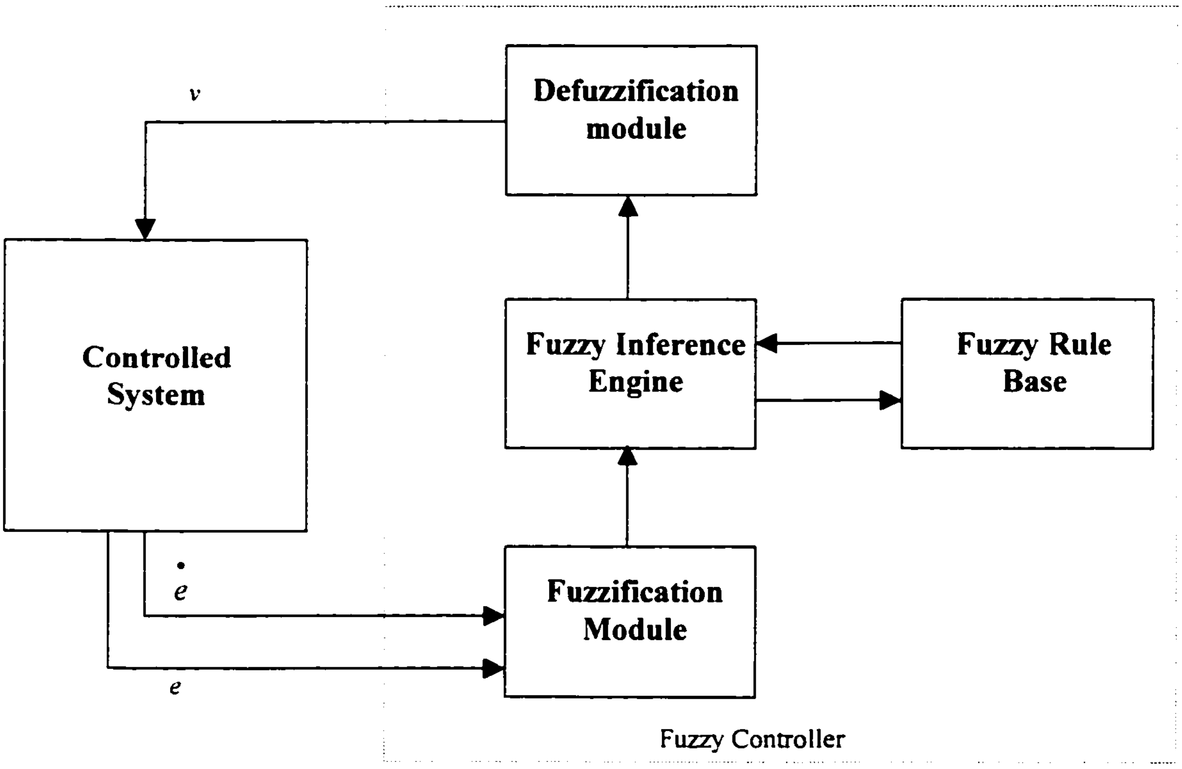


Figure 9 – 4 : Control System Featuring a Fuzzy Controller

The most popular method of defuzzification is the center of area method or the centroid method. The center of area method calculates the center of area under the curve. The defuzzified value, d_{CA} (C), is defined as the value within the range of variable v for which the area under the graph of membership function C is divided into two equal subareas. The variable v represents the relevant control actions of the fuzzy controller. This action is also known in classical control as feedback.[19] The remaining two variables are e and $\dot{e}$ are the error and the derivative of the error, respectively. These values are also shown in Figure 9 – 4. Furthermore, the center of area method is formally defined in Equation 9 – 2.

$$d_{CA}(C) = \frac{\sum_{k=1}^{n} C(z_k) z_k}{\sum_{k=1}^{n} C(z_k)} \quad \text{..Equation 9 - 2}$$

The next method of defuzzification is mean of maximum. The mean of maximum method selects the most typical value of the term that is most valid as an output value. This method's great advantage over other methods is that the computational time is very fast. The reason for this short execution time comes from its definition. This method is usually only defined in the discrete case. The defuzzified value d_{MM}(C) is the average of all values in the crisp set M. The formula for the mean of maximum is found in Equation 9 – 3.

$$d_{CM}(C) = \frac{\min\{z_k \mid z_k \in M\} + \max\{z_k \mid z_k \in M\}}{2}$$ ....Equation 9 - 3

The last defuzzification method discussed here is the center of maximum method. This method is not as fast computation wise as the mean of the maximum. This is due to the fact that it requires computation of a greater number of terms. Also in this method, the defuzzified value, $d_{CM}(C)$, is defined as the average of the smallest value and the largest value of v. The formula for this method is formally defined in Equation 9 – 4.

$$d_{MM}(C) = \frac{\sum_{z_k \in M} z_k}{|M|}$$ ....Equation 9 - 4

All of the previous methods are situation critical, meaning that the defuzzification methods may vary from one particular application to another. The application or system that is controlled will denote the type of method that bests suites its optimal operation. Considering all the previous aspects of a fuzzy controllers, there are three common attributes:

1. Base the controller on a human operator's experience and/or a control engineers knowledge.
2. Model the control actions of a human operator.
3. Base the control on a fuzzy model of the process.

Another type of fuzzy system that has gained a lot of attention in engineering science is neurofuzzy systems. The main components of neurofuzzy systems are neural networks. Neural networks have seen an explosion of interest over the last few years and are being successfully applied across an extraordinary range of problem domains. in areas as diverse as finance. medicine. engineering. geology and physics. Anywhere that there are problems of prediction. classification or control. neural networks are being introduced.

Neural networks grew out of research in artificial intelligence. specifically in attempts to mimic the fault-tolerance and capacity to learn of biological neural systems by modeling the low-level structure of the brain. The main branch of artificial intelligence research in the 1960s -1980s produced expert systems. These are based upon a high-level model of reasoning processes. It became rapidly apparent that these systems. although very useful in some domains. failed to capture certain key aspects of human intelligence. According to one line of speculation. this was due to their failure to mimic the underlying structure of the brain. In order to reproduce intelligence. it would be necessary to build systems with a similar architecture.

The brain is principally composed of a very large number (circa 10,000.000.000) of neurons. massively interconnected (with an average of several thousand interconnects per neuron. although this varies enormously). Each neuron is a specialized cell. which can propagate an electrochemical signal. The neuron has a branching input structure (the dendrites). a cell body. and a branching output structure (the axon). The axons of one cell connect to the dendrites of another via a synapse.

When a neuron is activated. it fires an electrochemical signal along the axon. This signal crosses the synapses to other neurons. which may in turn fire. Neuron fires only if the total signal received at the cell body from the dendrites exceeds a certain level (the firing threshold). The strength of the signal received by a neuron (and therefore its chances of firing) critically depends on the efficacy of the synapses. Each synapse actually contains a gap. with neurotransmitter chemicals poised to transmit a signal across the gap.

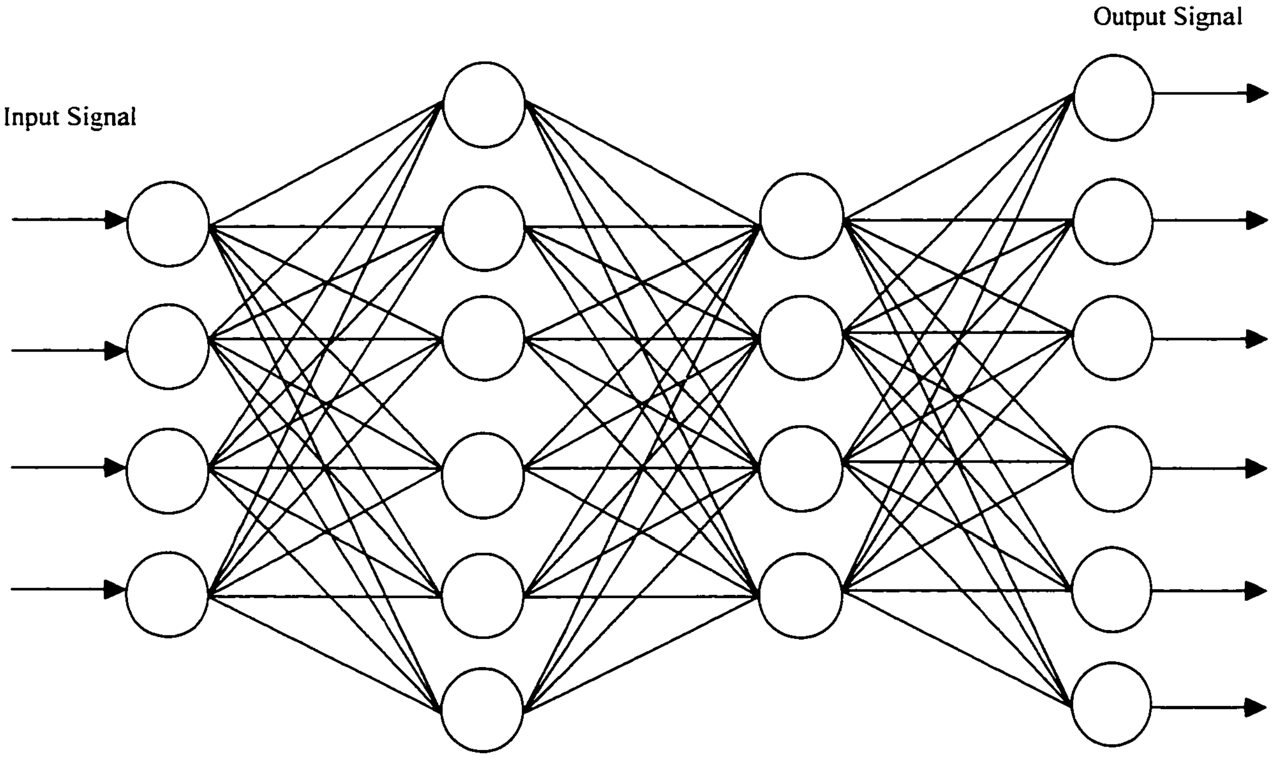


Figure 9 – 5 : Basic Structure of an Artificial Neural Network

Thus, from a very large number of extremely simple processing units (each performing a weighted sum of its inputs, and then firing a binary signal if the total input exceeds a certain level) the brain manages to perform extremely complex tasks. Of course, there is a great deal of complexity in the brain which has not been discussed here, but it is interesting that artificial neural networks can achieve some remarkable results using a model not much more complex than the one in Figure 9 – 5. It receives a number of inputs (either from original data, or from the output of other neurons in the neural network). Each input comes via a connection, which has a strength (or weight); these weights correspond to synaptic efficacy in a biological neuron. Each neuron also has a single threshold value. The weighted sum of the inputs is formed, and the threshold subtracted, to compose the activation of the neuron (also known as the post-synaptic potential, or PSP, of the neuron).

The activation signal is passed through an activation function (also known as a transfer function) to produce the output of the neuron. If the step activation function is used (i.e. the neuron's output is 0 if the input is less than zero, and 1 if the input is greater than or equal to 0), then the neuron acts just like the biological neuron described earlier (subtracting the threshold from the weighted sum and comparing with zero is equivalent to comparing the weighted sum to the threshold). Note also that weights may be negative, which implies that the synapse has an inhibitory rather than excitatory effect on the neuron: inhibitory neurons are found in the brain.

Neural nets are not often used in the way of applications for a number of reasons. First, neural net solutions remain a "black box" type of system, meaning that the user cannot interpret what causes a certain behavior or modify the neural network to change that behavior. Secondly, neural networks require prohibitive computational effort fur most commercial applications. Also, selection of the appropriate neural network model and parameters of the learning algorithm is a science within itself. Considering all the former statements, the lack of an easy way to verify and optimize a neural network solution is the major limitation.[7]

There are several weakness and strengths of neural nets that were discussed previously. But it must be noted that there are weaknesses and strengths of fuzzy logic as well. These basic characteristics of fuzzy logic and neural nets are shown in Table 9 -1.

Table 9 – 1 : Basic Characteristics of Fuzzy Logic and Neural Nets

	Fuzzy Logic	**Neural Networks**
Knowledge Representation	Explicit, verification and optimization is easy and efficient	Implicit, the system can be easily interpreted or modified
Trainability	None, the designer has to define everything explicitly	Trains itself by learning data sets

To obtain a clearer picture of neurofuzzy systems. a neurofuzzy controller is developed with the purpose of controlling a system. Any neurofuzzy controller must have the following four features:

1. Inputs are fuzzy numbers.
2. Outputs are fuzzy numbers.
3. Weights are fuzzy numbers.
4. Weighted inputs of each neuron are not aggregated by summation.

These features are representative of the melding of fuzzy logic and neural networks. As in Table 9 – 1. these two concepts complement each other in many ways. For instance. if an engineer would like to control a system. various measurements and other data are acquired. From these data sets the system is defined in great detail. To control this system conventionally. a mathematical model must be developed. To control this system by fuzzy logic. a rule set must be derived. This rule set can become extremely complex because of the enormous size of the rule base referring to the data sets. The neural network can train itself from these data sets. But if the engineer uses fuzzy logic he would have to derive the "if-then" rules from the data sets manually. Therefore a neurofuzzy controller would allow the engineer the ease of modifying or tuning the system with the rule base generated by the neural network. This concept of a general neurofuzzy controller is shown in Figure 9 – 6.

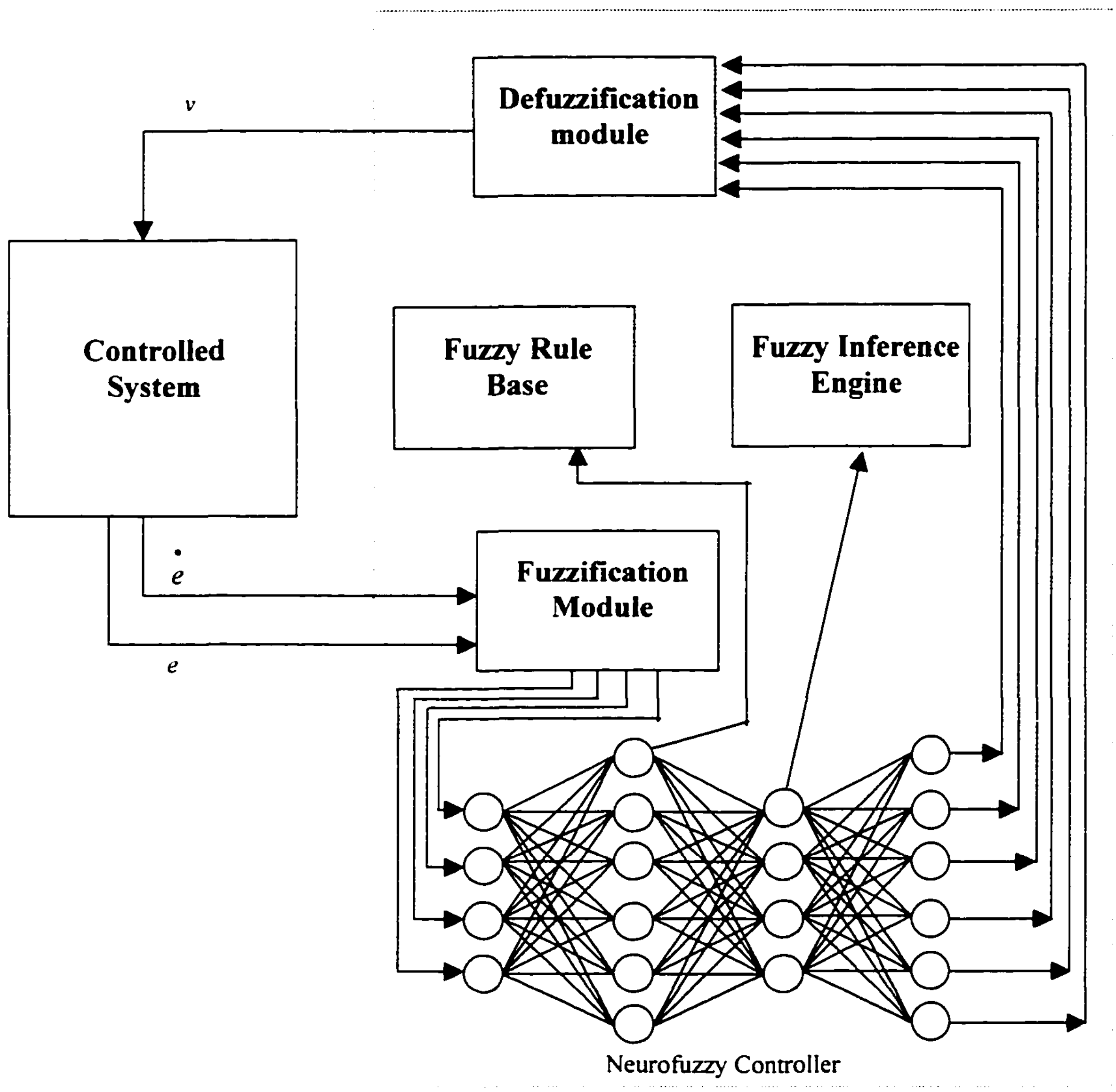


Figure 9 – 6 : Basic Structure of a Neurofuzzy Controller

CHAPTER X

AN APPLICATION OF FUZZY LOGIC

A new concept. no matter how great it is to the scientific community. must contribute to mankind in some way to be considered an asset. Fuzzy logic is simply a design tool that allows engineers to design products and control processes with greater ease. This is not the only use of fuzzy logic. but it is the focus of this chapter.

For fuzzy logic to be an asset to society, it must improve or enhance the lifestyles of people. One way to achieve these goals is to make the machines that are used in everyday life more user-friendly and efficient. These previous tasks have evolved by conventional means over the course of history. Many strides in product design and advancements are attributed to microelectronics and low cost computer components. The low cost computer components allow fuzzy logic to be exploited in a wide range of products for consumers.

A user-friendly product is very important, but the product must also be efficient in its use of resources. The resources may range from electricity. natural gas. gasoline. and even water. Water is a resource that is abundant on the earth's surface. Eighty percent of the earth's surface is water, and ninety-seven percent of the earth's water is saltwater in oceans and seas. Of the three percent that is freshwater. only one percent is available for drinking -- the remaining two are frozen in the polar ice caps. To further stress the importance of water. it can be related to human existence because people need about 2.5 quarts of water a day (from drinking or eating) to maintain good health. A person can live without water for approximately one week. depending upon the conditions.

To conserve this precious commodity the Environmental Protection Agency enacted several product standards. These standards have one objective in mind. and that is to conserve water. Toilets. nozzles. and showers have been modified to become water misers. For instance. a 10-minute shower can take 25 - 50 gallons of water. High flow shower heads spew water out at 6 - 10 gallons a minute. Low flow showerheads can cut the rate in half without reducing pressure.

About 60.000 public water systems across the United States process 34 billion gallons of water per day for home and commercial use. Eighty-five percent of the population is served by these facilities. The remaining 15 percent rely on private facilities. To help supply these facilities an average of 800.000 water wells are drilled each year in the United States. That's tapping into our underground water supplies at approximately 100 times each hour for domestic, farming, and commercial needs.

While water usage varies from community to community and person to person. on average. Americans use 183 gallons of water a day for cooking. washing, flushing. and watering purposes. The average family turns on the tap between 70 and 100 times daily. About 74% of home water is in the bathroom. about 21% is for laundry and cleaning. and about 5% is in the kitchen. A clothes washer uses about 50 gallons of water (the permanent press cycle uses an additional 15 gallons).[33]

Just by the percentages of water a family uses everyday, this is considered gluttonous by water usage over the previous centuries. But to accommodate the fast lives that Americans lead, it is not surprising that Americans use more water than they do. The largest percentage of water used is associated to the bathroom, which include toiletry and hygienic purposes. It is impossible to curtail these functions, but the second largest household water consumption, clothes washing, can be modified. This is where fuzzy logic can help to conserve water. Hence, this will conserve electricity and even precious time. But fuzzy logic does not require a mathematical model of the system to control the system. The requirement of fuzzy logic design is extensive knowledge of the system's operation. The system here is the washing machine found in a typical American household.

Today's washing machine is a far cry from the early wooden-tub devices that began it all, but the principles are not that different. Along with a few improvements that automate the washing process, the basic washing machine remains an electrically powered mechanical device that agitates clothes, soap, and water to remove the dirt. There are two types of washing machines in use today--the top loader and the front loader. The front-loading machine features a horizontal tub. After the machine is loaded with clothes and detergent, and the door is closed and locked, water enters the drum and the drum starts to turn. Agitation occurs as the clothes lift out of the water and then fall back down while the tub rotates.

In the second type, the top loader, the tub is mounted vertically. Inside the tub is a basket, or perforated tub, and inside that is an agitator--a component with radial fins.

In contrast to the front loader, the top loader cleans the wash through the back-and-forth rotation of the agitator. This type of design is the more popular washing machine and is shown in Figure 10 – 1.[34]

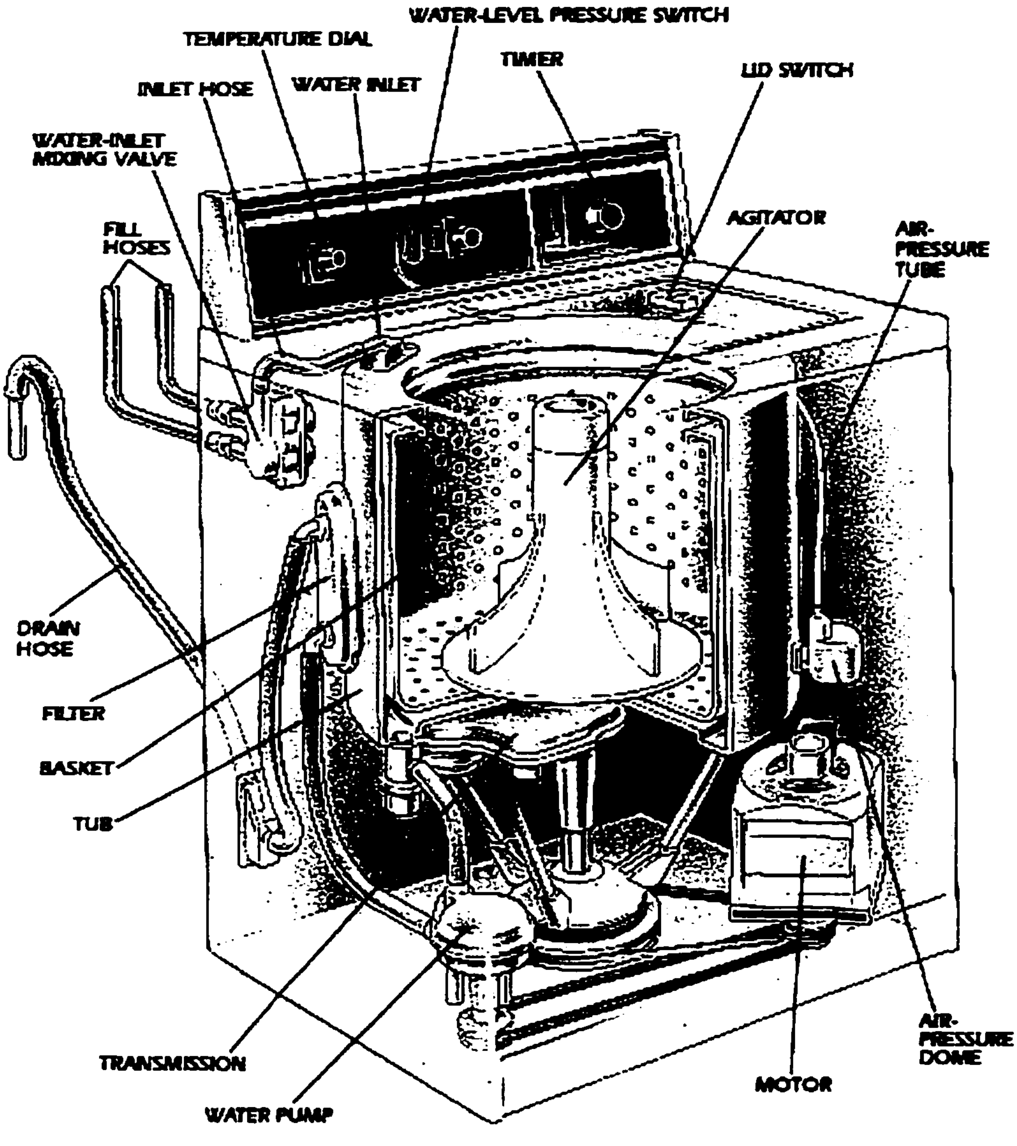


Figure 10 – 1 : Cut Away View of a Top Loaded Washing Machine

The brain of every automatic washing machine is its timer. This electromechanical device is powered by a small electric motor. Like a clock. the timer motor turns a series of gears to move cams that activate switches. The switches. in turn. control the various functions--wash, spin and rinse. for example--that make up the entire washing process. In addition to controlling the various cycles by setting the timer. you set the water level to suit the size of the load at the water-level dial and control the water temperature--either hot. warm or cold--with the temperature dial.

To start the machine. you first select a program of cycles. the wash time and the water temperature. Pressing the timer knob starts the timer motor and completes a circuit through the water-temperature selector switch and the water-inlet-mixing valve. The temperature switch regulates the amount of cold and hot water that passes through the water-inlet-mixing valve. Once the mixing valve has opened. water flows into the tub from the hot and cold water valves attached to your home's plumbing system.

From here. the water fills the tub and the perforated basket that contains the dirty clothes and detergent. This fill process is shown in Figure 10 – 2. As the water level rises. it forces air into the air-pressure dome mounted on the side of the tub. A tube connected to the air-pressure dome carries the pressurized air to the water-level pressure switch in the console. When the air pressure reaches a point that corresponds to the water-level setting on the water-level dial. the switch shuts off the flow through the water-inlet mixing valve.

Next. the timer signals the washer motor to start turning. The motor is connected to the agitator and basket through a series of gears. springs. cams and shafts that make up the transmission. Although transmission designs vary and have evolved over the years, the function of this assembly is the same: to convert the rotary motion of the motor shaft into the back-and-forth motion of the agitator, and, at the appropriate time, stop agitation and engage the basket for the spin cycle.

