To the Graduate Council:

I am submitting herewith a thesis written by Kevin Joseph Doran entitled "Fuzzy Logic Alternative Membership Function Shape Analysis With Application To Vehicular Steering Control In Extreme Situations." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

JCH

Dr. James C. Hung, Major Professor

We have read this thesis
and recommend its acceptance:

J Wayne Wallen

Herbert S. Neff Jr.

Accepted for the Council:

C W Minkel

Associate Vice Chancellor and
Dean of The Graduate School

FUZZY LOGIC ALTERNATIVE MEMBERSHIP FUNCTION SHAPE ANALYSIS WITH APPLICATION TO VEHICULAR STEERING CONTROL IN EXTREME SITUATIONS

A Thesis

Presented for the

Master of Science Degree

The University of Tennessee, Knoxville

Kevin Joseph Doran

December 1996

DEDICATION

This thesis is dedicated to my loving wife,

Deidre,

who is the cornerstone in my life.

ACKNOWLEDGMENTS

Whenever finishing a long journey, there are so many to recognize who help you along the way. I would like to thank Raytheon Company for providing financial support and to our organization's manager, Allen Hieber, for challenging me to become a more competent engineer. Many thanks to my manager, Gary Brown, for introducing me to fuzzy logic through his thesis work. I would also like to thank Dr. James Hung who not only mentored me as my major professor, but also taught many of my master-level classes. My colleague, David Conboy is to be thanked for the numerous times he helped me with the Matlab software package. Thanks also to Richard Moakler, who provided some much-needed software. Thanks to Margaret Casado for the many times she tirelessly pulled articles for me from the library. I would like to reserve a special thank you to Dr. David Brubaker, who helped form the scope of this work and provided invaluable assistance in its creation. Most importantly, I would like to thank Ms. Lois Presley, who never let me quit.

ABSTRACT

Imprecision has been a major concern to engineers for many years. One way to deal with the type of imprecision where the classes have no sharp transition from membership to non-membership is through fuzzy logic.

While much research and work has been accomplished in the area of fuzzy logic control using triangular-shaped and trapezoidal-shaped membership functions, relatively little effort has been placed on the membership function's shape itself and its impact on overall system response, performance and stability. Thus, the four alternative membership function shapes of trapezoidal, Gaussian, sigmoidal and cubic spline will be compared to the triangular standard by attempting to mimic first-order, second-order, third-order, piecewise-linear and sinusoidal response functions. Zero-order Sugeno inference will be utilized in these rulebased, fuzzy systems.

After identifying the "best" membership function, using defined criteria, its application to a three degree-of-freedom vehicular model will provide data to determine if system performance can be improved through the use of the alternative membership function shape alone.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

It is a persistent and frustrating fact of engineering life that mathematical models and estimations are valid for precise data, yet actual real-world implementation of these models use noisy and imprecise data. This fact has led to the redesign of many systems throughout the years. According to Ribeiro et al. [1], three main forms of imprecision may be identified. They are: (1) incompleteness, which can arise in situations lacking some alternatives, attributes or having insufficient data to stipulate a constraint limit for example; (2) fuzziness, a type of imprecision where the classes (sets) have no sharp transition from membership to non-membership; and (3) illusion of validity, which arises from detecting erroneous outputs such as large deviations from 'expected' deviations or selecting non-relevant alternatives.

One way to successfully deal with the second problem is through the use of fuzzy logic. Although originally defined mathematically in 1965 by Dr. Lotfi A. Zadeh, the concept of fuzzy logic has only recently gained acceptance and appeal in the United States. Contrary to its name, fuzzy logic is soundly based on precise mathematical principles which apply a rigorous approach to the theory of approximation and vagueness based on a generalization of the standard set theory called fuzzy sets [2].

Combining both engineering intuition and heuristics, fuzzy logic allows the use of elements from everyday language (i.e. "hot", "fast", "tall") to represent desired system behavior, thus circumventing the need for meticulous mathematical modeling. Fuzzy logic attempts to model actions to control a system rather than modeling the system process itself. This design philosophy is especially well-suited for applications when:

1. No adequate mathematical model for a given problem exists for conventional design purposes;
2. Non-linearities, time constraints, or multiple parameters exist;
3. Engineering know-how about the given problem is available or can acquired during the design process; and
4. In systems where vagueness is common.

Fuzzy logic concepts have been successfully implemented into products ranging from Mitsubishi air conditioners to Canon auto-focus cameras. This technology has also been tested in Matsushita washing machines and Nissan automobile transmissions and anti-lock braking systems. Although the most visible applications to fuzzy logic may be control systems, where the entire control space is not clearly understood or well defined, fuzzy logic is also used in a wide variety of other applications, including estimation, detection, classification and decision systems.

1.1. Problem Statement

While much research and work has been accomplished in the area of fuzzy logic control using triangular-shaped and trapezoidal-shaped membership functions, relatively little effort has been placed on the membership function's shape itself and its impact on overall system response, performance and stability. Originally, bell-shaped membership function shapes were used in fuzzy design [3]. However, the complexity of calculating the bell shape was time-prohibitive at that time. Therefore, the bell-