In the wash phase, the fins on the agitator slosh the water and detergent through

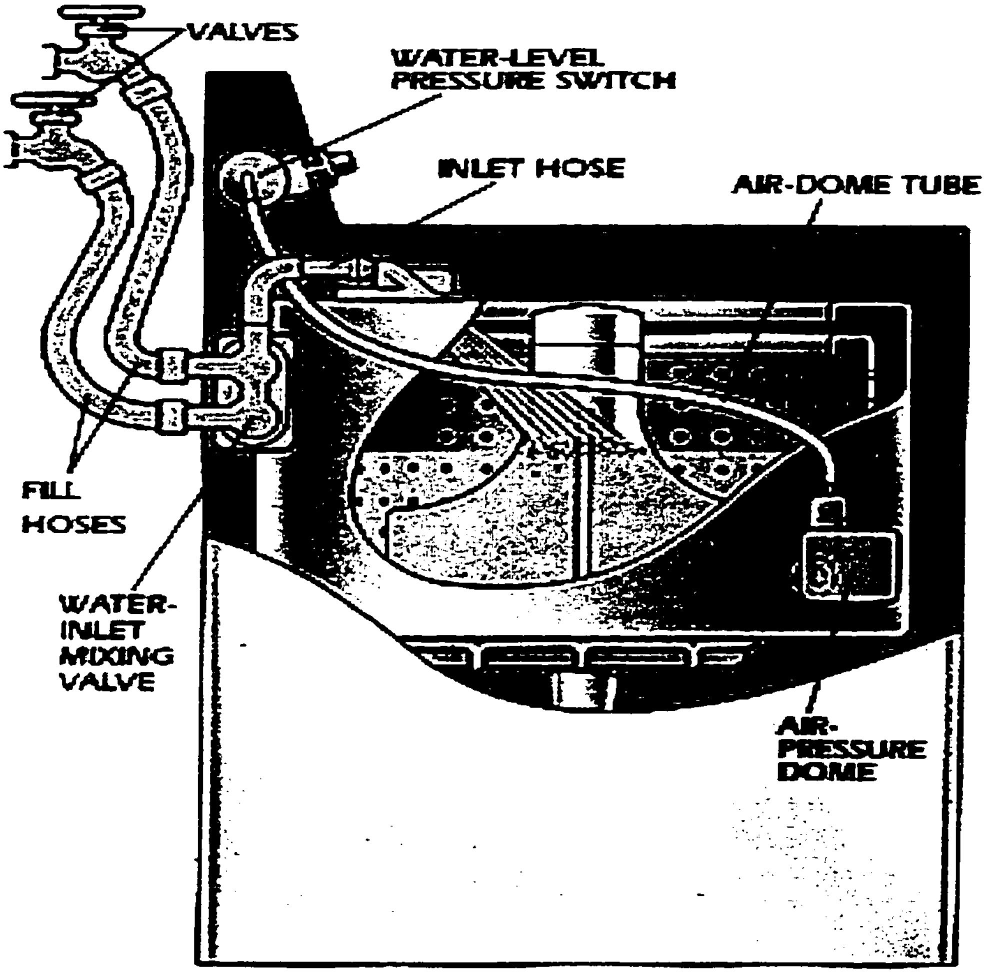


Figure 10 –2 : Fill Process of a Washing Machine

the clothes to remove the dirt. As the dirt is loosened from the clothes, it becomes suspended in the water. At the same time that agitation begins, the water pump is engaged. The pump circulates water from the tub bottom to the top, and routes the water through a filter that catches lint and other particles. When the wash cycle is over, the motor stops momentarily and then starts in reverse. At this point, the agitator

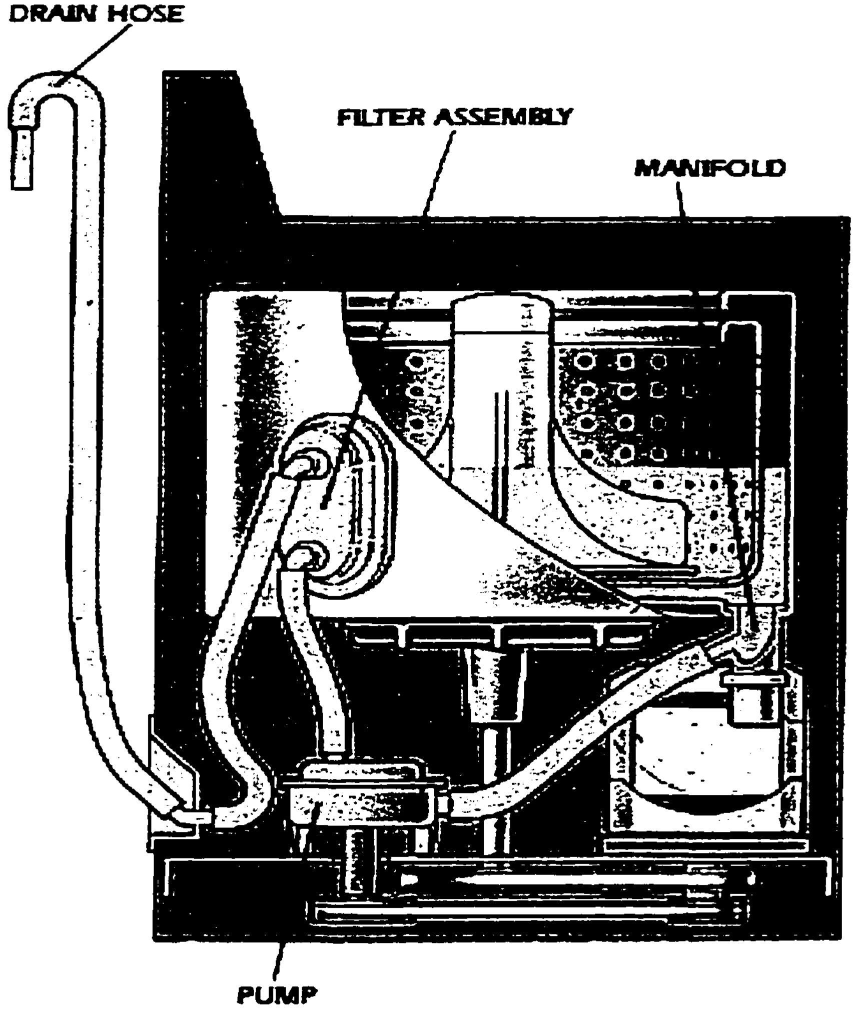


Figure 10 - 3: Drain Cycle of Washing Machine

disengages and the pump moves the water in the opposite direction. flushing the filter and sending the dirty water out the drain hose. This process is shown in Figure 10 - 3.

Once the pump has removed most of the water. the timer advances to a spin cycle. The transmission now connects the motor to both the agitator and the basket. but disconnects the gears that control the agitation. As the rotating basket picks up speed. centrifugal force moves the water out of the clothes and through the basket perforations to the tub where it's pumped to the drain. As a safety feature. washers have a switch inside the lid that disconnects the motor if the lid is opened during the spin cycle.

When the spin cycle is finished. the timer advances to the rinse cycle. During this phase. the tub again fills with water to the predetermined level and temperature. The timer then begins another short agitation cycle to remove any dirt and detergent that may remain in the clothes.

Following this agitation. the timer drains the tub and signals the motor and transmission to begin a high-speed spin that removes most of the water from the clothes. During the final spin cycle. the timer opens and closes switches that control the water-inlet mixing valve. This sends bursts of water into the baskets that are called spray rinses. The spray rinse helps remove any remaining dirt or detergent in the clothes. When the final spin cycle is complete. the timer stops the motor and shuts itself off.[34]

Defining the system and how it operates is very important to the redesign of a commercial product. The description of system operation of a typical washing machine has been done adequately in previous paragraphs. But this is only one step to developing a redesigned product.

The other factors involved in product redesign are the cost of the product, demand of the product, and the minimization of resources which the product utilizes. The latter deals with conservation, and conservation is a factor of commodity economics. For a person to conserve energy or natural resources there must be an underlining benefit to that consumer in the way of cost reduction. People do not generally conserve anything unless it creates an undue monetary burden. In some areas of the country, water is extremely expensive, and therefore it is conserved. But in other areas water is fairly cheap so there is no reason for the consumer to conserve water. For a product to be designed strictly for the conservation of water is somewhat shortsighted by the designer. But, the product must be cost efficient and have comparable performance to other washing machines on the market.

Enhancing a washing machine with a multitude of sensors and other electronic components will drive the price of the product up so high that the majority of consumers will not purchase the product. This is a very elementary concept called the law of supply and demand. According to the theory, or law, of supply and demand, the market prices of commodities and services are determined by the relationship of supply to demand. Theoretically, when supply exceeds demand, sellers must lower prices to stimulate sales; when demand exceeds supply, prices increase as buyers compete to buy goods. In economic theory, supply is the amount available for sale or the amount that sellers are willing to sell at a specified price, and demand is the amount purchasers are willing to buy at a specified price. This concept is depicted graphically in Figure 10 – 4. From this figure one can see that the point of equilibrium is the most pleasing place for manufacturer to locate their product. The reason for this is that if demand is great and supply is not, present prices go up and a void in the market opens and competition fills in the gaps. But when supple is high and demand is low, prices drop and the manufacturer has a surplus of inventory causing an erosion of profits for the company.

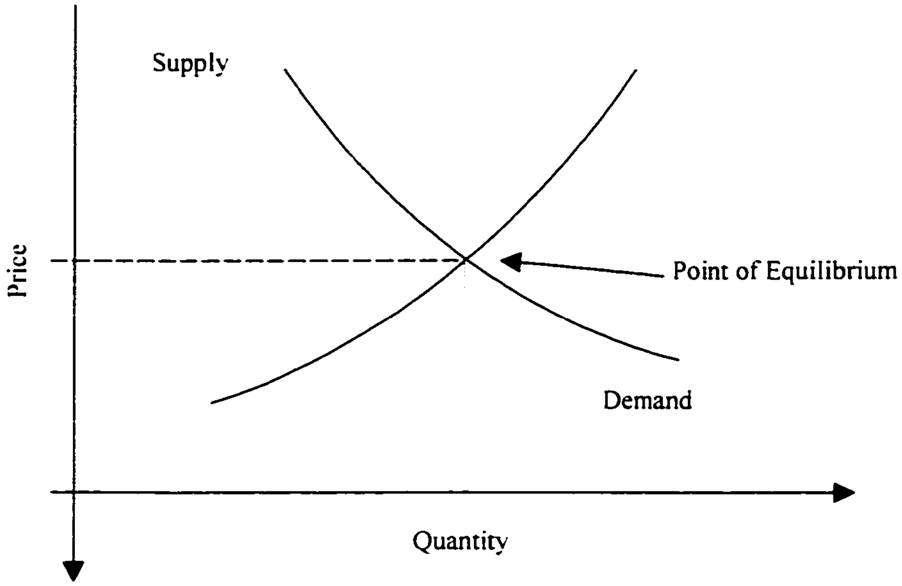


Figure 10 – 4 : Supply and Demand Curves for a Product

From these facts a better washing machine can be developed that meets the needs of consumers and also conserves water. There have been recent developments in other parts of the world, such as Asia and Europe, where advancements in washing machines included fuzzy logic. These washing machines are very different than the washing machines in America. The foreign washing machines engage in washing cycles for more than 2 hours. This is unheard of in America where washing cycles take 30 to 45 minutes. Also, these foreign washing machines heat the water and the American machines do not.

The main manufacturers of washing machine in America have not experimented with fuzzy logic control in washing machines as of yet. The reason for this is the acceptability of fuzzy logic in engineering fields. The concept of fuzzy logic control is just starting to emerge in consumer products in the United States, but we are far behind Europe and Asia in this area. These advancements can be seen in Table 10 – 1, where a fuzzy European washer is compared to three top American washers. It must be noted that the wash cycle is different, but this European machine is still cleaning clothes.

Table 10 -1 : Energy and Water Consumed by the Leading Washing Machines

	Kasko	Maytag	GE	Whirlpool
water used	28 gal	41 gal.	47 gal.	64 gal
power/ wash	260 WH	490 WH	550 WH	380 WH

There is significant savings of 13 gallons or more of water associated with the Kasko washing machine. So if fuzzy logic could be adapted to domestic washing machines, the water savings would be enormous. Water saving is a good benchmark but why stop there. With fuzzy logic control, the washing machine could also save time and only wash clothes as long as they need to be washed. This is also true with the rinse cycle - rinse clothes until the rinse water is clean. The rinse and wash cycles are two different control systems. But fuzzy logic is utilized in both systems to increase performance and efficiency.

Two different wash systems are considered. The first wash cycle incorporates only one sensor which distinguishes the dirtiness of the water. The sensor is of an optical origin and it measures the amount of light that passes through the water. This measurement happens right after the wash cycle starts. The reason for this is to get a good reading of the dirtiness of the water once the clothes have absorbed the water and saturation is reached. From this point the fuzzy controller determines the length of the wash cycle. This fuzzy controller is very simple because it only has one input and one output. This could be accomplished by a conventional controller and the performance would be the same. But if other inputs were added, as in the next wash cycle, the program would become very complex. The block diagram of the system is shown in Figure 10 – 5 and a input vs. output graph for this wash cycle is shown in Graph 10 – 1.

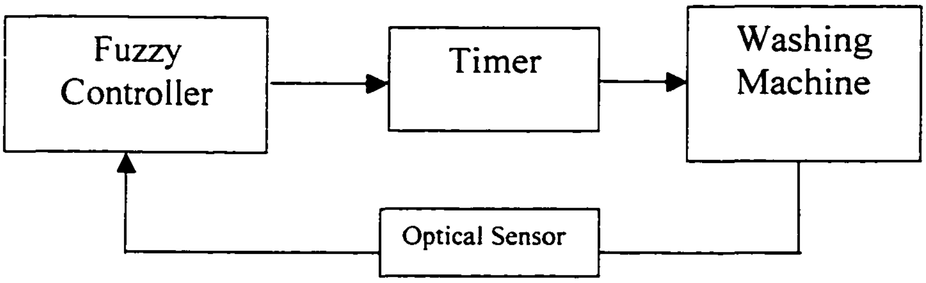
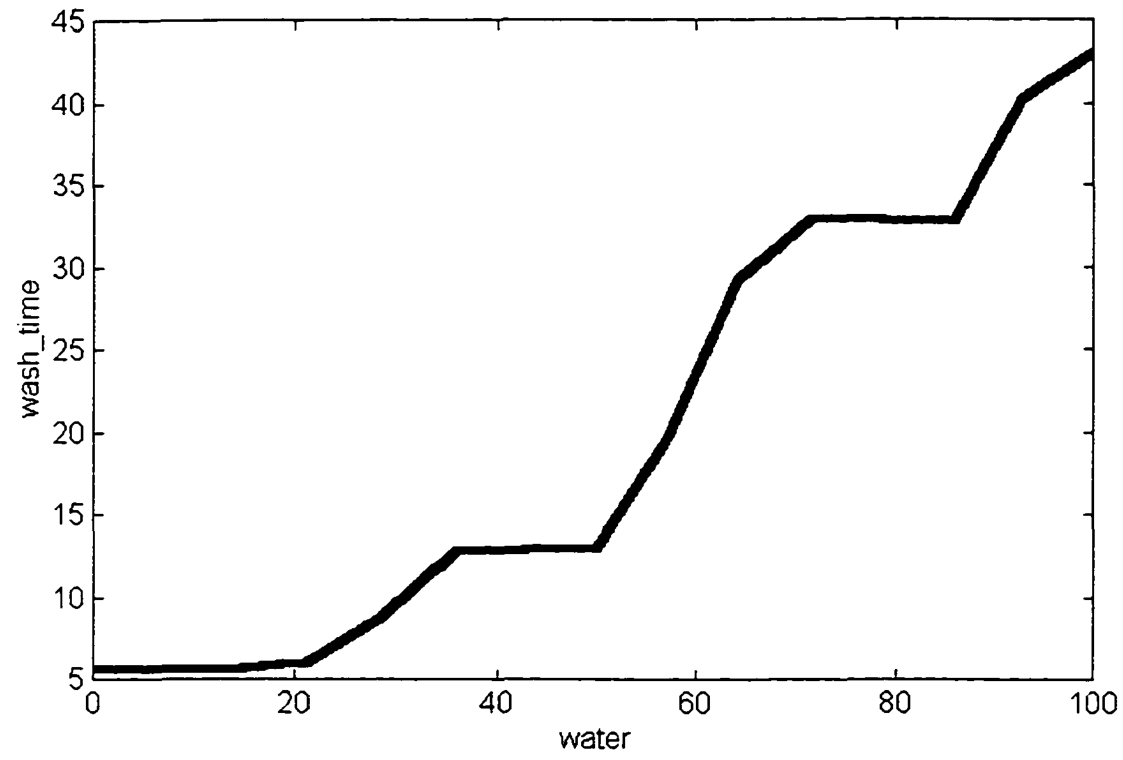


Figure 10 - 5 : Basic Fuzzy Washing Machine Control for Wash Cycle



Graph 10 - 1 : Wash Time vs. Water Dirt Level for Basic Wash Cycle for the Fuzzy Controller

This wash cycle is a closed loop control system with one feedback loop. There is not any actual feedback because the optical sensor only supplies one value to the fuzzy controller per cycle. This system has advantages and disadvantages:

Advantages

1. Prevents over washing, which saves electricity and prolongs clothes life
2. Prevents under-washing, which saves water that would be used in another wash cycle

Disadvantages

1. Does not save any water in wash cycle
2. Does not conserve washing detergent
3. User must select water temperature
4. User must select load size

The wash cycle is half of the design and the rinse cycle is the other half. There are two different types of rinse cycles that are used in washing machines. The first rinse cycle fills the tub partly with water and agitates the clothes. The other rinse cycle sprays the clothes with water and spins the clothes at the same time. The later type is more efficient and uses less water than the first rinse cycle. Therefore the second rinse cycle shall be included in this design.

To improve upon the rinse cycle the same sensor used in the wash cycle is used in this cycle. The optical sensor is used to determine if the clothes are rinsed to a suitable degree. To determine how long to rinse the clothes. a rinse cycle is started that is of a small time frame. The water is drained and the optical sensor measures the dirtiness of the water. The fuzzy controller determines the time allowed for the rinse cycle. If necessary. other rinse cycles are done until the clothes are sufficiently clean. Also there is a preset limit of dirtiness. where the rinse cycle ends. The block diagram of the system is shown in Figure 10 – 6 and an input vs. output graph for this rinse cycle is shown in Graph 10 – 2.

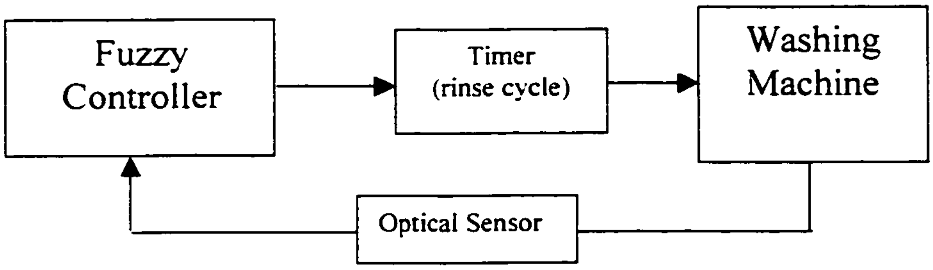
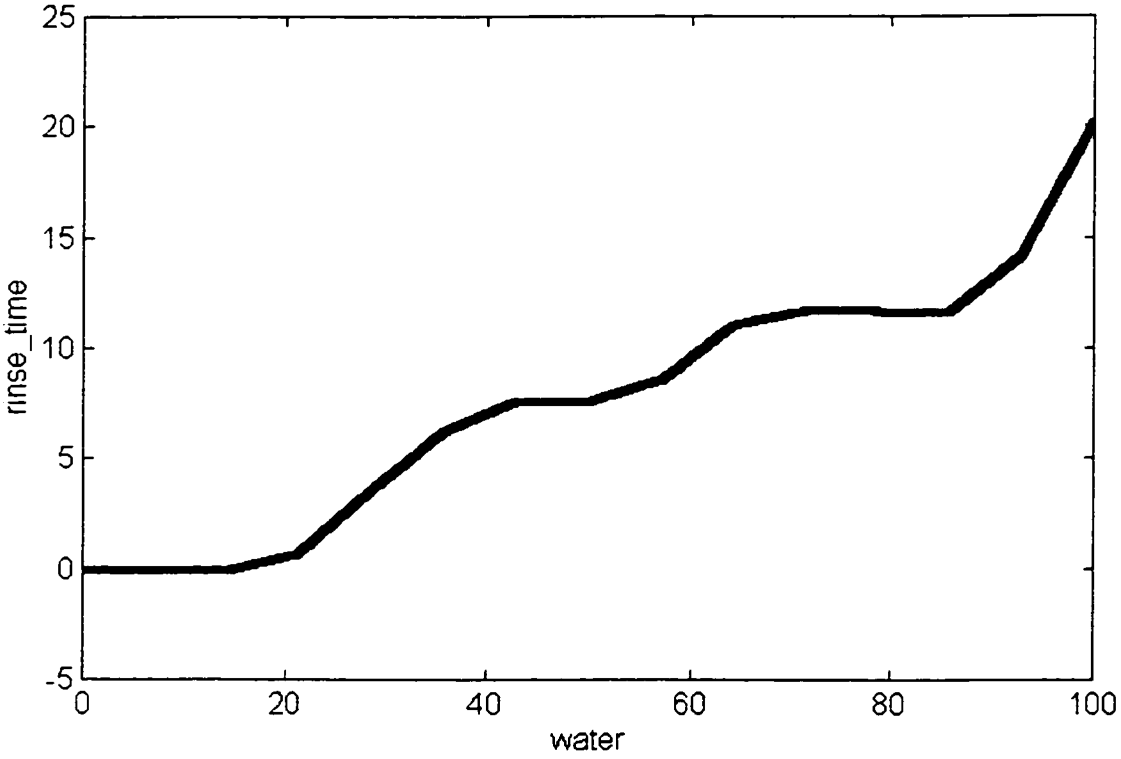


Figure 10 - 6 : Basic Fuzzy Washing Machine Control for Rinse Cycle



Graph 10 - 2 : Rinse Time vs. Water Dirt Level for Basic Rinse Cycle for the Fuzzy Controller

This wash cycle is also a closed loop control system, with one feedback loop. There is feedback in this system unlike the wash cycle. Just like the wash cycle this system has advantages and disadvantages, such as:

Advantages

1. Prevents over rinsing, which saves electricity
2. Prevents under-rinsing, which saves water that would be used in another rinse cycle

Disadvantages

1. None

This type of basic fuzzy washing machine would be very competitive against other low cost washing machines. The washing machine developed here would not fit every consumer, but it does offer a low cost alternative which saves water. The water savings is estimated to be 5 to 7 percent, but fine tuning of the system could result in a greater savings. These savings are estimated from the average washing cycle compared to the different washing cycles generated by the fuzzy controller. It must be noted that in Graphs 10 - 1 and 10 - 2 the water dirtiness level is factored between 0 and 100. This range is arbitrary and was used to get a better resolution for the fuzzy system. Any reasonable range could be used in place of this one. For a detailed look at the fuzzy logic system for the wash and rinse cycles, refer to Appendix 1 and 2, respectively.

Every consumer does not want a low budget washing machine. For those consumers, options, efficiency, and features are the selling points. Price is not necessarily a factor in the purchase decision.

The second type of fuzzy washing machine design contains a variety of sensors and user-friendly features. A block diagram of this system is shown in Figure 10 - 7.

The output graphs for the deluxe wash cycles for a fuzzy controller are shown in Graphs 10 - 3 through 10 - 5.

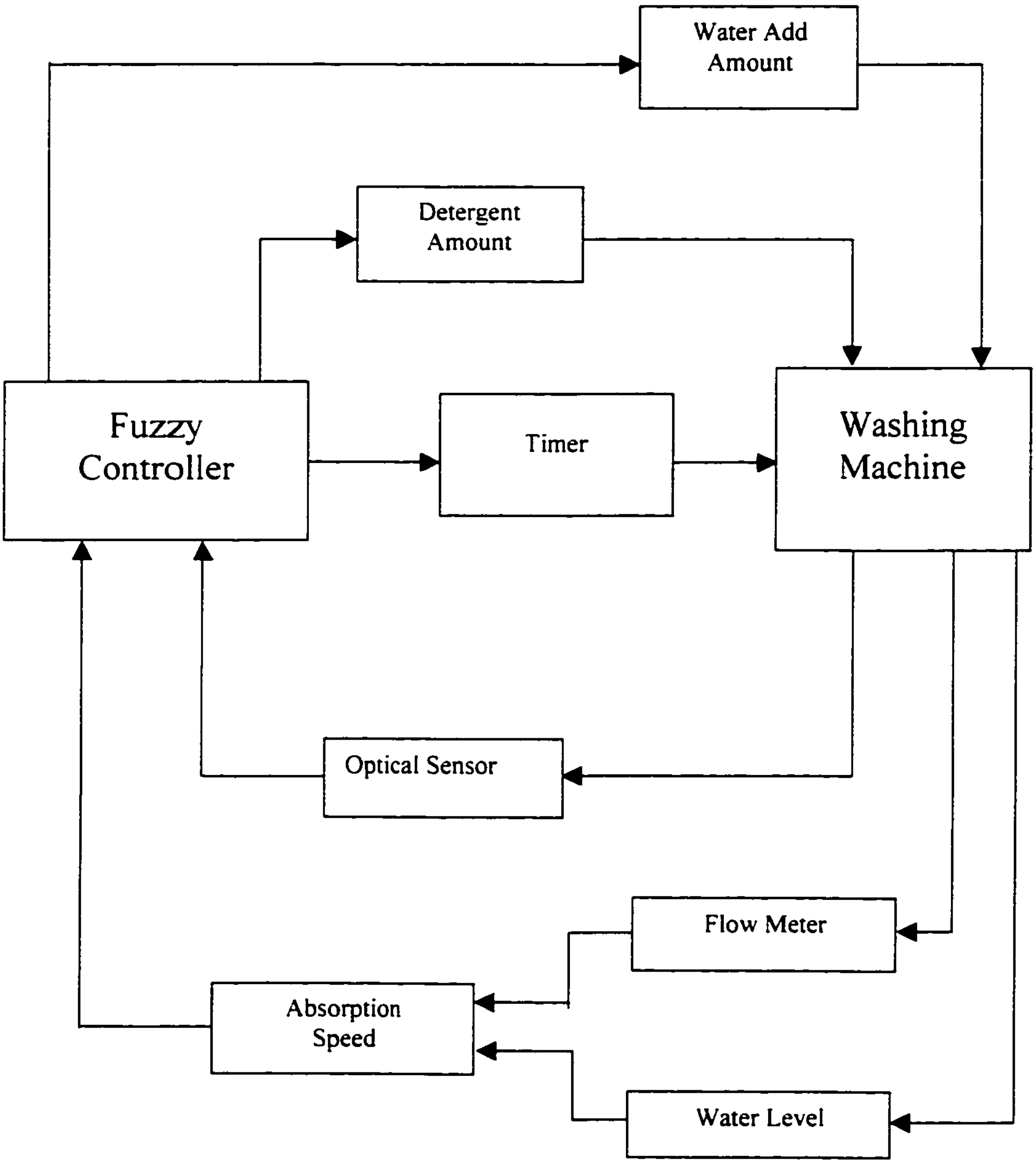
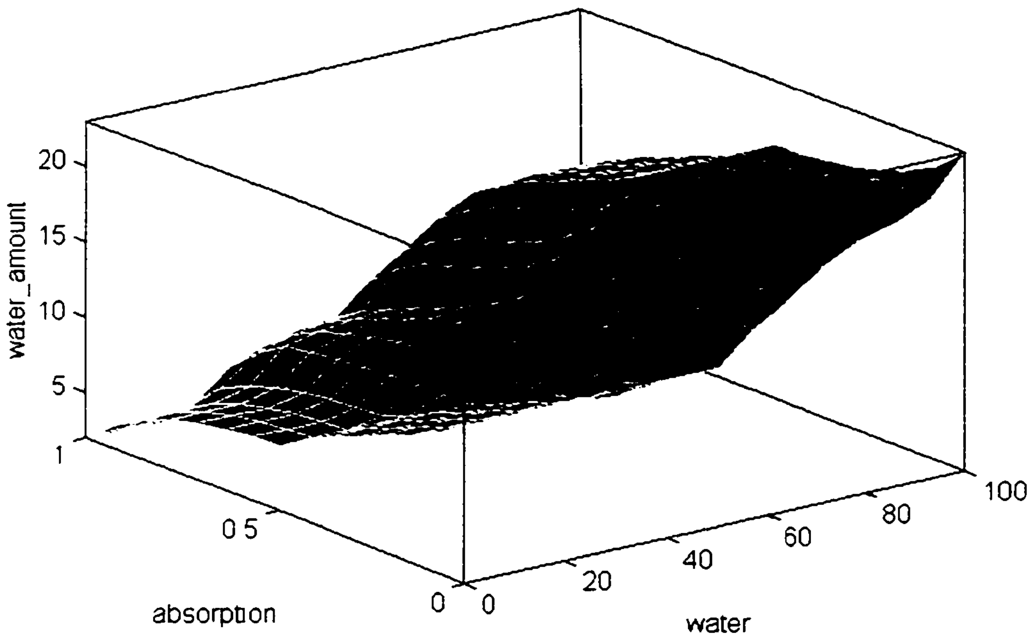
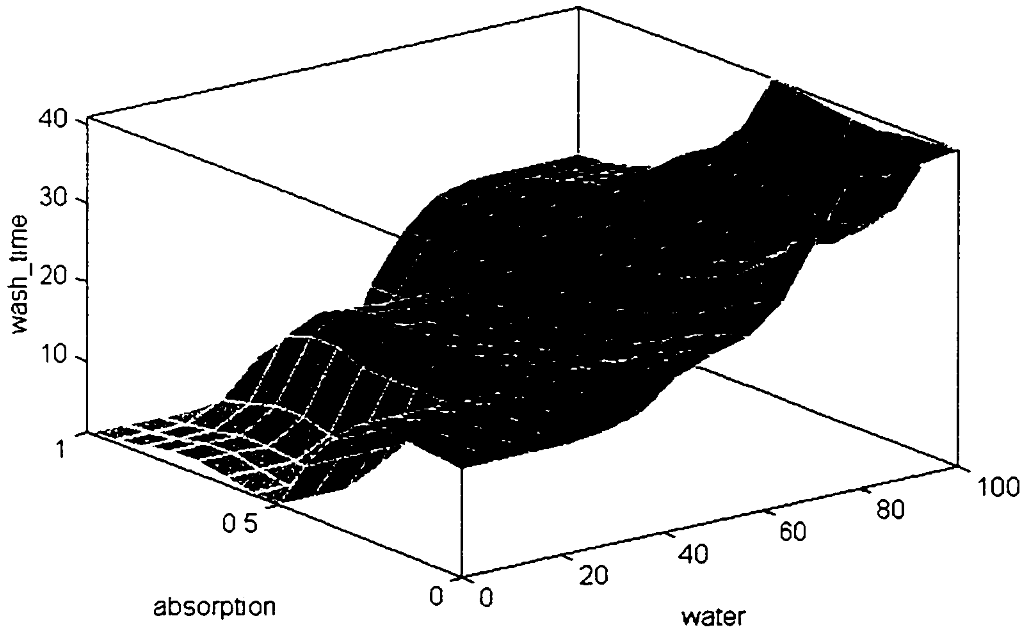


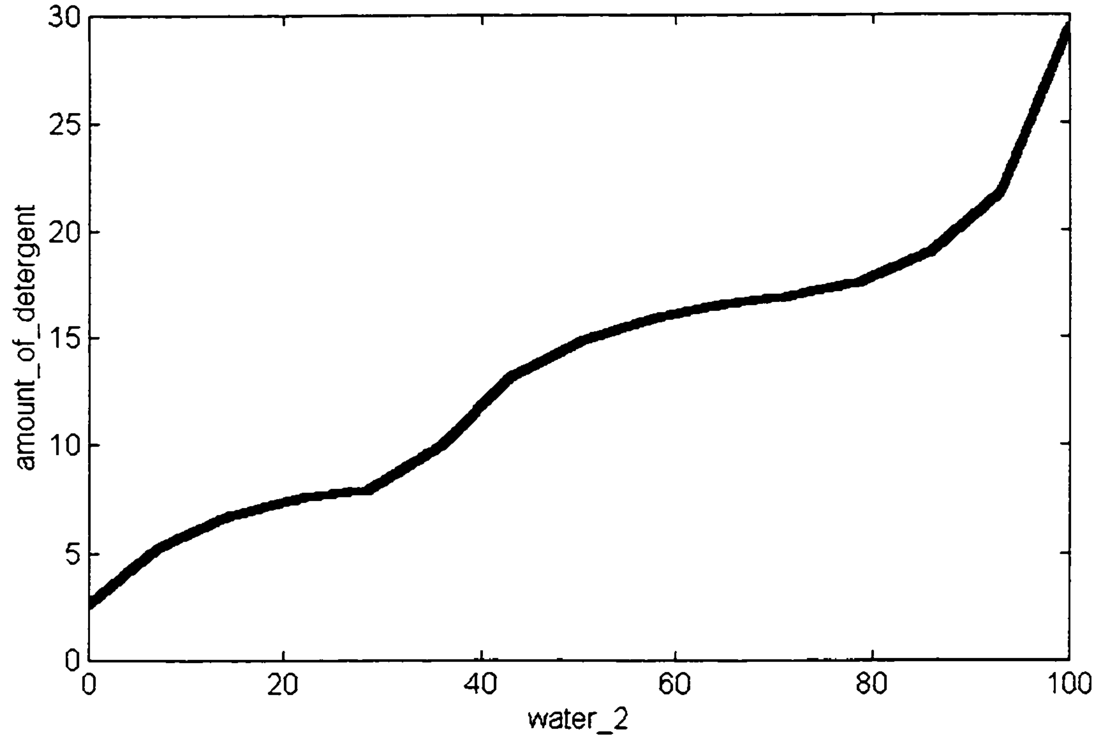
Figure 10 - 7 : Deluxe Fuzzy Washing Machine Control for Wash Cycle



Graph 10 - 3 : Deluxe Fuzzy Washing Machine Control for Water Amount



Graph 10 - 4 : Deluxe Fuzzy Washing Machine Control for Wash Time



Graph 10 - 5 : Deluxe Fuzzy Washing Machine Control for Amount of Detergent

The block diagram shown in Figure 10 – 6 shows the system flow of the deluxe wash cycle. From this block diagram one can ascertain the fuzzy controller's inputs. and outputs. These inputs and outputs are as follows:

Inputs

1. Optical sensor or water dirtiness
2. Absorption speed

Outputs

1. Wash time
2. Water amount
3. Detergent amount

The optical sensor input is the same as the one for the basic wash cycle. This sensor determines the water's dirtiness. The absorption speed is calculated from the water level sensor and the flow meter. A level sensor measures the height of water in the drum of the washing machine. The flow meter measures the amount of water that enters the drum of the washing machine. By dividing the amount of water entering the tub of the washing machine by the water level in the tub of the washing machine, a value of absorption is generated. This absorption value gives an accurate account of the water needs of clothes in the washing machine. But this value can determine the water needs of clothes with a fixed dirtiness. For the design to approach real world criteria, the optical input is used along with the absorption factor to calculate the water amount. This calculation is continued until the tub of the washing machine is filled to the needed capacity. A three dimensional plot of water amount vs. water dirtiness and absorption is shown in a Graph 10 - 3. The dirtier the water and the slower the absorption, the greater the water amount; the cleaner the clothes and the faster the absorption, the smaller the water amount. For the absorption to be fast, the amount of clothes is small or the types of clothes have a low absorption rate. There are also two situations for a slow absorption rate - there are a lot of clothes in the washer or some of the clothes in the washer have a high absorption rate. A subprogram could be included to avert the tub from overflowing if too many clothes were put in the washer.

To compute the wash time for a specific load of laundry, the optical sensor input and absorption input are used for this calculation. These two inputs are the same as the ones used above in the water amount computation. To compute the wash time the tub must be filled with water. The absorption rate is used along with the dirtiness of the water to compute a wash time. This is formally shown in Graph 10 - 4. In this graph the dirtier the water and the slower the absorption. the greater the wash time: the cleaner the clothes and the faster the absorption. the smaller the wash time. But the clothes can be extremely dirty and if the absorption rate is low. the wash time will not be at maximum. The reason for this decision by the system is that clothes that do not absorb great amounts of water come cleaner faster. Clothes that do absorb great amounts of water come clean slower and. hence. there is a greater wash time.

The last output of the system is the detergent amount used in the wash cycle. This output is solely dependent on the optical sensor. If the water is dirty then more detergent is added to the wash cycle. The computation is only done once after the tub is filled with water and before the agitation begins. This is formally shown in Graph 10 - 5.

The rinse cycle for the deluxe fuzzy washing machine is exactly like the rinse cycle for the basic fuzzy washing machine. There was no need for improvement of this rinse cycle and the system integration was extremely simple. For a detailed look at the fuzzy logic system for the wash cycle of the deluxe fuzzy washing machine. refer to Appendix 3.

In conclusion, washers can be needless water-guzzlers as well as watt-hogs. The advent of fuzzy logic controllers allows less water than is considered "normal" per washing cycle. giving far more "wash for the slosh." If these water saving methods were employed in major brands of washing machines to their fullest. less water would be wasted on doing laundry.

CHAPTER XI

CONCLUSION

Fuzzy logic is truly a revolutionary concept that will no doubt become an integral part of engineering and design in the coming years. Designs that feature fuzzy logic are cropping up everywhere from toasters to automobiles. But these products are from Europe or Asia. The scientific community in the United States must make a concerted effort to catch up with the rest of the world in the area of fuzzy logic. If this does not happen soon, corporations in the United States that produce durable goods will see their market share dwindle right before their eyes. This is a very serious situation and must not be taken lightly.

The main focus in the development of this dissertation was to determine the usefulness of fuzzy logic and fuzzy control when applied to a commercial appliance. But since fuzzy logic is extremely new in the country by the way of applications, journal papers, and other literature, there was a need to develop background concerning fuzzy logic. This background was divided into two different areas, set theory and fuzzy set theory. The second part presented fuzzy sets, fuzzy systems, fuzzy mathematics, and fuzzy control, with the model example conveying the use of all of the previous subjects. The model example also parallels and contrasts the two different concepts, fuzzy set and classical sets.

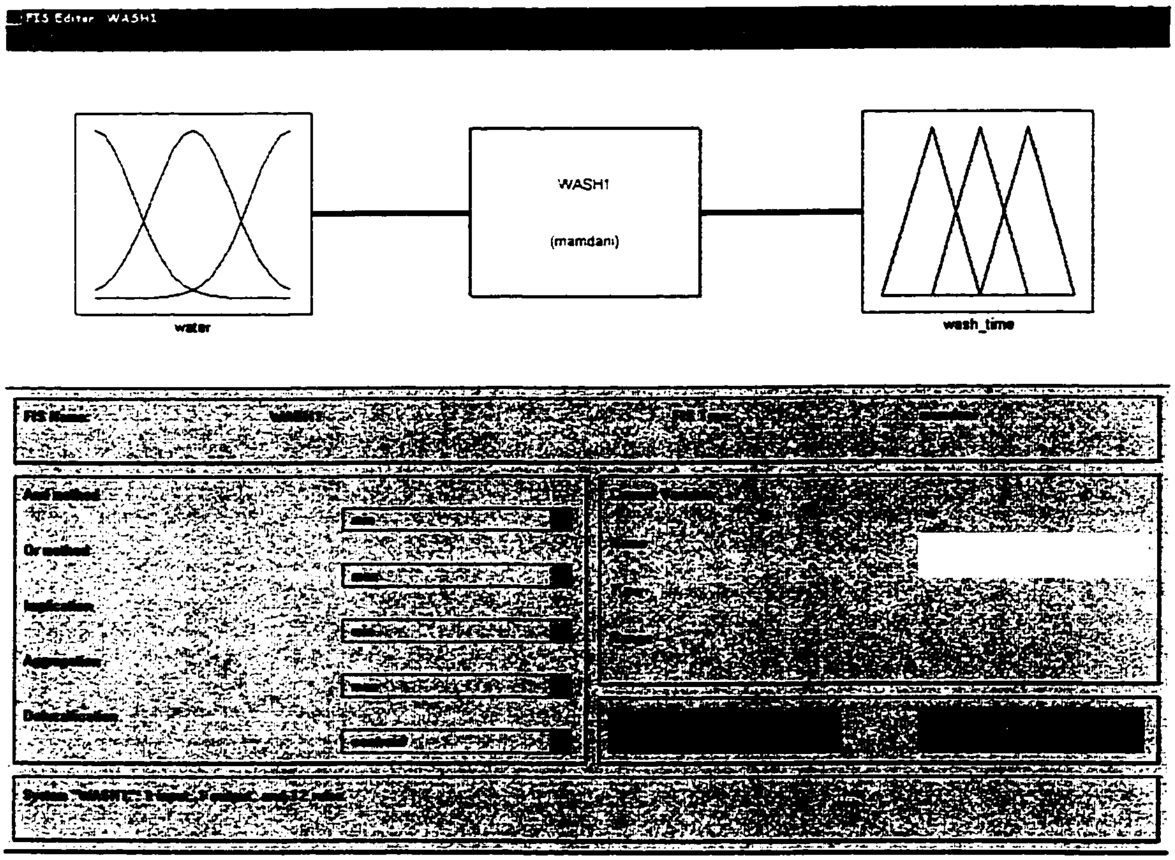
The last major part of this dissertation was fuzzy control of a consumer product. This part demonstrates that the use of fuzzy systems makes a viable addition to the field of engineering and. perhaps. more generally to the application of other consumer products to reduce energy consumption and increase the ease of operation. The consumer product considered was a typical washing machine found in any residential home. Two ways to improve this device were implemented by a fuzzy logic controller. The first improvement was water savings and the second was electrical energy savings. Two different types of washers were designed to meet the different needs of different consumers. One washer was a basic model. with few enhancements and a low price as well. The second type of washer had many sensors. which allowed several features on the machine.

The design of the washer's control system is very difficult to model in the conventional sense. But fuzzy logic is a structured. model-free estimator that approximates a function through linguistic input/output associations. Fuzzy rule-based systems apply these methods to solve many types of "real-world" problems. especially where a system is difficult to model. The fuzzy logic toolbox of the program MATLAB™ developed by The Mathworks Corporation was used to build and test the fuzzy logic systems for the model example and the washing machines. This toolbox was very straightforward but there were several bugs in the toolbox. It is not uncommon with a first version of any program. but none the less it was very annoying in the design stage. After using the program for many weeks. these bugs were anticipated and did not impede progress. but it is good to know that the new version of the software has fixed these bugs and other problems.

Again. in this dissertation the concept of fuzzy logic was explored fully. Background and theoretical information was derived to provide a good base for a variety of applications. Classical logic. crisp sets. fuzzy sets. and operations on fuzzy sets were explained in order to cover a wide spectrum of applications. Fuzzy logic allows many household goods to be manufactured more quickly and with more options. and also be energy efficient. user friendly. and cost effective. The washing machine is only one example where fuzzy logic can be applied. The savings in water and electricity make this design very attractive to many consumers. It will be interesting to see if the major U.S. appliance manufactures embrace fuzzy logic or continue on the path of conventional control.

APPENDIX 1

Basic Fuzzy Wash Cycle Control



FIS Name: WASH1

FIS Type: mamdani

And method: [illegible]

Or method: [illegible]

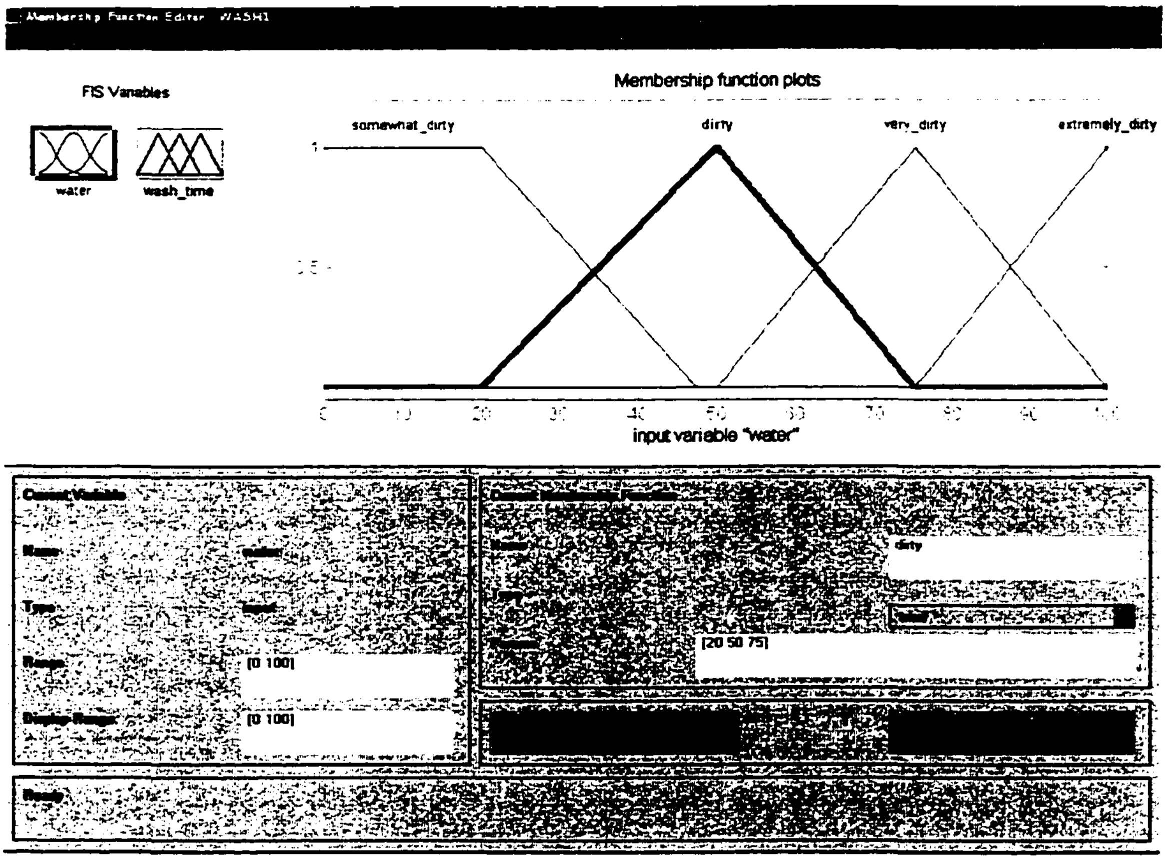
Implication: [illegible]

Aggregation: [illegible]

Defuzzification: centroid

System "WASH1": [illegible]

Figure A1 - 1 : FIS Editor for Basic Wash Cycle



Current Variable

Name: water

Type: input

Range: [0 100]

Display Range: [0 100]

Current Membership Function

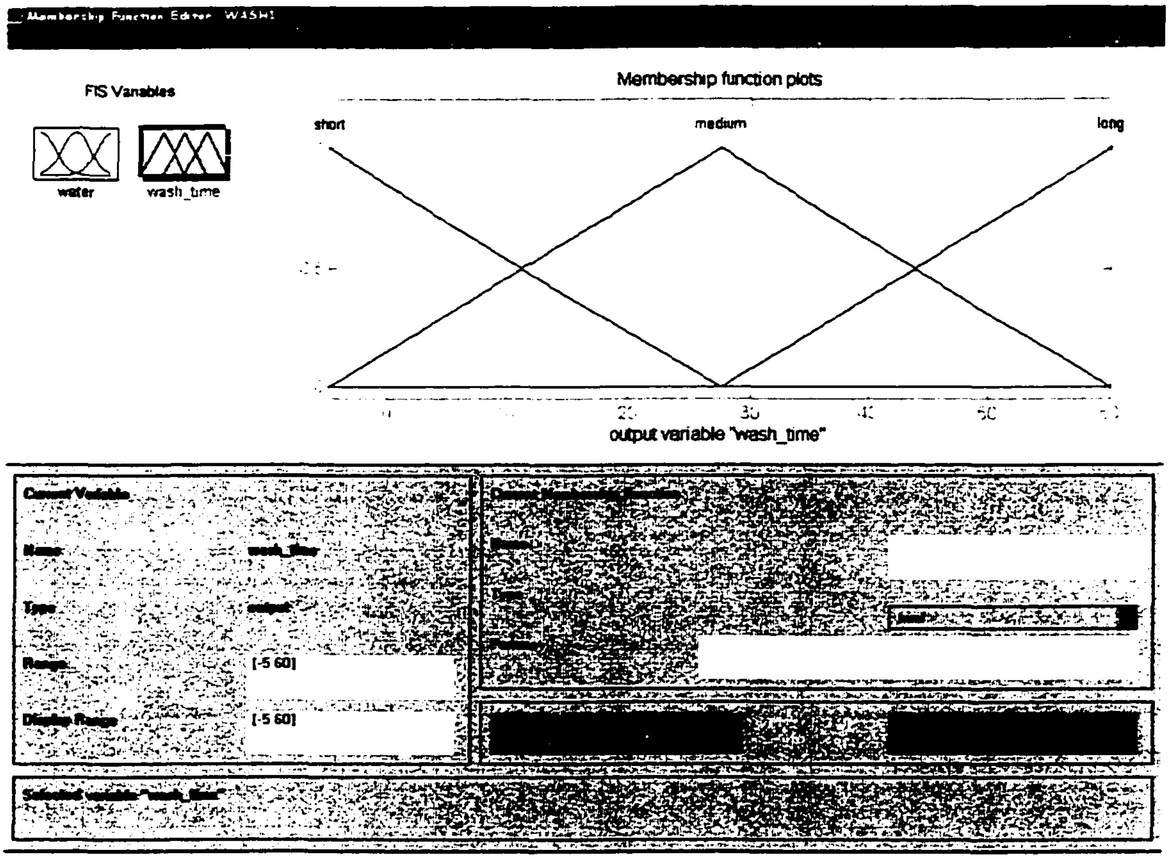
Name: dirty

Type: trimf

Params: [20 50 75]

Ready

Figure A1 - 2 : Membership Function Editor for Water Input of Basic Wash Cycle



Membership Function Editor: WASH1
FIS Variables
water
wash_time
Membership function plots
short
medium
long
0.5
0
20
30
40
50
60
output variable "wash_time"

Current Variable

Name wash_time

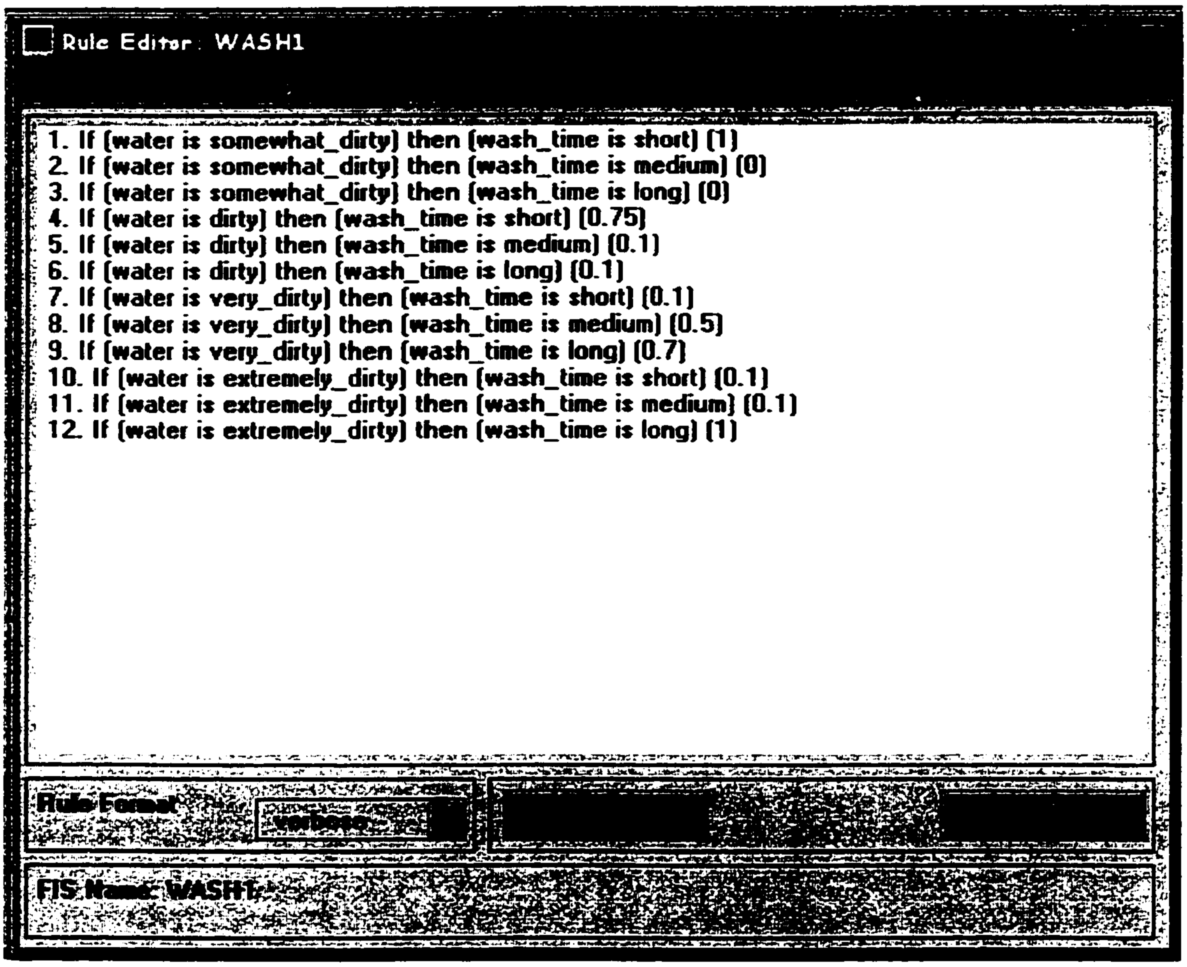
Type output

Range [-5 60]

Display Range [-5 60]

Selected variable "wash_time"

Figure A1 - 3 : Membership Function Editor for Wash Time Output of Basic Wash Cycle



1. If (water is somewhat_dirty) then (wash_time is short) (1)
2. If (water is somewhat_dirty) then (wash_time is medium) (0)
3. If (water is somewhat_dirty) then (wash_time is long) (0)
4. If (water is dirty) then (wash_time is short) (0.75)
5. If (water is dirty) then (wash_time is medium) (0.1)
6. If (water is dirty) then (wash_time is long) (0.1)
7. If (water is very_dirty) then (wash_time is short) (0.1)
8. If (water is very_dirty) then (wash_time is medium) (0.5)
9. If (water is very_dirty) then (wash_time is long) (0.7)
10. If (water is extremely_dirty) then (wash_time is short) (0.1)
11. If (water is extremely_dirty) then (wash_time is medium) (0.1)
12. If (water is extremely_dirty) then (wash_time is long) (1)

Rule Format: verbose

FIS Name: WASH1

Figure A1 - 4 : Rule Editor for Basic Wash Cycle

```
[System]
Name='WASH1'
Type='mamdani'
NumInputs=1
NumOutputs=1
NumRules=12
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='water'
Range=[0 100]
NumMFs=4
MF1='somewhat_dirty':'trapmf',[-6.5 0 20 47.5]
MF2='dirty':'trimf',[20 50 75]
MF3='very_dirty':'trimf',[50 75 100]
MF4='extremely_dirty':'trimf',[75 100 125]

[Output1]
Name='wash_time'
Range=[-5 60]
NumMFs=3
MF1='short':'trimf',[-37.5 -5 27.5]
MF2='medium':'trimf',[-5 27.5 60]
MF3='long':'trimf',[27.5 60 92.5]

[Rules]
1, 1 (1) : 1
1, 2 (0) : 1
1, 3 (0) : 1
2, 1 (0.75) : 1
2, 2 (0.1) : 1
2, 3 (0.1) : 1
3, 1 (0.1) : 1
3, 2 (0.5) : 1
3, 3 (0.7) : 1
4, 1 (0.1) : 1
4, 2 (0.1) : 1
4, 3 (1) : 1
```

Listing A1 - 1 : Program Listing for Basic Wash Cycle

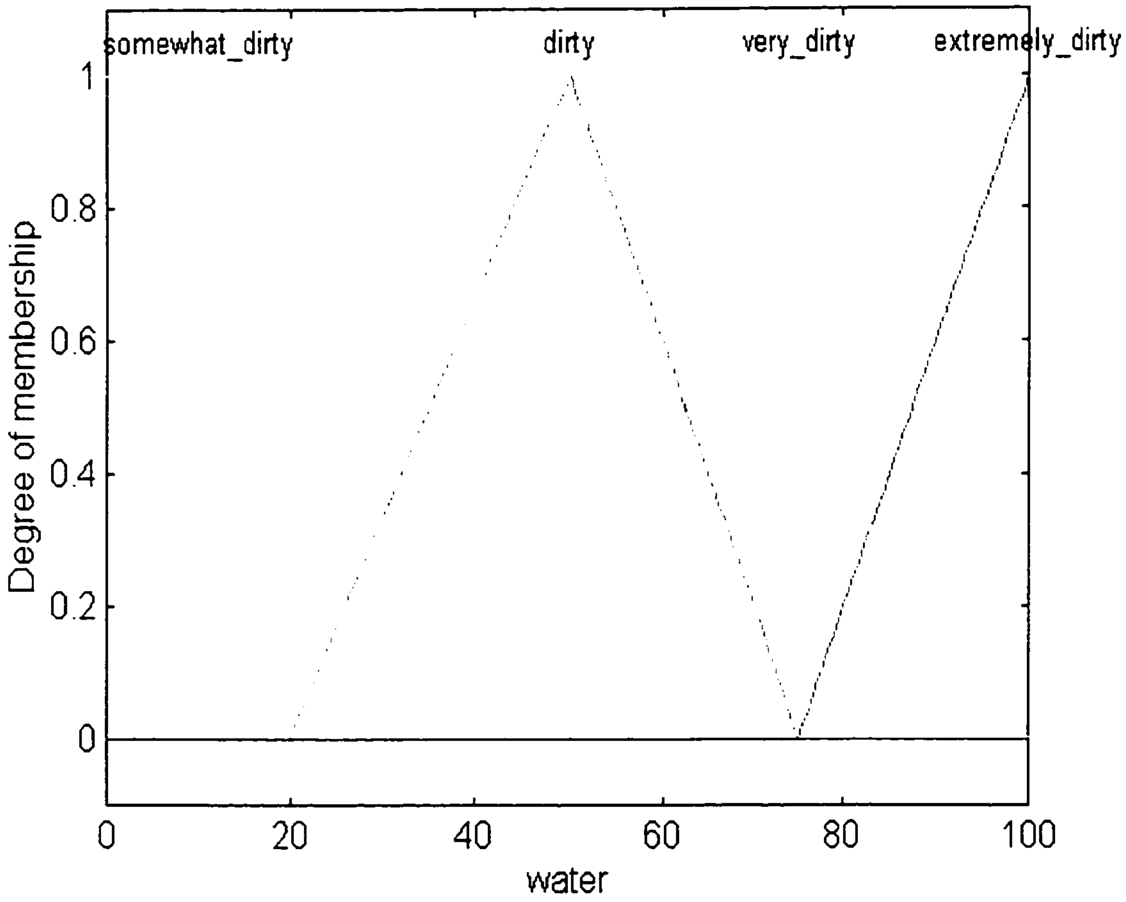


Figure A1 - 5 : MATLAB Output for Model Height Input

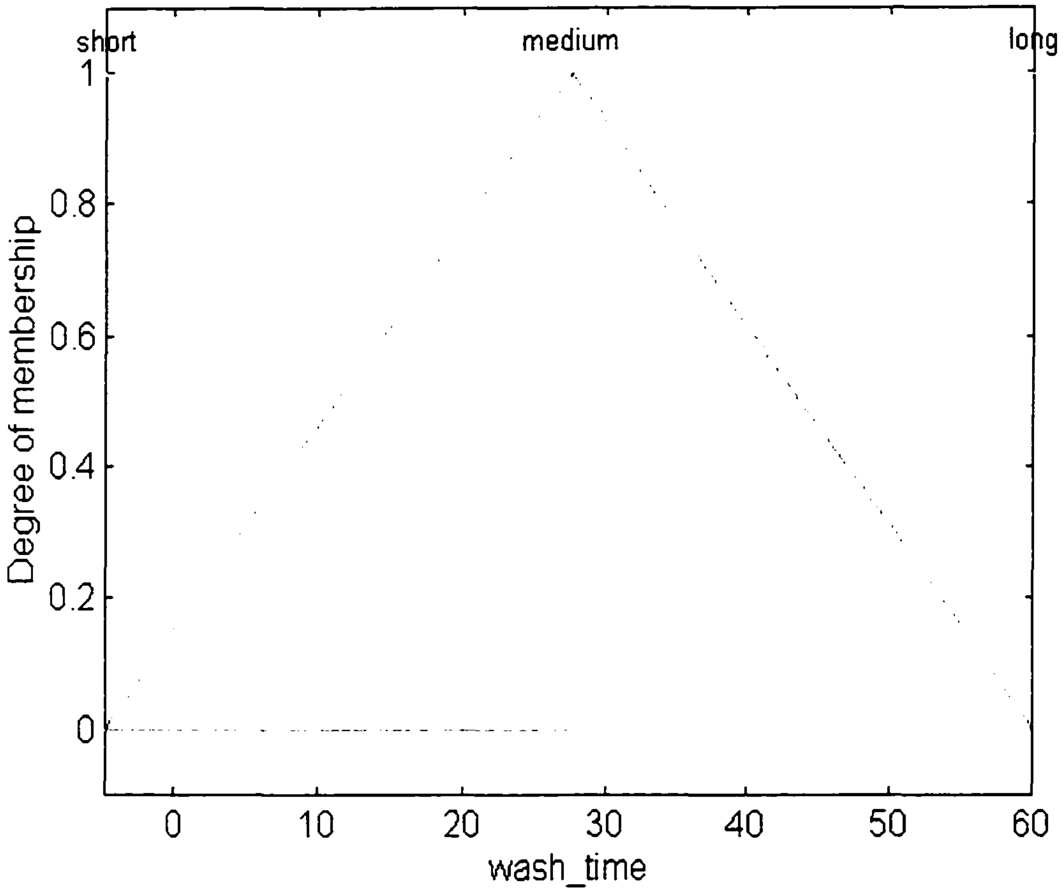


Figure A1 - 6 : MATLAB Output for Wash Time Output

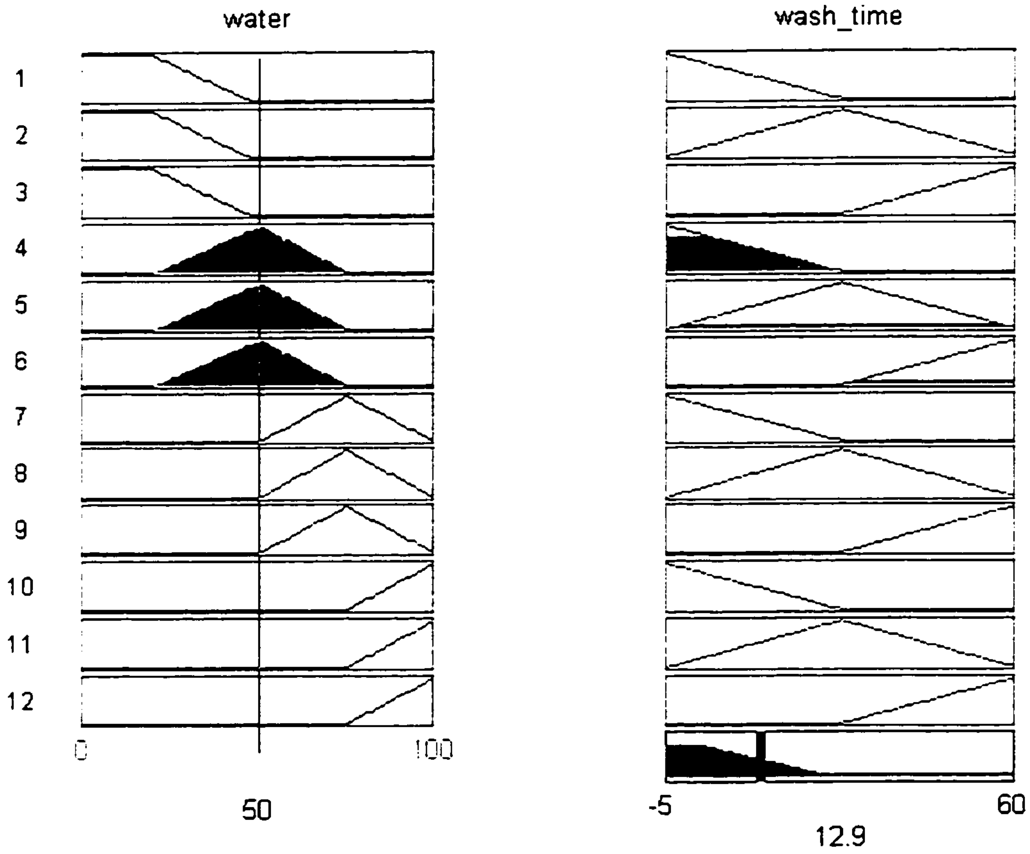
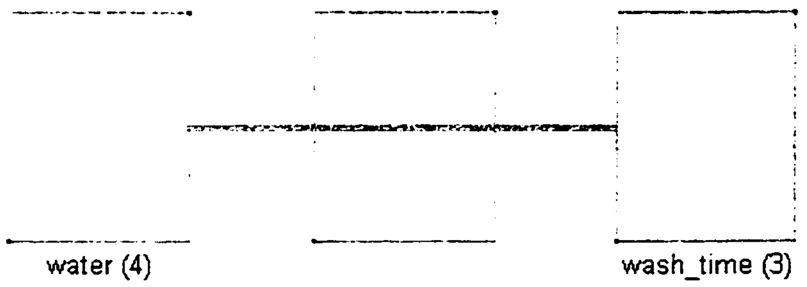


Figure A1 - 7 : Rule View for Basic Wash Cycle



System WASH1: 1 inputs, 1 outputs, 12 rules

Figure A1 - 8 : System Plot for Basic Wash Cycle

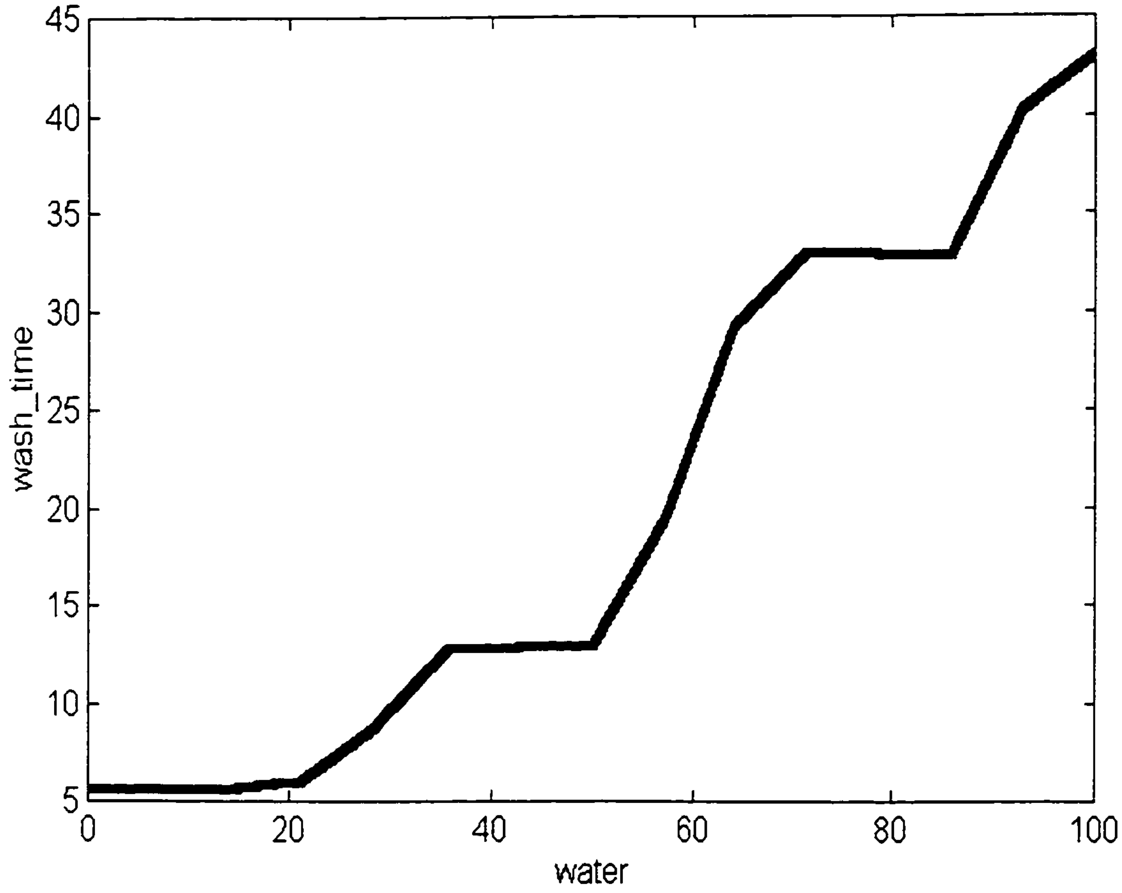
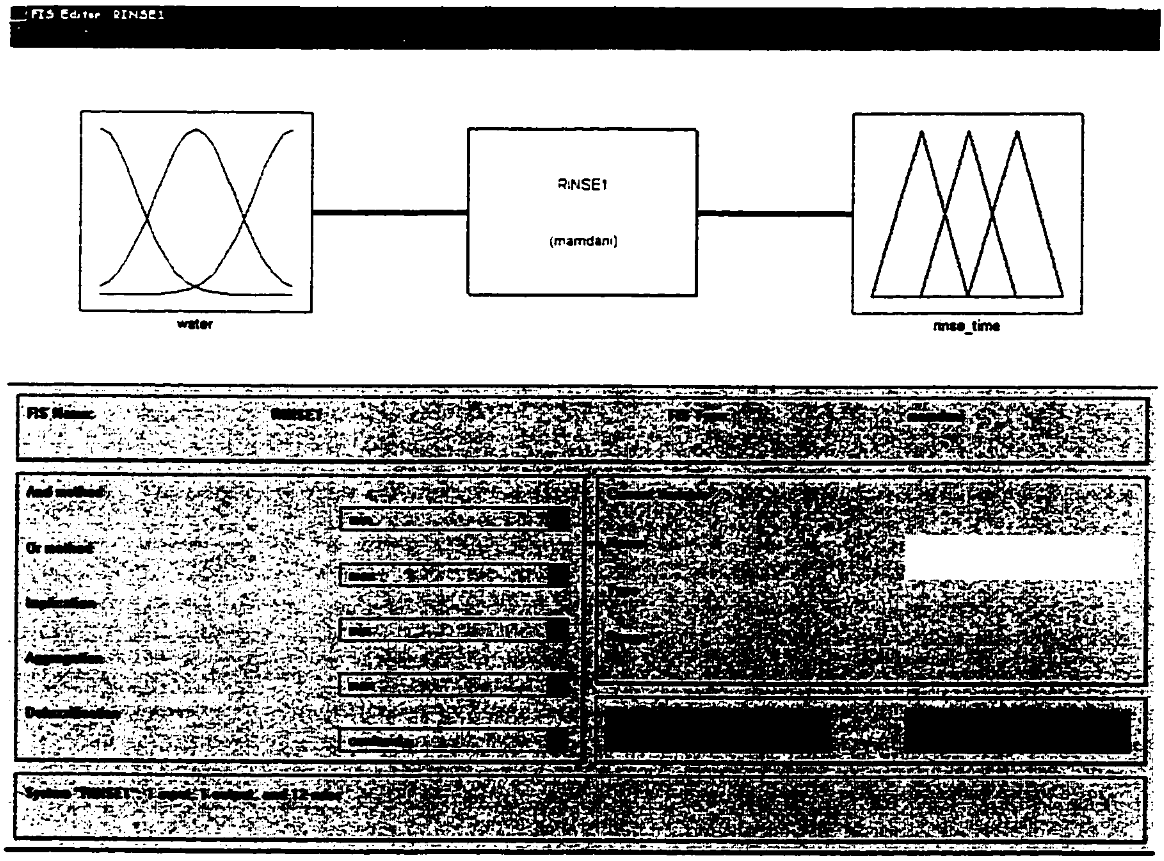


Figure A1 - 9 : Surface View for Basic Wash Cycle

APPENDIX 2

Basic Fuzzy Rinse Cycle Control



FIS Name: RINSE1

FIS Type: mamdani

And method: min

Or method: max

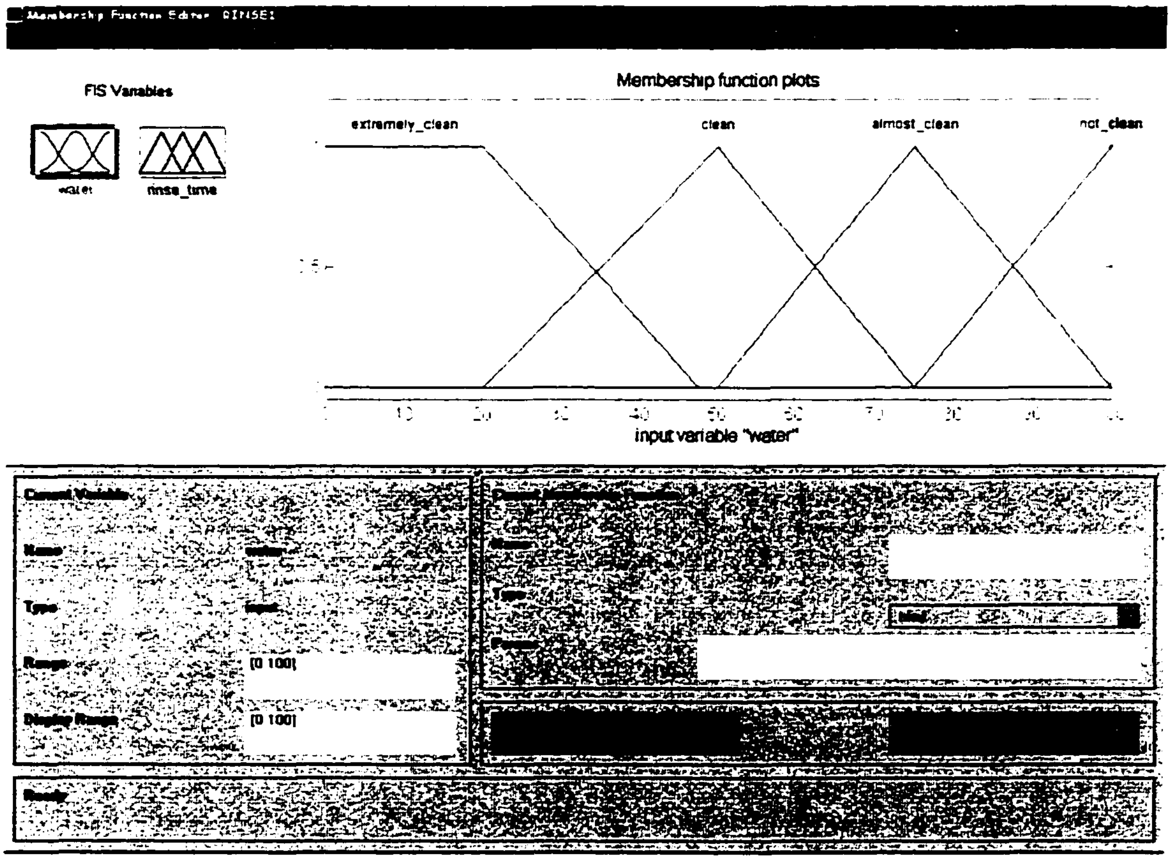
Implication: min

Aggregation: max

Defuzzification: centroid

System "RINSE1": 1 input, 1 output, and 12 rules

Figure A2 - 1 : FIS Editor for Basic Rinse Cycle



Current Variable

Name water

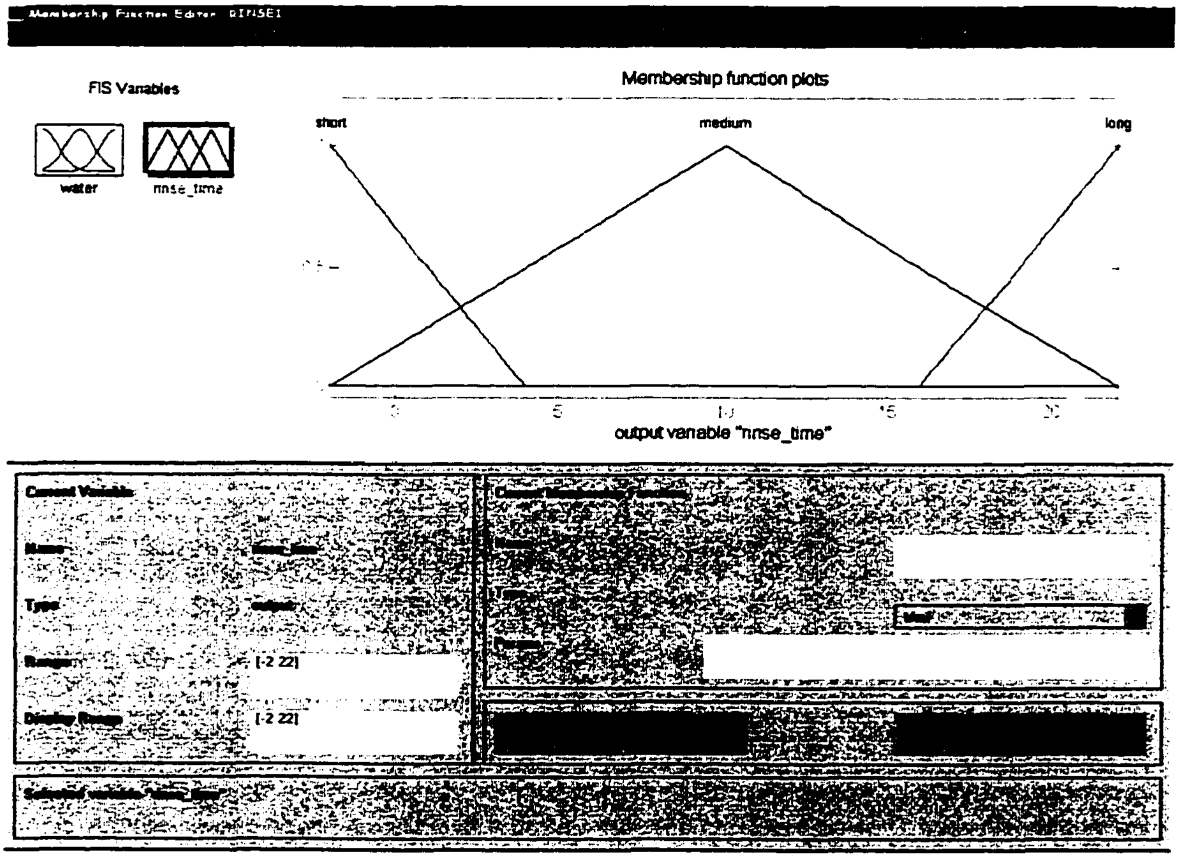
Type input

Range [0 100]

Display Range [0 100]

Ready

Figure A2 - 2 : Membership Function Editor for Water Input of Basic Rinse Cycle



Membership Function Editor: RINSE1
FIS Variables
water
rinse_time
Membership function plots
short
medium
long
0.5
0
5
10
15
20
output variable "rinse_time"

Current Variable

Name rinse_time

Type output

Range [-2 22]

Display Range [-2 22]

Current Membership Function

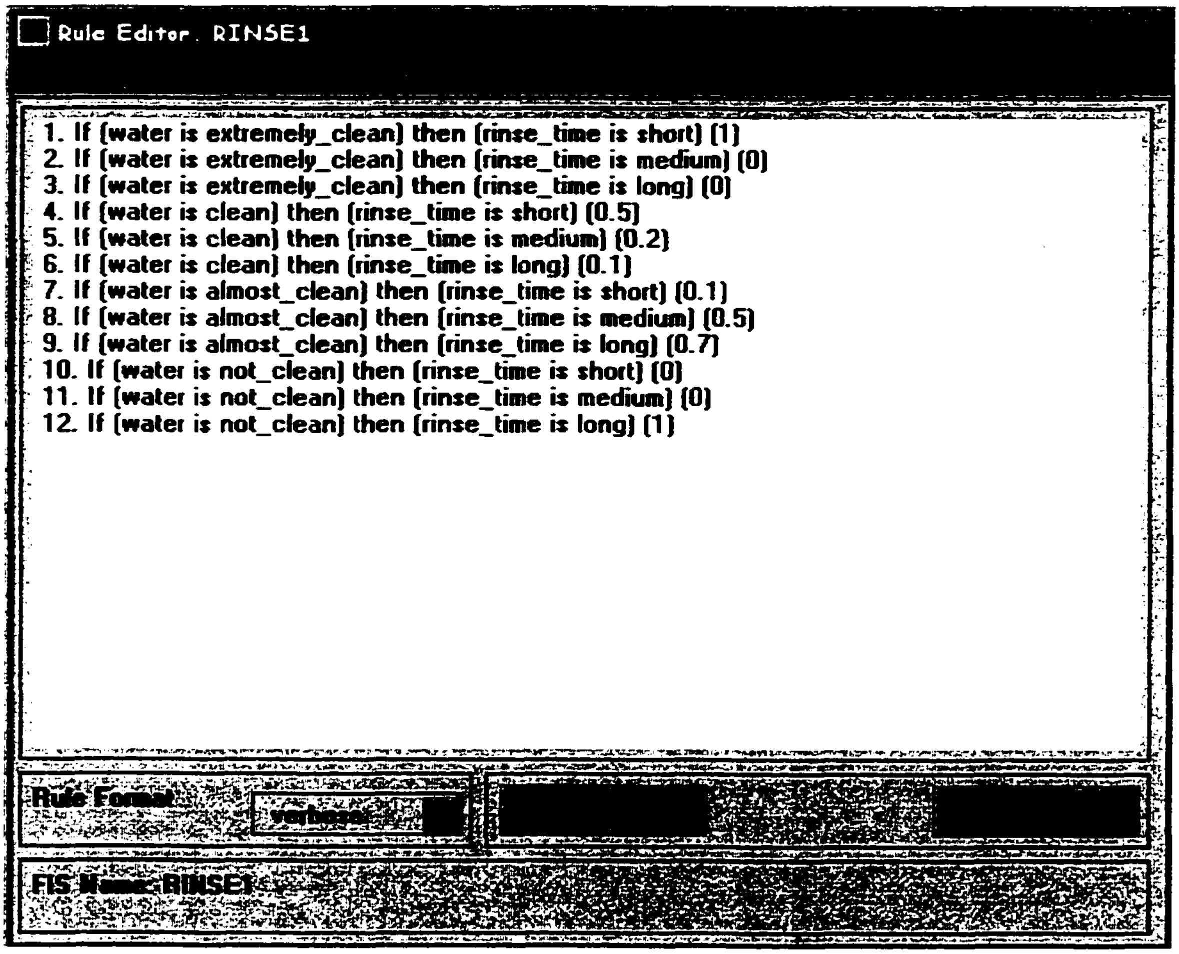
Name

Type

Params

Selected variable "rinse_time"

Figure A2 - 3 : Membership Function Editor for Rinse Time Output of Basic Rinse Cycle



1. If (water is extremely_clean) then (rinse_time is short) (1)
2. If (water is extremely_clean) then (rinse_time is medium) (0)
3. If (water is extremely_clean) then (rinse_time is long) (0)
4. If (water is clean) then (rinse_time is short) (0.5)
5. If (water is clean) then (rinse_time is medium) (0.2)
6. If (water is clean) then (rinse_time is long) (0.1)
7. If (water is almost_clean) then (rinse_time is short) (0.1)
8. If (water is almost_clean) then (rinse_time is medium) (0.5)
9. If (water is almost_clean) then (rinse_time is long) (0.7)
10. If (water is not_clean) then (rinse_time is short) (0)
11. If (water is not_clean) then (rinse_time is medium) (0)
12. If (water is not_clean) then (rinse_time is long) (1)

Figure A2 -4 : Rule Editor for Basic Rinse Cycle

```
[System]
Name='RINSE1'
Type='mamdani'
NumInputs=1
NumOutputs=1
NumRules=12
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='water'
Range=[0 100]
NumMFs=4
MF1='extremely_clean':'trapmf',[-47.5 -20 20 47.5]
MF2='clean':'trimf',[20 50 75]
MF3='almost_clean':'trimf',[50 75 100]
MF4='not_clean':'trimf',[75 100 125]

[Output1]
Name='rinse_time'
Range=[-2 22]
NumMFs=3
MF1='short':'trapmf',[-24.79 -14 -2 4]
MF2='medium':'trimf',[-2 10 22]
MF3='long':'trimf',[15.99 22 34]

[Rules]
1, 1 (1) : 1
1, 2 (0) : 1
1, 3 (0) : 1
2, 1 (0.5) : 1
2, 2 (0.2) : 1
2, 3 (0.1) : 1
3, 1 (0.1) : 1
3, 2 (0.5) : 1
3, 3 (0.7) : 1
4, 1 (0) : 1
4, 2 (0) : 1
4, 3 (1) : 1
```

Listing A2 - 1 : Program Listing for Basic Rinse Cycle

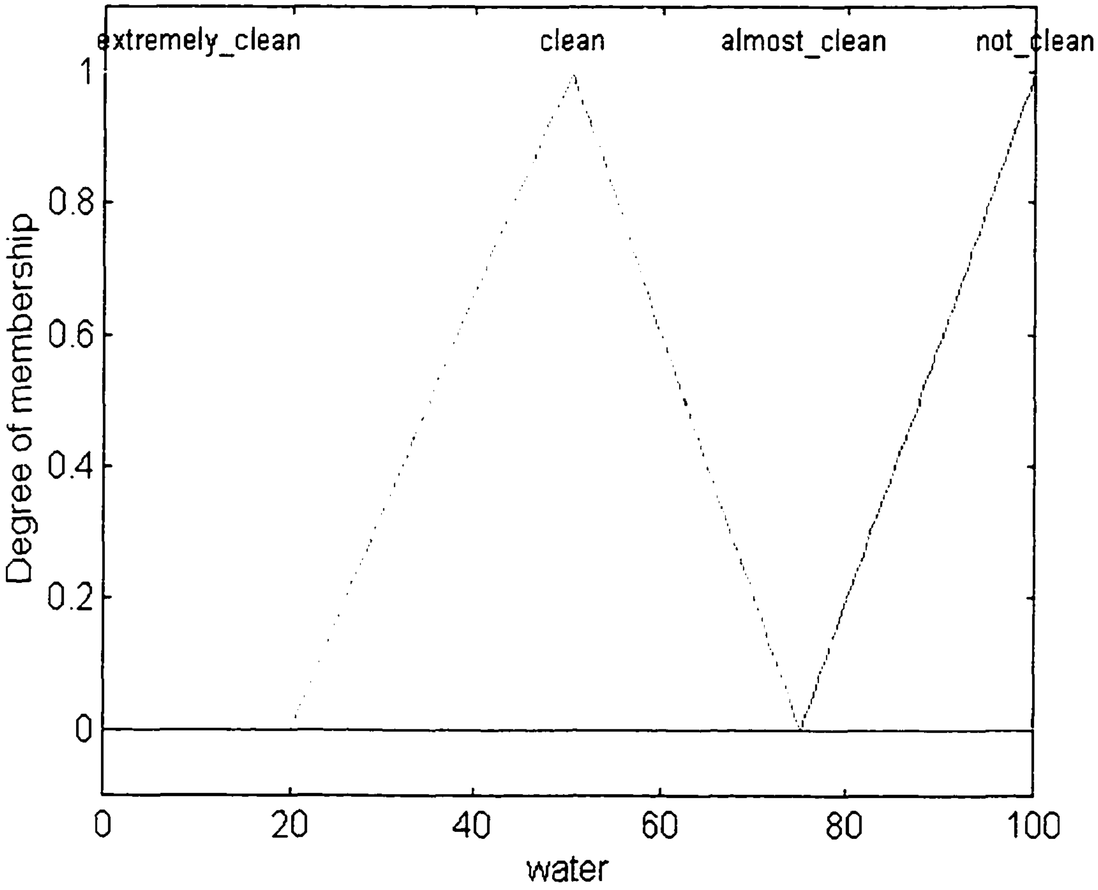


Figure A2 - 5 : MATLAB Output for Basic Rinse Cycle Input

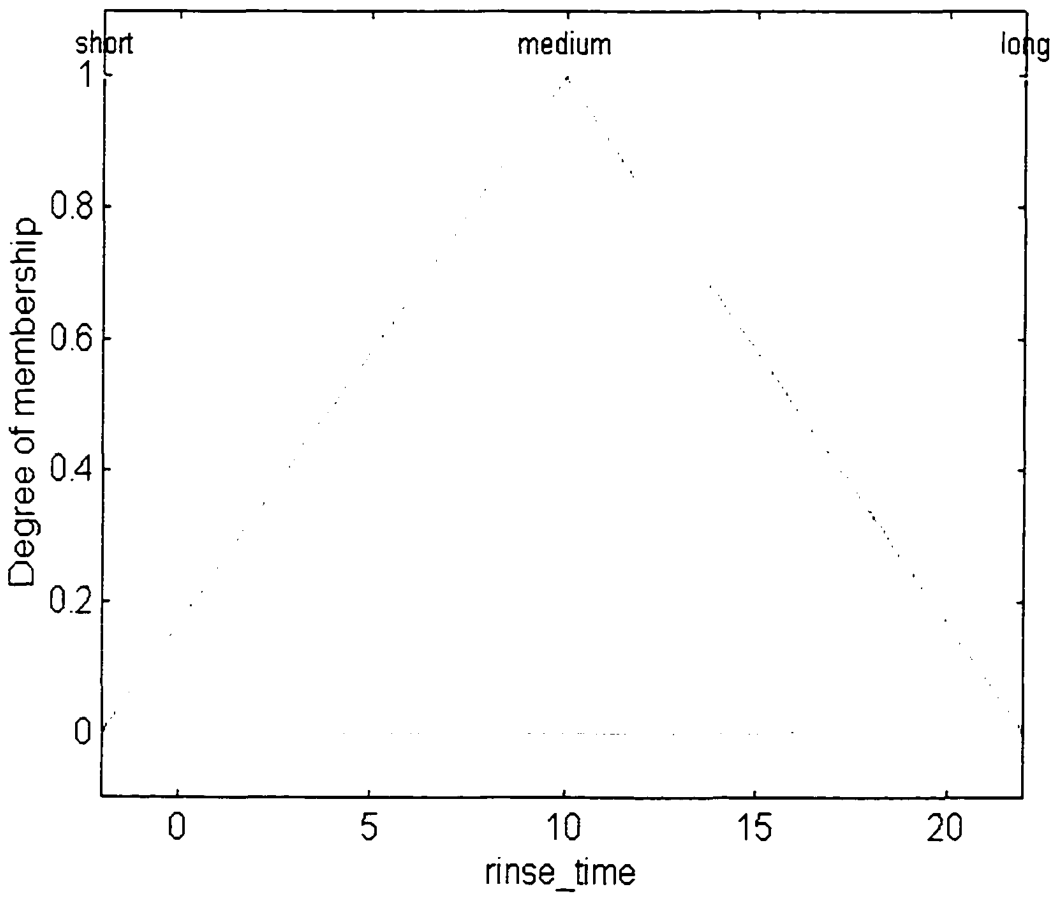


Figure A2 - 6 : MATLAB Output for Basic Rinse Cycle Output

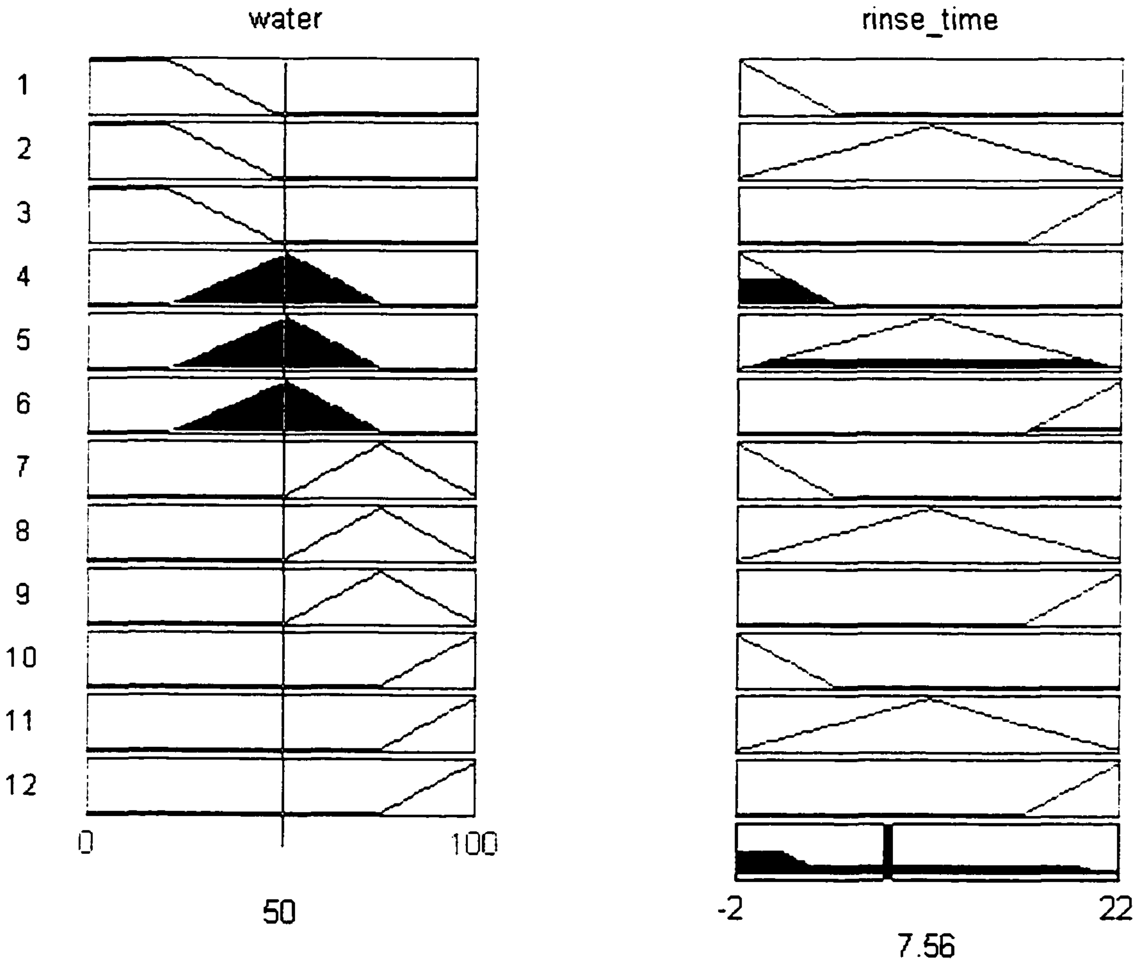
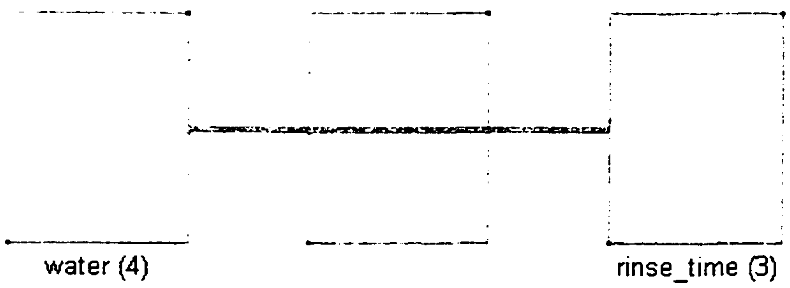


Figure A2 - 7 : Rule View for Basic Rinse Cycle



System RINSE1: 1 inputs, 1 outputs, 12 rules

Figure A2 - 8 : System Plot for Basic Rinse Cycle

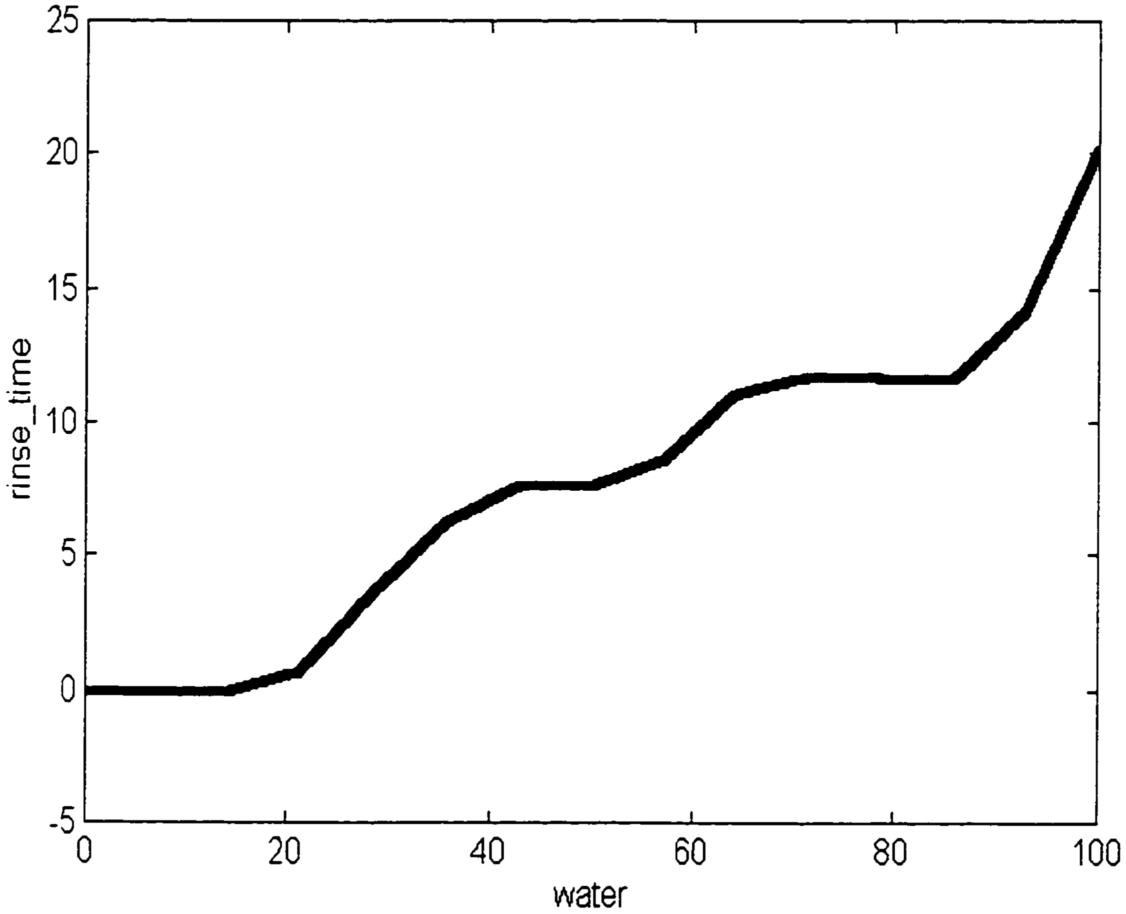
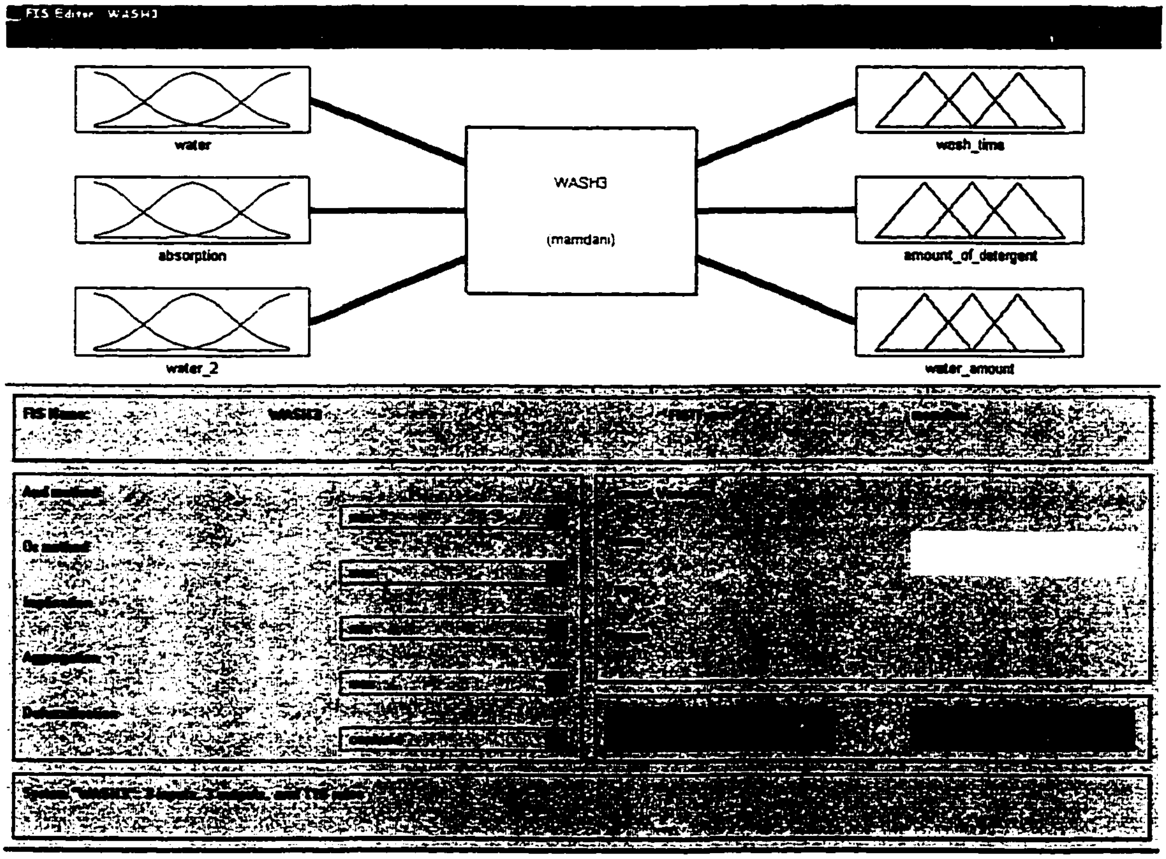


Figure A2 - 9 : Surface View for Basic Rinse Cycle

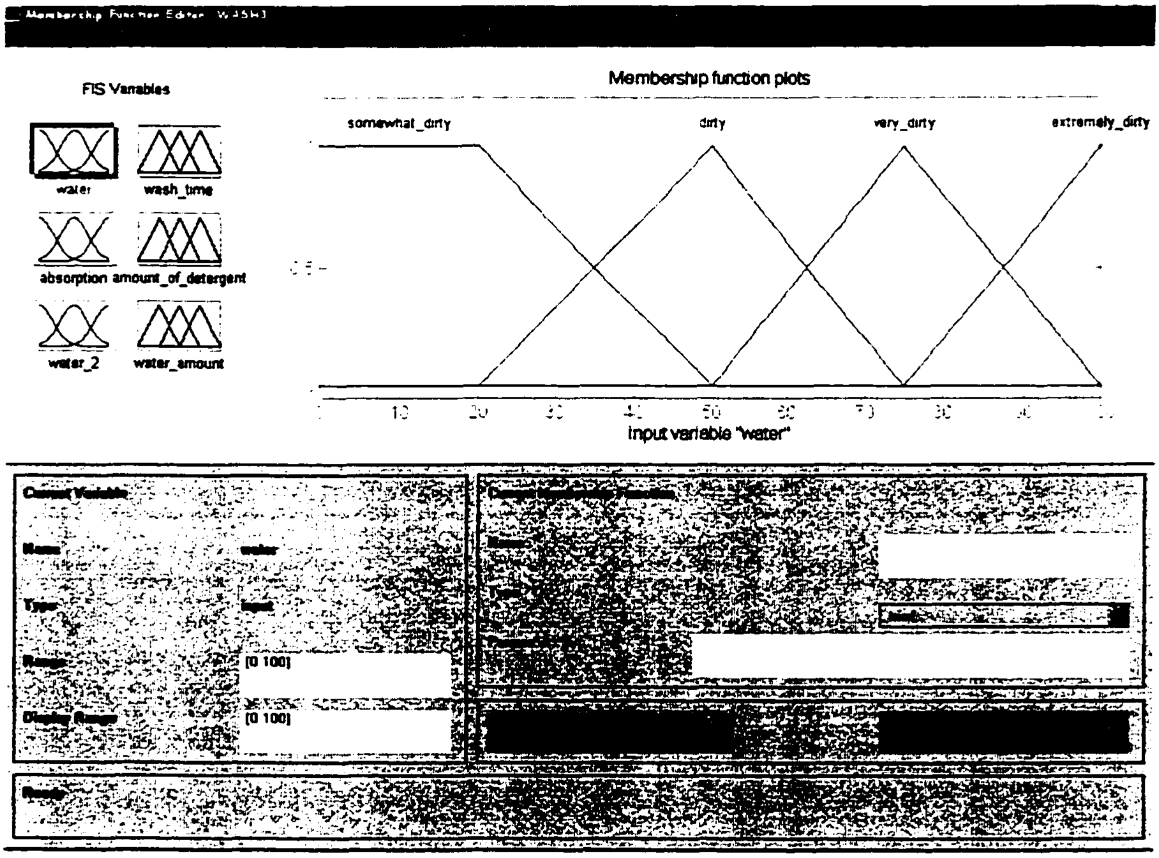
APPENDIX 3

Deluxe Fuzzy Wash Cycle Control



FIS Name: WASH3

Figure A3 - 1 : FIS Editor for Deluxe Wash Cycle



Membership Function Editor: WASH3

Current Variable

Name water

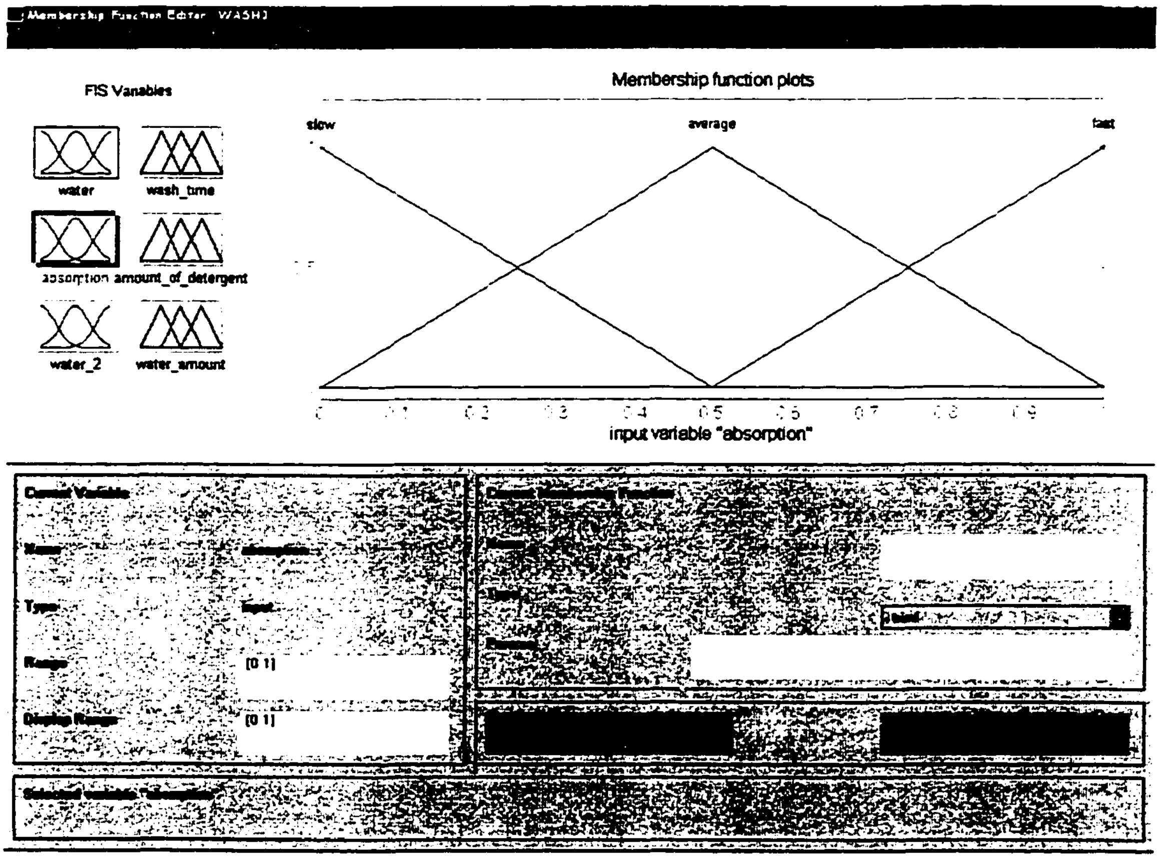
Type input

Range [0 100]

Display Range [0 100]

Ready

Figure A3 - 2 : Membership Function Editor for Water Input of Deluxe Wash Cycle



Current Variable

Name absorption

Type input

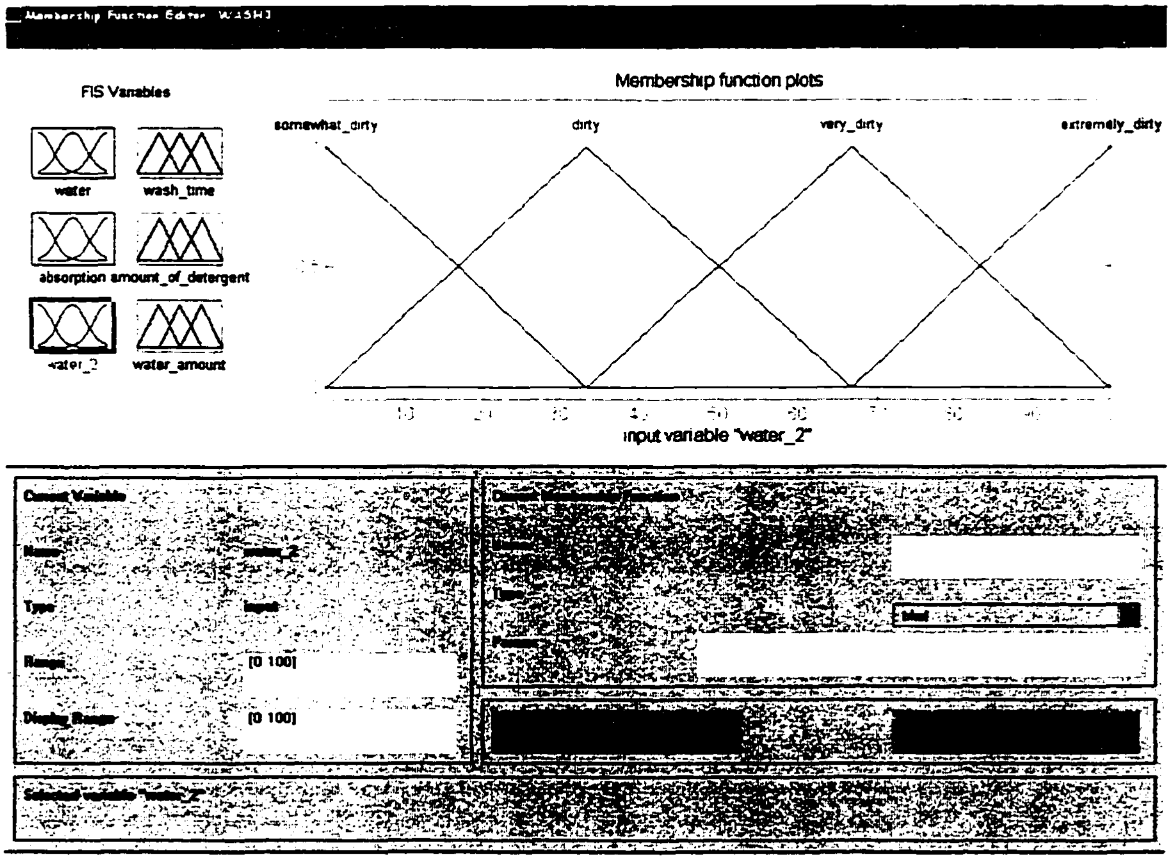
Range [0 1]

Display Range [0 1]

Current Membership Function

Selected variable "absorption"

Figure A3 - 3 : Membership Function Editor for Absorption Input of Deluxe Wash Cycle



Current Variable

Name water_2

Type input

Range [0 100]

Display Range [0 100]

Current Membership Function

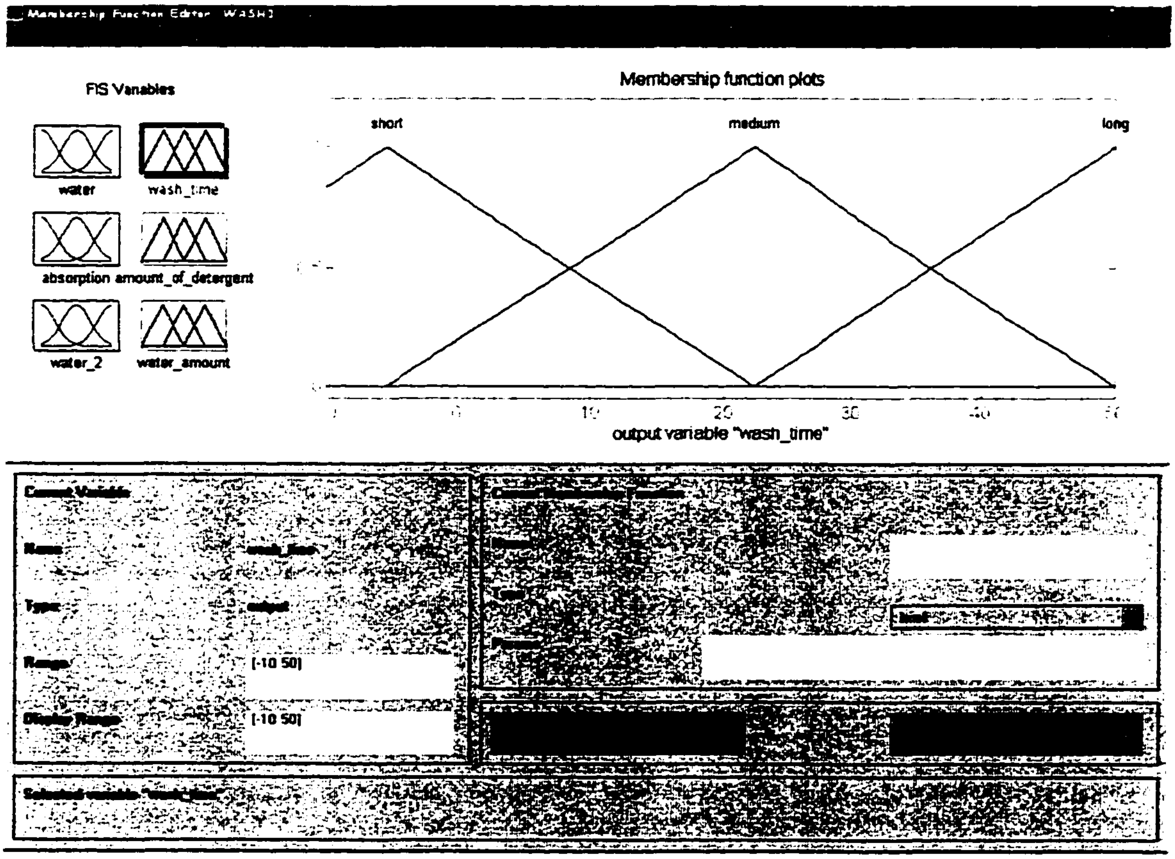
Name

Type trimf

Params

Selected variable "water_2"

Figure A3 - 4 : Membership Function Editor for Water Input of Deluxe Wash Cycle



Current Variable

Name wash_time

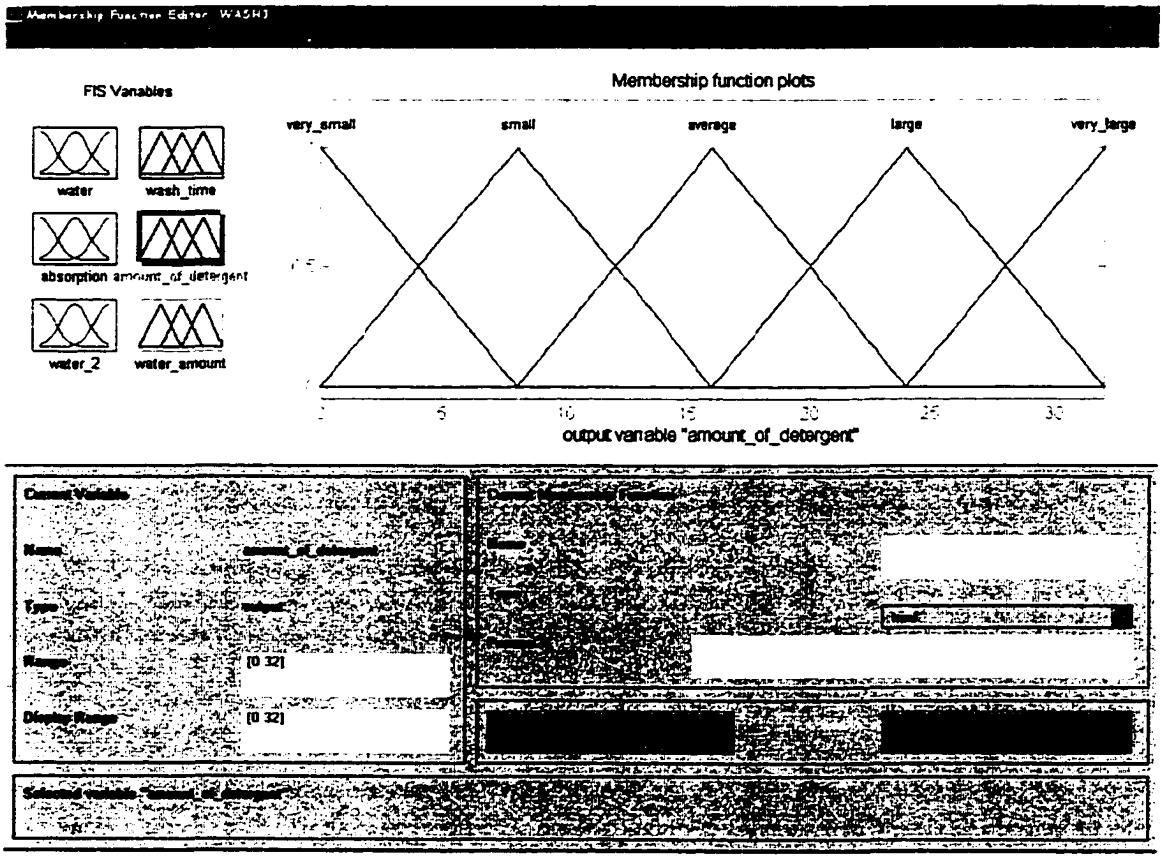
Type output

Range [-10 50]

Display Range [-10 50]

Selected variable "wash_time"

Figure A3 - 5 : Membership Function Editor for Wash Time Output of Deluxe Wash Cycle



Current Variable

Name amount_of_detergent

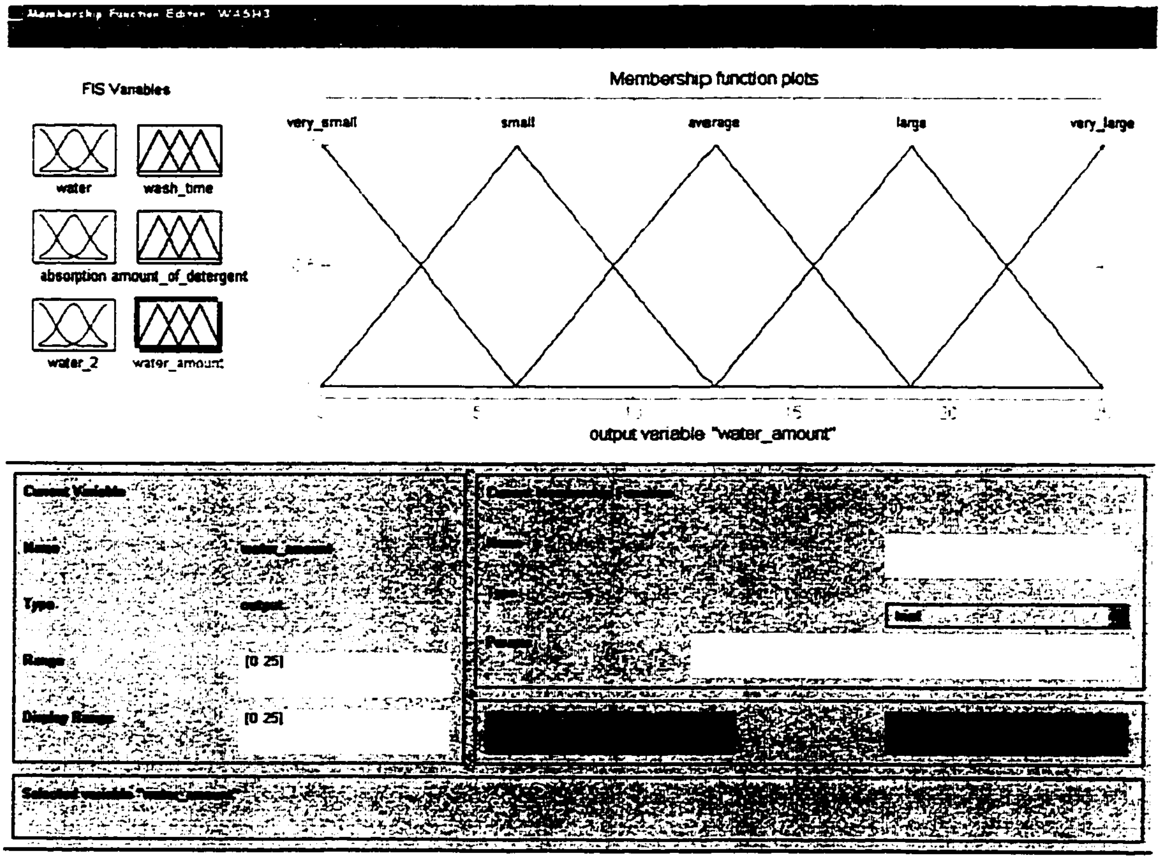
Type output

Range [0 32]

Display Range [0 32]

Selected variable "amount_of_detergent"

Figure A3 - 6 : Membership Function Editor for Amount of Detergent Output of Deluxe Wash Cycle



Current Variable

Name water_amount

Type output

Range [0 25]

Display Range [0 25]

Current Membership Function

Name

Type trimf

Params

Selected variable "water_amount"

Figure A3 - 7 : Membership Function Editor for Water Amount Output of Deluxe Wash Cycle

1. If (water is somewhat_dirty) and (absorption is slow) then (wash_time is short) (1)
2. If (water is somewhat_dirty) and (absorption is slow) then (wash_time is medium) (1)
3. If (water is somewhat_dirty) and (absorption is slow) then (wash_time is long) (0)
4. If (water is somewhat_dirty) and (absorption is average) then (wash_time is short) (1)
5. If (water is somewhat_dirty) and (absorption is average) then (wash_time is medium) (0)
6. If (water is somewhat_dirty) and (absorption is average) then (wash_time is long) (0)
7. If (water is somewhat_dirty) and (absorption is fast) then (wash_time is short) (1)
8. If (water is somewhat_dirty) and (absorption is fast) then (wash_time is medium) (0)
9. If (water is somewhat_dirty) and (absorption is fast) then (wash_time is long) (0)
10. If (water is dirty) and (absorption is slow) then (wash_time is short) (0)
11. If (water is dirty) and (absorption is slow) then (wash_time is medium) (1)
12. If (water is dirty) and (absorption is slow) then (wash_time is long) (0)
13. If (water is dirty) and (absorption is average) then (wash_time is short) (1)
14. If (water is dirty) and (absorption is average) then (wash_time is medium) (1)
15. If (water is dirty) and (absorption is average) then (wash_time is long) (0)
16. If (water is dirty) and (absorption is fast) then (wash_time is short) (1)
17. If (water is dirty) and (absorption is fast) then (wash_time is medium) (0)
18. If (water is dirty) and (absorption is fast) then (wash_time is long) (0)
19. If (water is very_dirty) and (absorption is slow) then (wash_time is short) (0)
20. If (water is very_dirty) and (absorption is slow) then (wash_time is medium) (0)
21. If (water is very_dirty) and (absorption is slow) then (wash_time is long) (1)
22. If (water is very_dirty) and (absorption is average) then (wash_time is short) (0)
23. If (water is very_dirty) and (absorption is average) then (wash_time is medium) (1)
24. If (water is very_dirty) and (absorption is average) then (wash_time is long) (0)
25. If (water is very_dirty) and (absorption is fast) then (wash_time is short) (0)
26. If (water is very_dirty) and (absorption is fast) then (wash_time is medium) (1)
27. If (water is very_dirty) and (absorption is fast) then (wash_time is long) (0)
28. If (water is extremely_dirty) and (absorption is slow) then (wash_time is short) (0)
29. If (water is extremely_dirty) and (absorption is slow) then (wash_time is medium) (0)
30. If (water is extremely_dirty) and (absorption is slow) then (wash_time is long) (1)
31. If (water is extremely_dirty) and (absorption is average) then (wash_time is short) (0)
32. If (water is extremely_dirty) and (absorption is average) then (wash_time is medium) (0)
33. If (water is extremely_dirty) and (absorption is average) then (wash_time is long) (1)
34. If (water is extremely_dirty) and (absorption is fast) then (wash_time is short) (0)
35. If (water is extremely_dirty) and (absorption is fast) then (wash_time is medium) (1)
36. If (water is extremely_dirty) and (absorption is fast) then (wash_time is long) (0)
37. If (water_2 is somewhat_dirty) then (amount_of_detergent is very_small) (1)
38. If (water_2 is somewhat_dirty) then (amount_of_detergent is small) (0)
39. If (water_2 is somewhat_dirty) then (amount_of_detergent is average) (0)
40. If (water_2 is somewhat_dirty) then (amount_of_detergent is large) (0)
41. If (water_2 is somewhat_dirty) then (amount_of_detergent is very_large) (0)
42. If (water_2 is dirty) then (amount_of_detergent is very_small) (0)
43. If (water_2 is dirty) then (amount_of_detergent is small) (1)
44. If (water_2 is dirty) then (amount_of_detergent is average) (0)
45. If (water_2 is dirty) then (amount_of_detergent is large) (0)
46. If (water_2 is dirty) then (amount_of_detergent is very_large) (0)
47. If (water_2 is very_dirty) then (amount_of_detergent is very_small) (1)
48. If (water_2 is very_dirty) then (amount_of_detergent is smali) (0)
49. If (water_2 is very_dirty) then (amount_of_detergent is average) (0)
50. If (water_2 is very_dirty) then (amount_of_detergent is large) (1)
51. If (water_2 is very_dirty) then (amount_of_detergent is very_large) (0)
52. If (water_2 is extremely_dirty) then (amount_of_detergent is very_small) (0)
53. If (water_2 is extremely_dirty) then (amount_of_detergent is small) (0)

54. If (water_2 is extremely_dirty) then (amount_of_detergent is average) (0)
55. If (water_2 is extremely_dirty) then (amount_of_detergent is large) (0)
56. If (water_2 is extremely_dirty) then (amount_of_detergent is very_large) (1)
57. If (water is somewhat_dirty) and (absorption is slow) then (water_amount is very_small) (0)
58. If (water is somewhat_dirty) and (absorption is slow) then (water_amount is small) (0)
59. If (water is somewhat_dirty) and (absorption is slow) then (water_amount is average) (1)
60. If (water is somewhat_dirty) and (absorption is slow) then (water_amount is large) (0)
61. If (water is somewhat_dirty) and (absorption is slow) then (water_amount is very_large) (0)
62. If (water is somewhat_dirty) and (absorption is average) then (water_amount is very_small) (0)
63. If (water is somewhat_dirty) and (absorption is average) then (water_amount is small) (1)
64. If (water is somewhat_dirty) and (absorption is average) then (water_amount is average) (0)
65. If (water is somewhat_dirty) and (absorption is average) then (water_amount is large) (0)
66. If (water is somewhat_dirty) and (absorption is average) then (water_amount is very_large) (0)
67. If (water is somewhat_dirty) and (absorption is fast) then (water_amount is very_small) (1)
68. If (water is somewhat_dirty) and (absorption is fast) then (water_amount is small) (0)
69. If (water is somewhat_dirty) and (absorption is fast) then (water_amount is average) (0)
70. If (water is somewhat_dirty) and (absorption is fast) then (water_amount is large) (0)
71. If (water is somewhat_dirty) and (absorption is fast) then (water_amount is very_large) (0)
72. If (water is dirty) and (absorption is slow) then (water_amount is very_small) (0)
73. If (water is dirty) and (absorption is slow) then (water_amount is small) (0)
74. If (water is dirty) and (absorption is slow) then (water_amount is average) (1)
75. If (water is dirty) and (absorption is slow) then (water_amount is large) (0)
76. If (water is dirty) and (absorption is slow) then (water_amount is very_large) (0)
77. If (water is dirty) and (absorption is average) then (water_amount is very_small) (0)
78. If (water is dirty) and (absorption is average) then (water_amount is small) (0)
79. If (water is dirty) and (absorption is average) then (water_amount is average) (1)
80. If (water is dirty) and (absorption is average) then (water_amount is large) (0)
81. If (water is dirty) and (absorption is average) then (water_amount is very_large) (0)
82. If (water is dirty) and (absorption is fast) then (water_amount is very_small) (0)
83. If (water is dirty) and (absorption is fast) then (water_amount is small) (1)
84. If (water is dirty) and (absorption is fast) then (water_amount is average) (0)
85. If (water is dirty) and (absorption is fast) then (water_amount is large) (0)
86. If (water is dirty) and (absorption is fast) then (water_amount is very_large) (0)
87. If (water is very_dirty) and (absorption is slow) then (water_amount is very_small) (0)
88. If (water is very_dirty) and (absorption is slow) then (water_amount is small) (0)
89. If (water is very_dirty) and (absorption is slow) then (water_amount is average) (0)
90. If (water is very_dirty) and (absorption is slow) then (water_amount is large) (1)
91. If (water is very_dirty) and (absorption is slow) then (water_amount is very_large) (0)
92. If (water is very_dirty) and (absorption is average) then (water_amount is very_small) (0)
93. If (water is very_dirty) and (absorption is average) then (water_amount is small) (0)
94. If (water is very_dirty) and (absorption is average) then (water_amount is average) (0)
95. If (water is very_dirty) and (absorption is average) then (water_amount is large) (1)
96. If (water is very_dirty) and (absorption is average) then (water_amount is very_large) (0)
97. If (water is very_dirty) and (absorption is fast) then (water_amount is very_small) (0)
98. If (water is very_dirty) and (absorption is fast) then (water_amount is small) (0)
99. If (water is very_dirty) and (absorption is fast) then (water_amount is average) (1)
100. If (water is very_dirty) and (absorption is fast) then (water_amount is large) (0)
101. If (water is very_dirty) and (absorption is fast) then (water_amount is very_large) (0)
102. If (water is extremely_dirty) and (absorption is slow) then (water_amount is very_small) (0)
103. If (water is extremely_dirty) and (absorption is slow) then (water_amount is small) (0)
104. If (water is extremely_dirty) and (absorption is slow) then (water_amount is average) (0)
105. If (water is extremely_dirty) and (absorption is slow) then (water_amount is large) (0)
106. If (water is extremely_dirty) and (absorption is slow) then (water_amount is very_large) (1)

107. If (water is extremely_dirty) and (absorption is average) then (water_amount is very_small) (0)
108. If (water is extremely_dirty) and (absorption is average) then (water_amount is small) (0)
109. If (water is extremely_dirty) and (absorption is average) then (water_amount is average) (0)
110. If (water is extremely_dirty) and (absorption is average) then (water_amount is large) (1)
111. If (water is extremely_dirty) and (absorption is average) then (water_amount is very_large) (0)
112. If (water is extremely_dirty) and (absorption is fast) then (water_amount is very_small) (0)
113. If (water is extremely_dirty) and (absorption is fast) then (water_amount is small) (0)
114. If (water is extremely_dirty) and (absorption is fast) then (water_amount is average) (1)
115. If (water is extremely_dirty) and (absorption is fast) then (water_amount is large) (0)
116. If (water is extremely_dirty) and (absorption is fast) then (water_amount is very_large) (0)

Listing A3 - 1 : Rule List for Deluxe Wash Cycle

```
[System]
Name='WASH3'
Type='mamdani'
NumInputs=3
NumOutputs=3
NumRules=116
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='water'
Range=[0 100]
NumMFs=4
MF1='somewhat_dirty':'trapmf',[-20 0 20 50]
MF2='dirty':'trimf',[20 50 75]
MF3='very_dirty':'trimf',[50 75 100]
MF4='extremely_dirty':'trapmf',[75 100 110 120]

[Input2]
Name='absorption'
Range=[0 1]
NumMFs=3
MF1='average':'trimf',[0 0.5 1]
MF2='slow':'trimf',[-0.5 0 0.5]
MF3='fast':'trimf',[0.5 1 1.5]

[Input3]
Name='water_2'
Range=[0 100]
NumMFs=4
MF1='somewhat_dirty':'trimf',[-33.33 4.441e-016 33.33]
MF2='dirty':'trimf',[0 33.33 66.67]
MF3='very_dirty':'trimf',[33.33 66.67 100]
MF4='extremely_dirty':'trimf',[66.67 100 133.3]

[Output1]
Name='wash_time'
Range=[-10 50]
NumMFs=3
MF1='short':'trimf',[-33.08 -5.385 22.32]
MF2='medium':'trimf',[-5.385 22.32 50]
MF3='long':'trimf',[22.32 50 77.66]

[Output2]
Name='amount_of_detergent'
Range=[0 32]
NumMFs=5
MF1='very_small':'trimf',[-8 0 8]
MF2='small':'trimf',[0 8 16]
MF3='average':'trimf',[8 16 24]
```

```
MF4='large':'trimf',[16 24 32]
MF5='very_large':'trimf',[24 32 40]

[Output3]
Name='water_amount'
Range=[0 25]
NumMFs=5
MF1='very_small':'trimf',[-6.25 0 6.25]
MF2='small':'trimf',[0 6.25 12.5]
MF3='average':'trimf',[6.25 12.5 18.75]
MF4='large':'trimf',[12.5 18.75 25]
MF5='very_large':'trimf',[18.75 25 31.25]

[Rules]
1 2 0, 1 0 0 (1) : 1
1 2 0, 2 0 0 (1) : 1
1 2 0, 3 0 0 (0) : 1
1 1 0, 1 0 0 (1) : 1
1 1 0, 2 0 0 (0) : 1
1 1 0, 3 0 0 (0) : 1
1 3 0, 1 0 0 (1) : 1
1 3 0, 2 0 0 (0) : 1
1 3 0, 3 0 0 (0) : 1
2 2 0, 1 0 0 (0) : 1
2 2 0, 2 0 0 (1) : 1
2 2 0, 3 0 0 (0) : 1
2 1 0, 1 0 0 (1) : 1
2 1 0, 2 0 0 (1) : 1
2 1 0, 3 0 0 (0) : 1
2 3 0, 1 0 0 (1) : 1
2 3 0, 2 0 0 (0) : 1
2 3 0, 3 0 0 (0) : 1
3 2 0, 1 0 0 (0) : 1
3 2 0, 2 0 0 (0) : 1
3 2 0, 3 0 0 (1) : 1
3 1 0, 1 0 0 (0) : 1
3 1 0, 2 0 0 (1) : 1
3 1 0, 3 0 0 (0) : 1
3 3 0, 1 0 0 (0) : 1
3 3 0, 2 0 0 (1) : 1
3 3 0, 3 0 0 (0) : 1
4 2 0, 1 0 0 (0) : 1
4 2 0, 2 0 0 (0) : 1
4 2 0, 3 0 0 (1) : 1
4 1 0, 1 0 0 (0) : 1
4 1 0, 2 0 0 (0) : 1
4 1 0, 3 0 0 (1) : 1
4 3 0, 1 0 0 (0) : 1
4 3 0, 2 0 0 (1) : 1
4 3 0, 3 0 0 (0) : 1
0 0 1, 0 1 0 (1) : 1
0 0 1, 0 2 0 (0) : 1
0 0 1, 0 3 0 (0) : 1
```

```
0 0 1, 0 4 0 (0) : 1
0 0 1, 0 5 0 (0) : 1
0 0 2, 0 1 0 (0) : 1
0 0 2, 0 2 0 (1) : 1
0 0 2, 0 3 0 (0) : 1
0 0 2, 0 4 0 (0) : 1
0 0 2, 0 5 0 (0) : 1
0 0 3, 0 1 0 (1) : 1
0 0 3, 0 2 0 (0) : 1
0 0 3, 0 3 0 (0) : 1
0 0 3, 0 4 0 (1) : 1
0 0 3, 0 5 0 (0) : 1
0 0 4, 0 1 0 (0) : 1
0 0 4, 0 2 0 (0) : 1
0 0 4, 0 3 0 (0) : 1
0 0 4, 0 4 0 (0) : 1
0 0 4, 0 5 0 (1) : 1
1 2 0, 0 0 1 (0) : 1
1 2 0, 0 0 2 (0) : 1
1 2 0, 0 0 3 (1) : 1
1 2 0, 0 0 4 (0) : 1
1 2 0, 0 0 5 (0) : 1
1 1 0, 0 0 1 (0) : 1
1 1 0, 0 0 2 (1) : 1
1 1 0, 0 0 3 (0) : 1
1 1 0, 0 0 4 (0) : 1
1 1 0, 0 0 5 (0) : 1
1 3 0, 0 0 1 (1) : 1
1 3 0, 0 0 2 (0) : 1
1 3 0, 0 0 3 (0) : 1
1 3 0, 0 0 4 (0) : 1
1 3 0, 0 0 5 (0) : 1
2 2 0, 0 0 1 (0) : 1
2 2 0, 0 0 2 (0) : 1
2 2 0, 0 0 3 (1) : 1
2 2 0, 0 0 4 (0) : 1
2 2 0, 0 0 5 (0) : 1
2 1 0, 0 0 1 (0) : 1
2 1 0, 0 0 2 (0) : 1
2 1 0, 0 0 3 (1) : 1
2 1 0, 0 0 4 (0) : 1
2 1 0, 0 0 5 (0) : 1
2 3 0, 0 0 1 (0) : 1
2 3 0, 0 0 2 (1) : 1
2 3 0, 0 0 3 (0) : 1
2 3 0, 0 0 4 (0) : 1
2 3 0, 0 0 5 (0) : 1
3 2 0, 0 0 1 (0) : 1
3 2 0, 0 0 2 (0) : 1
3 2 0, 0 0 3 (0) : 1
3 2 0, 0 0 4 (1) : 1
3 2 0, 0 0 5 (0) : 1
3 1 0, 0 0 1 (0) : 1
```

```
3 1 0. 0 0 2 (0) : 1
3 1 0. 0 0 3 (0) : 1
3 1 0. 0 0 4 (1) : 1
3 1 0. 0 0 5 (0) : 1
3 3 0. 0 0 1 (0) : 1
3 3 0. 0 0 2 (0) : 1
3 3 0. 0 0 3 (1) : 1
3 3 0. 0 0 4 (0) : 1
3 3 0. 0 0 5 (0) : 1
4 2 0. 0 0 1 (0) : 1
4 2 0. 0 0 2 (0) : 1
4 2 0. 0 0 3 (0) : 1
4 2 0. 0 0 4 (0) : 1
4 2 0. 0 0 5 (1) : 1
4 1 0. 0 0 1 (0) : 1
4 1 0. 0 0 2 (0) : 1
4 1 0. 0 0 3 (0) : 1
4 1 0. 0 0 4 (1) : 1
4 1 0. 0 0 5 (0) : 1
4 3 0. 0 0 1 (0) : 1
4 3 0. 0 0 2 (0) : 1
4 3 0. 0 0 3 (1) : 1
4 3 0. 0 0 4 (0) : 1
4 3 0. 0 0 5 (0) : 1
```

Listing A3 - 2 : Program Listing for Deluxe Wash Cycle

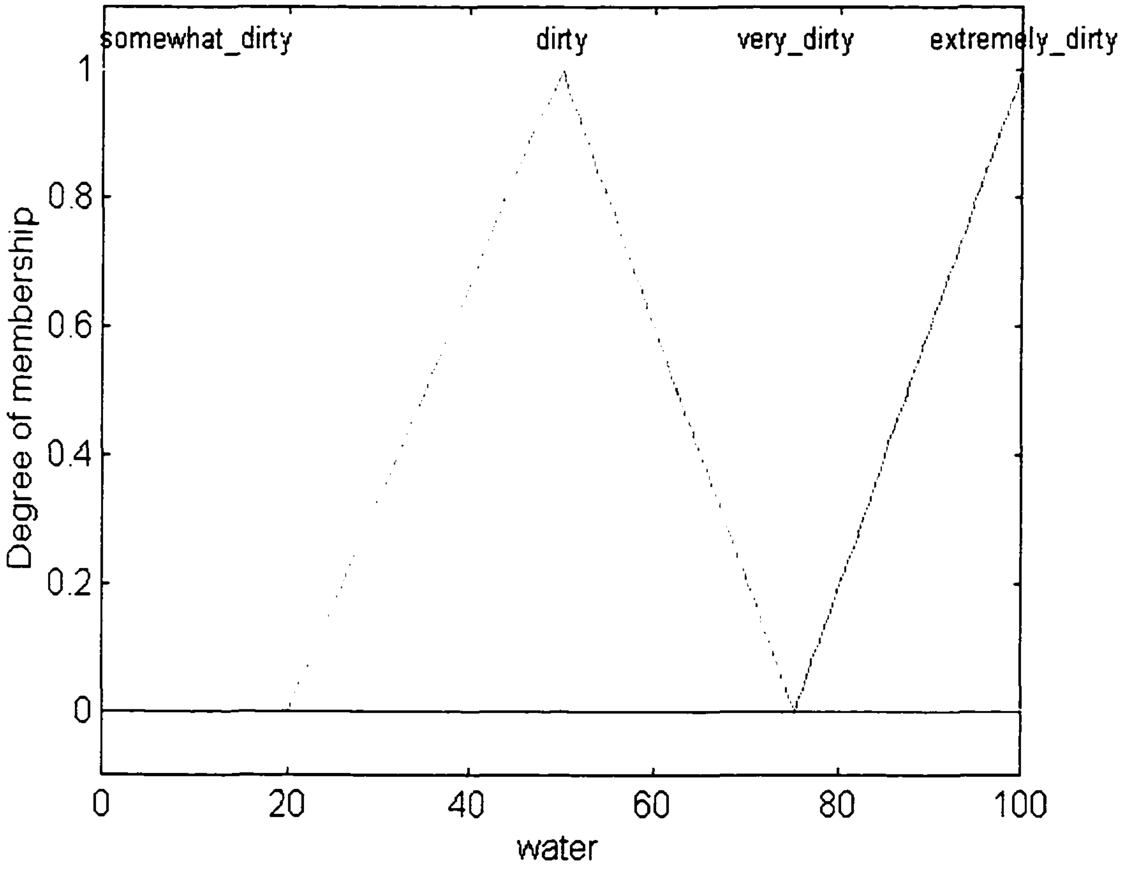


Figure A3 - 8 : MATLAB Output For Deluxe Wash Input, Water

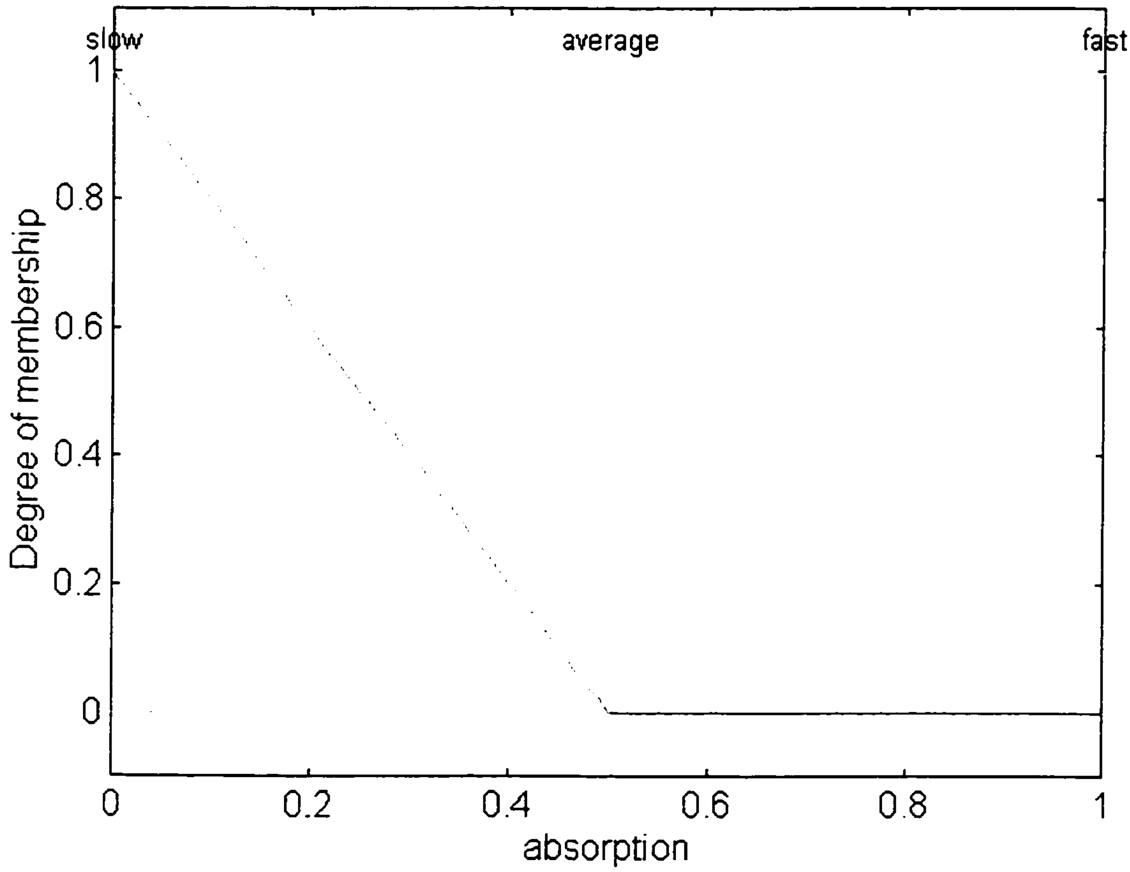


Figure A3 - 9 : MATLAB Output For Deluxe Wash Cycle Input, Absorption

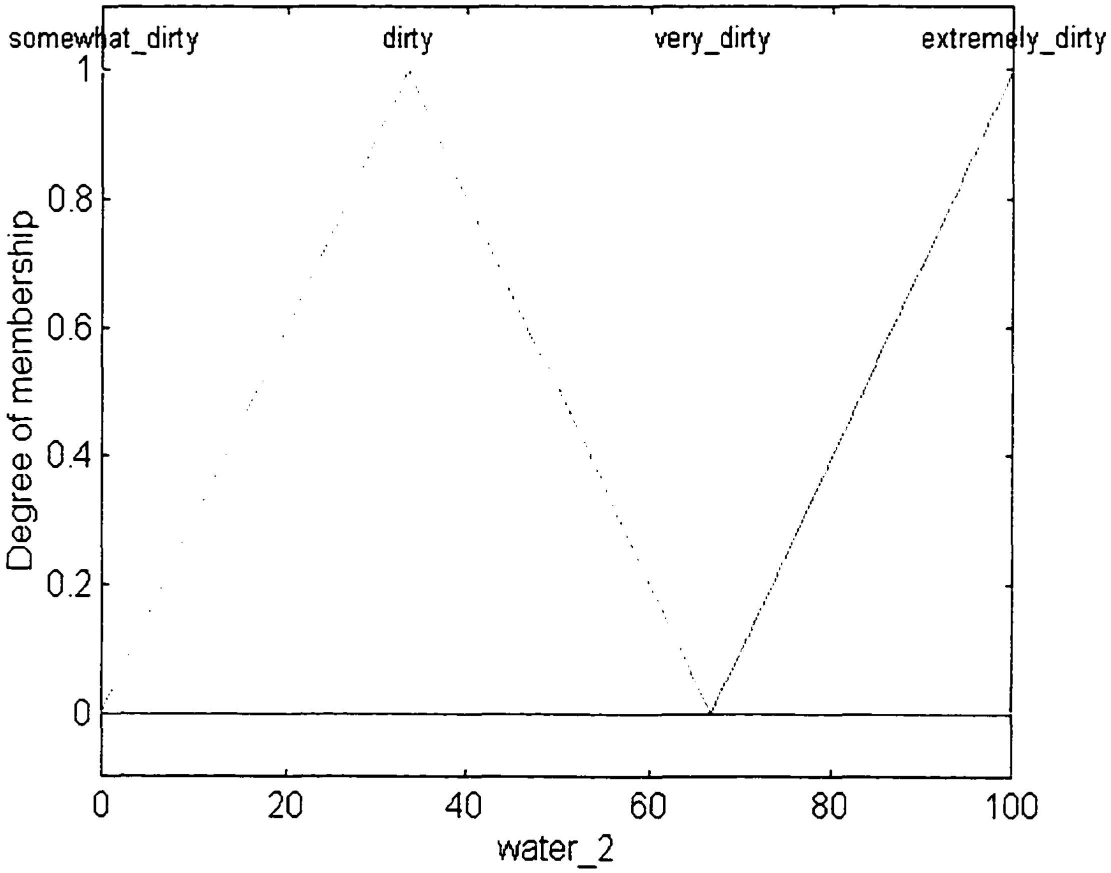


Figure A3 - 10 : MATLAB Output for Deluxe Wash Cycle Input. Water 2

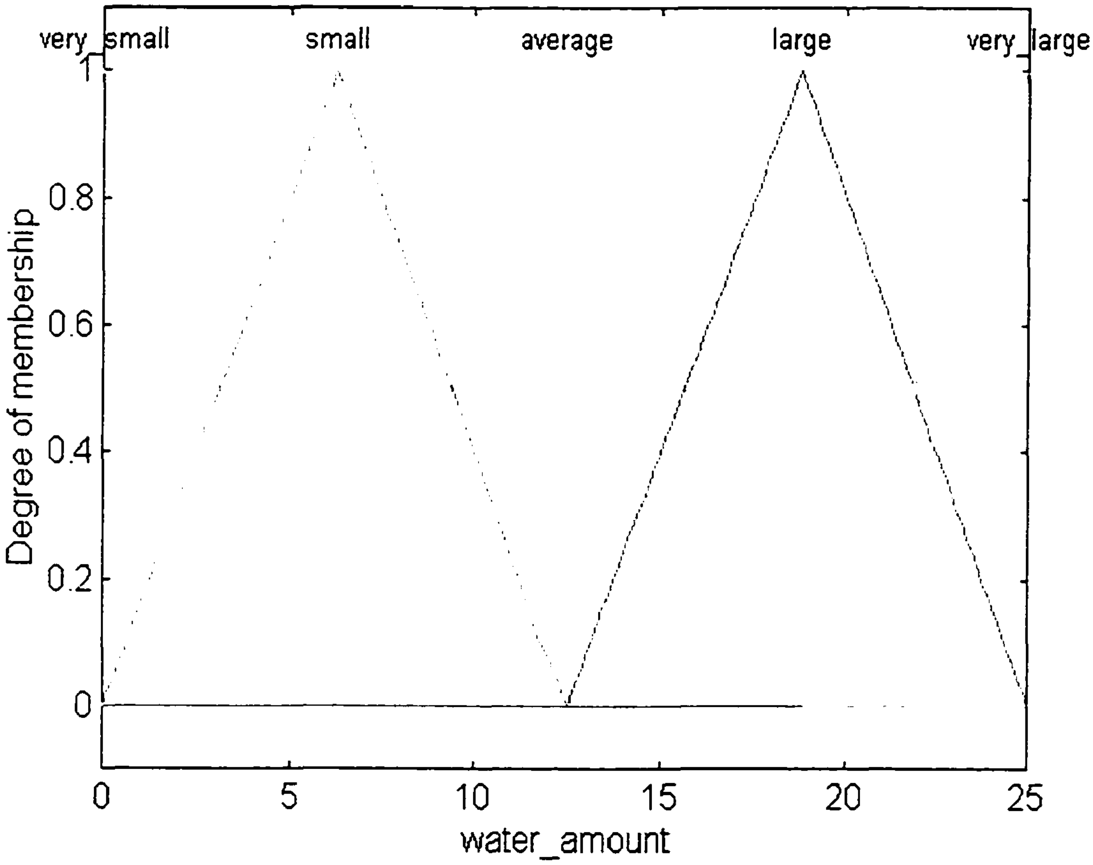


Figure A3 - 11 : MATLAB Output for Deluxe Wash Cycle Output. Water Amount

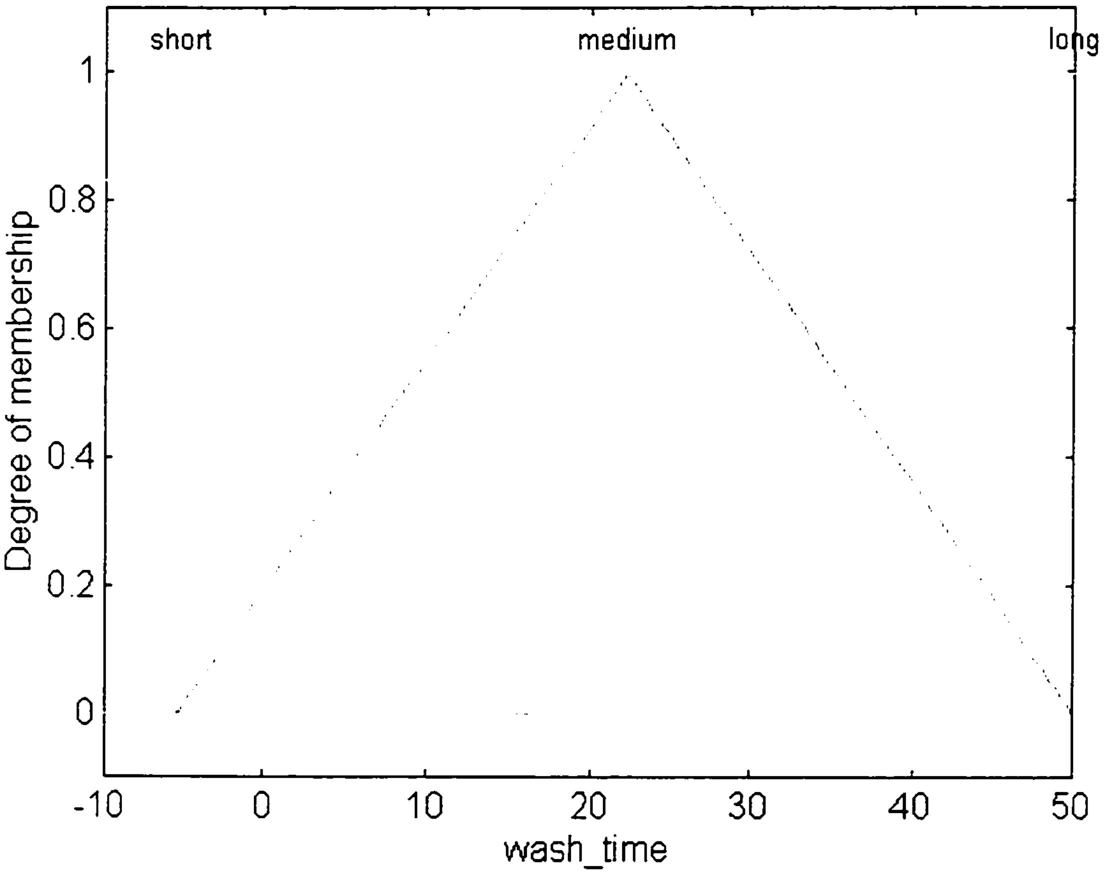


Figure A3 - 12 : MATLAB Output for Deluxe Wash Cycle Output. Wash Time

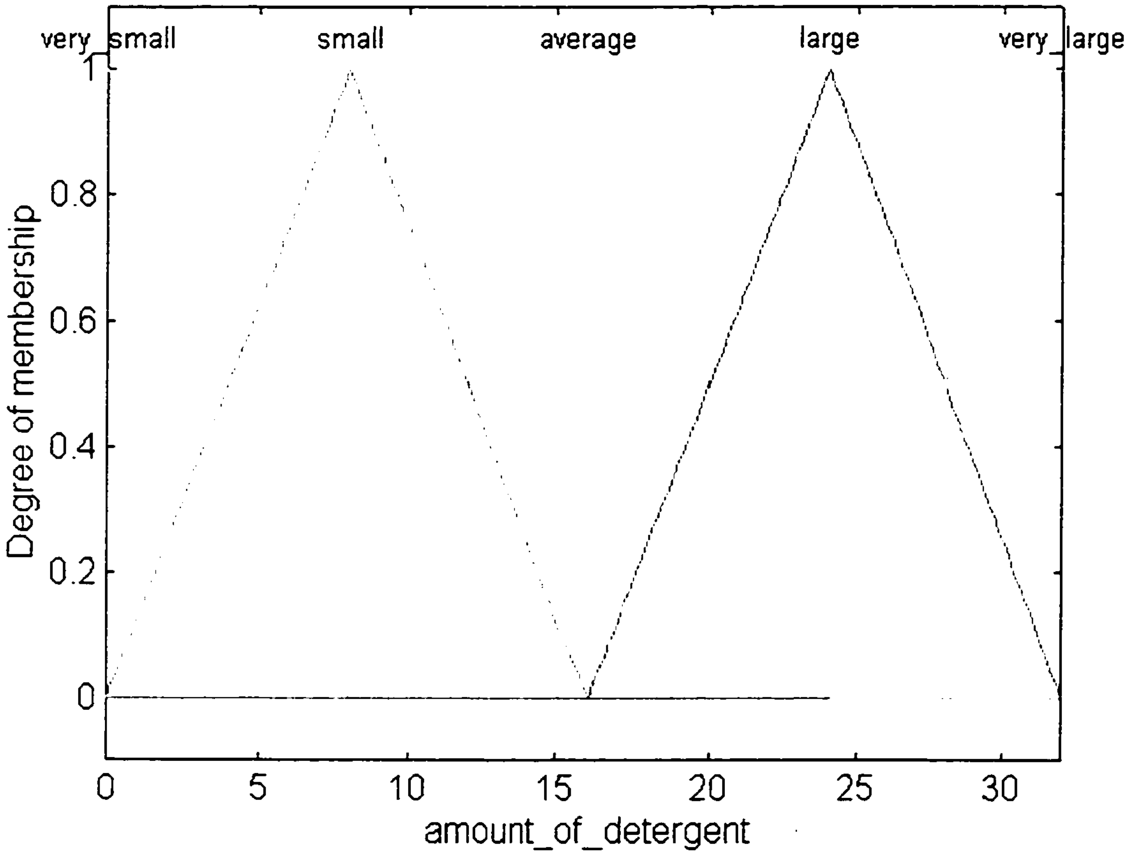


Figure A3 - 13 : MATLAB Output for Deluxe Wash Cycle Output, Water of Detergent

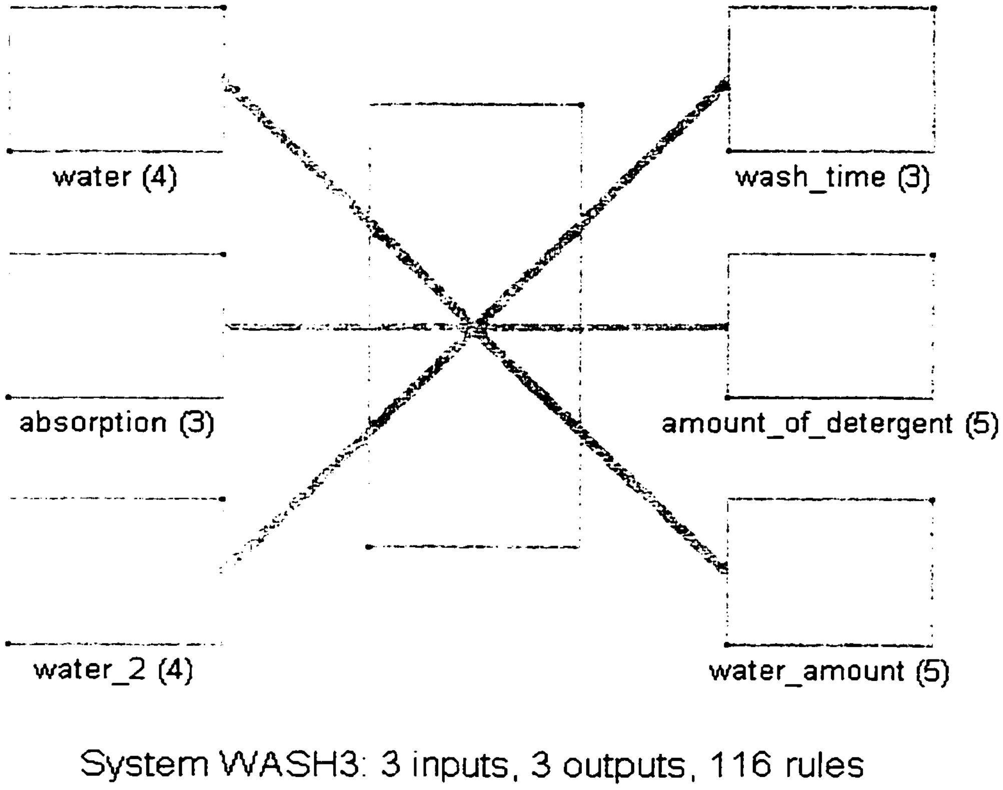


Figure A3 - 14 : System Plot for Deluxe Wash Cycle

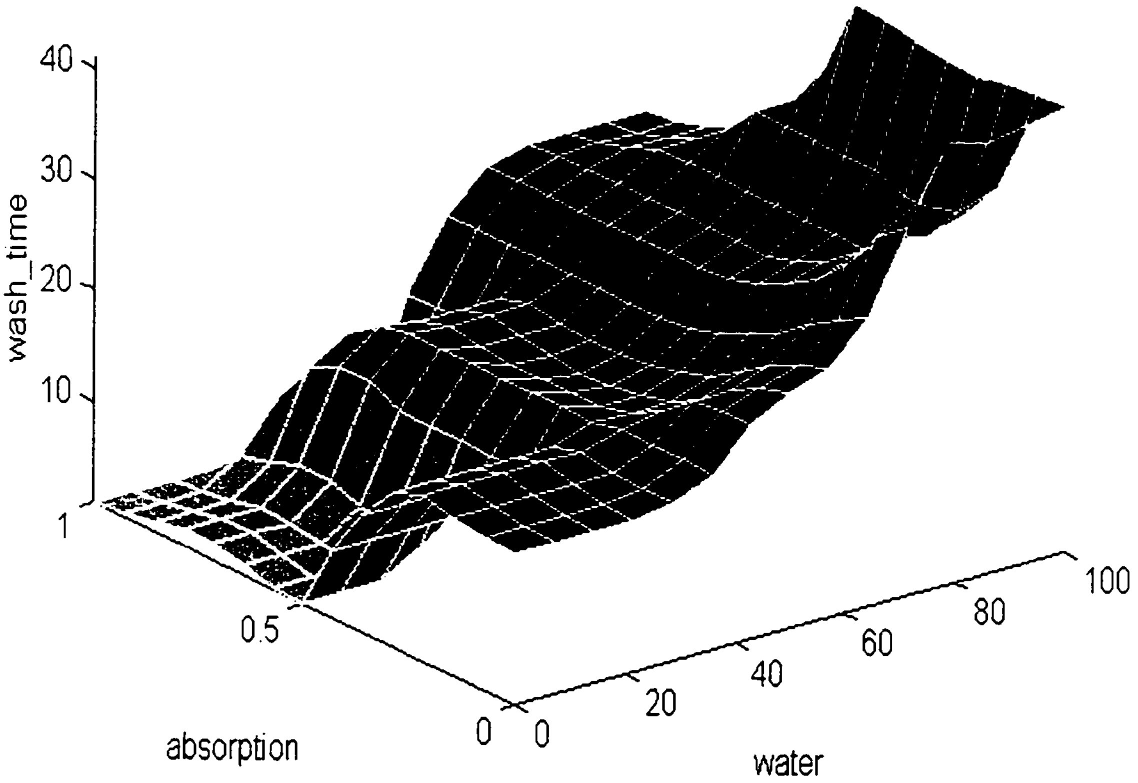


Figure A3 - 15 : Surface View for Deluxe Wash Cycle. Wash Time

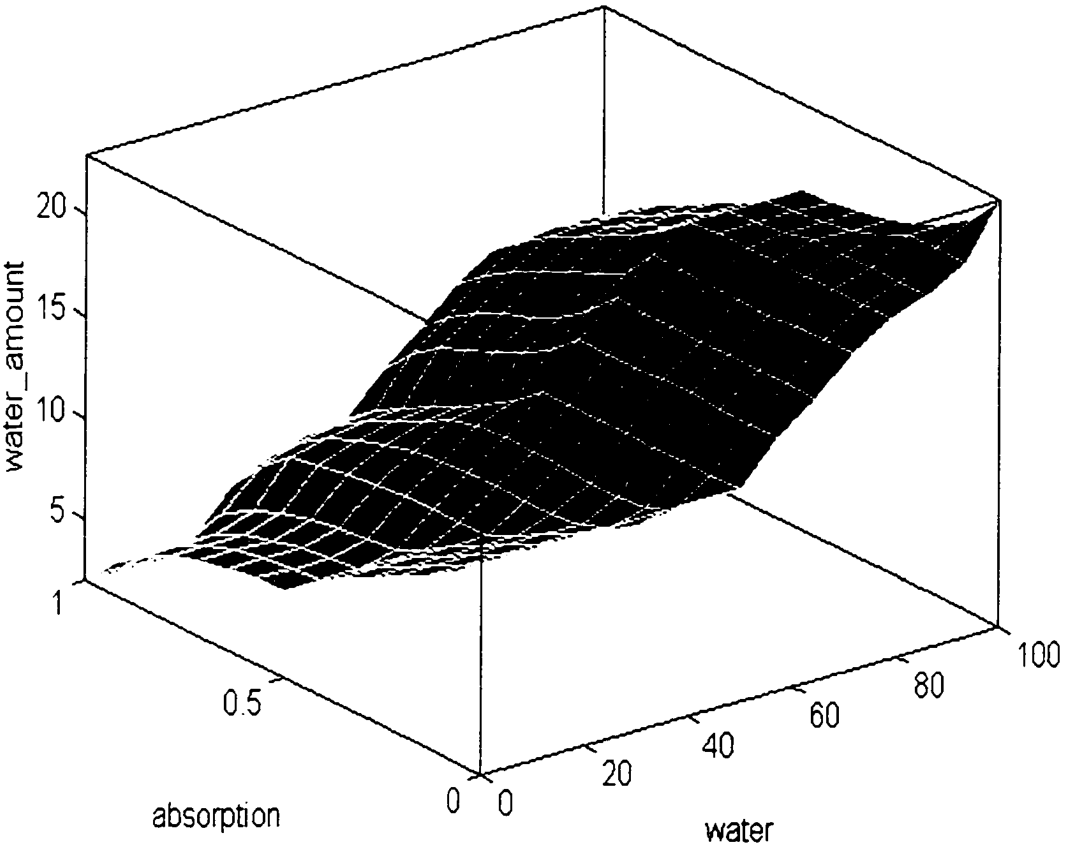


Figure A3 - 16 : Surface View for Deluxe Wash Cycle, Water Amount

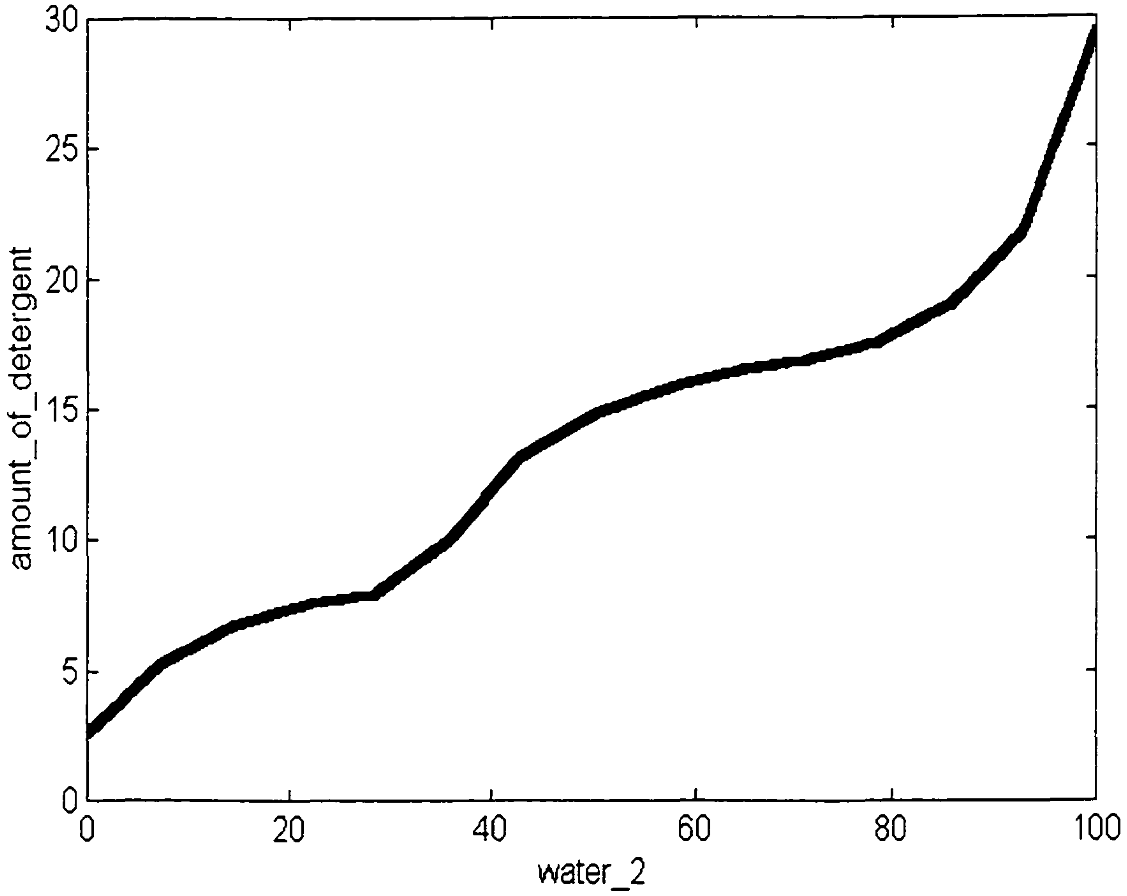


Figure A3 - 17 : Surface View for Deluxe Wash Cycle, Amount of Detergent

REFERENCES

1. McNeill, D. and P. Freiberger, Fuzzy Logic, Simon and Schuster, Inc., 1994.

2. Mir, S. A., M. E. Elbuluk, and D. S. Zinger, "Fuzzy Implementation of Direct Self Control of Induction Machines," *IEEE Trans.Industry Applications*, vol. 30, no. 3, pp. 729-735, 1994.

3. Guillemin, P., "Fuzzy Logic Applied to Motor Control," *IEEE Trans.Industry Applications*, vol. 32, no. 1, pp. 51-56, 1996.

4. Lai, M., M. Nakano, and G. Hsieh, "Application of Fuzzy Logic in the Phase-LockedLoop Speed Control of Induction Motor Drive," *IEEE Trans. Industry Electronics*, vol. 43, no. 6, pp. 630-639, 1996.

5. Sousa, G., B. Bose, J. Cleland, "Fuzzy Logic Based On-Line Efficiency Optimization Control of an Indirect Vector-Controlled Induction Motor," *IEEE Trans. Industry Electronics*, vol. 42, no. 2, pp. 192-198, 1995.

6. Nguyen, H. T. and E. A. Walker, A First Course in Fuzzy Logic, CRC Press, Inc., 1997.

7. Altrock, C., Fuzzy Logic and NeuroFuzzy Applications Explained, Prentice-Hall, Inc., 1995.

8. Phillips, C. L. and H. T. Nagle, Digital Control System Analysis and Design, Prentice-Hall, Inc., 1994.

9. Forbes, M. and B. B. Brey, Digital Electronics, Bobbs-Merrill Company, Inc., 1985.

10. DiStefano, J. J., A. R. Stubbberud, and I. J. Williams, Feedback and Control Systems, McGraw-Hill Book Company, Inc., 1967.

11. Klir, G. J. and B. Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice-Hall, Inc., 1995.

12. Yen, J., R. Langari, and L. A. Zedeh, Industrial Applications of Fuzzy Logic and Intelligent Systems, IEEE Press, 1995.

13. Kruse, R., J. Gebhardt, and F. Klawonn, Foundations of Fuzzy Systems, John Wiley & Sons Ltd., 1994.

14. Altrock, C., Fuzzy Logic and NeuroFuzzy Applications in Business and Finance, Prentice-Hall, Inc., 1997.

15. Klir, G. J., U. H. St. Clair, and B. Yuan, Fuzzy Set Theory : Foundations and Applications, Prentice-Hall, Inc., 1997.

16. Beale, M. and H. Demuth, Fuzzy Systems Toolbox, PWS Publishing Company, 1994.

17. StatSoft, Inc. "Neural Networks" Electronic Statistics Textbook. 1997. <http://www.statsoft.com/textbook/stathome.html> (June 1998).

18. Kantrowitz, M., "Answers to Questions about Fuzzy Logic and Fuzzy Expert". 1997. <http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/fuzzy/part1/faq-doc-0.html> (Feb. 1998).

19. Bauer, P., S. Nouak, and R. Winkler, " A brief course in Fuzzy Logic and Fuzzy Control." 1996. <http://www.flll.uni-linz.ac.at/pdw/fuzzy/fuzzy.html> (Sept. 1997).

20. Attar Software Limited, "Fuzzy Logic." 1996. <http://www.attar.com/pages/fuzzy.htm> (Dec. 1997).

21. Siler, W., "Building Fuzzy Expert Systems." <http://members.aol.com/wsiler/index.htm> (July 1998).

22. Grolier Encyclopedia, "Set Theory." <http://pratt.edu/~arch543p/help/set_theory.html> (May 1998).

23. Grolier Encyclopedia, "Georg Cantor." <http://pratt.edu/~arch543p/help/set_ theory.html> (May 1998).

24. "Notation and Set Theory" *Sets and Relation.* <http://www.shu.edu/html/teaching/math/reals/logic/index.html> (June 1998).

25. Math Works, Inc., "MATLAB." 1998. <http://www.mathworks.com/products/matlab/> (Mar. 1998).

26. NIBS Pte Ltd Technical Report, "NeuroFuzzy Computing." 1996. <http://www.singapore.com / products/nfga/neufuzzy.htm> (June 1998).

27. Aziz, S. A., "You fuzzyin' with me ?" 1996. <http://www-dse.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/sbaa/article1.html#Sub (May 1998).

28. Horstkotte, E., "Fuzzy Logic Overview." 1996. <http://www.austinlinks.com/ Fuzzy/overview.html (April 1998).

29. Brule, J. F., " Fuzzy Systems - A Tutorial." 1985. <http://www.austinlinks.com/ Fuzzy/tutorial.html> (April 1998).

30. Jang, J. and N. Gulley, Fuzzy Logic Toolbox. Math Works Inc., 1995.

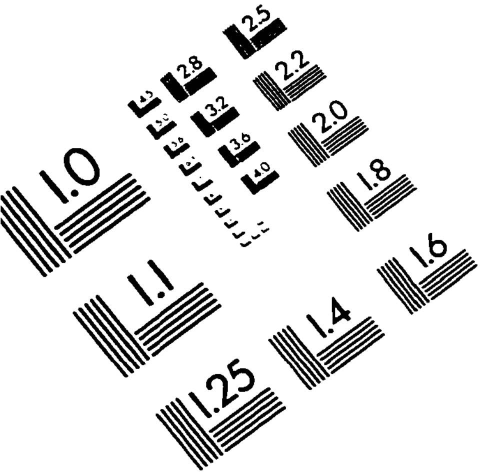
31. The Huntington Group, "Fuzzy Environmental Control." 1992. <http://www. austinlinks.com /Fuzzy/control.html> (April 1998).

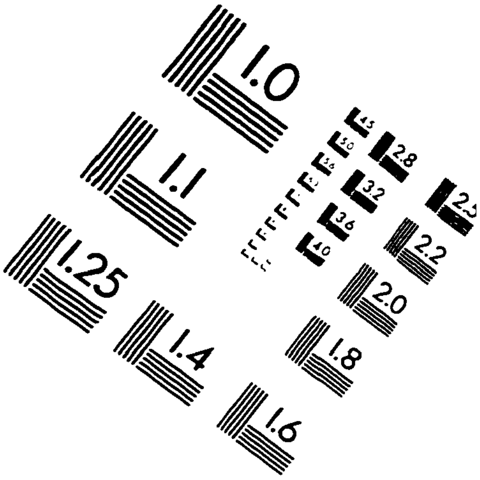
32. Martin, M., Milac, Rand, and Stoll, "Introduction to MATLAB." <http://www. amath. washington.edu/~mmartin/work/MATLAB.html> (April 1998).

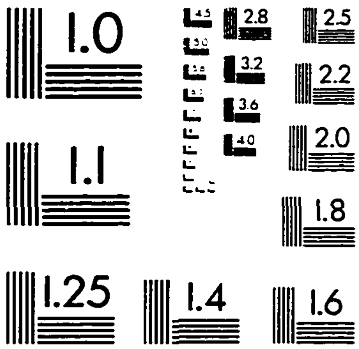
33. Department Of Energy, "Facts About Water." <http://www.epa.gov/OW/facts-quotes/facts.html> (June 1998).

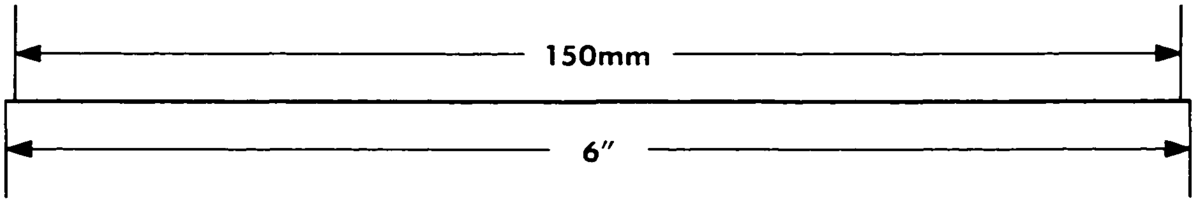
34. Toth, S., "Washing Machines."<http://www.www.which.net/nonsub/washing machines.htm> (May 1998).

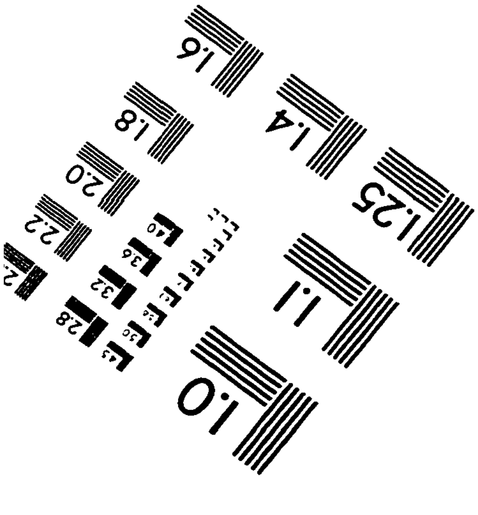
IMAGE EVALUATION
TEST TARGET (QA-3)











APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989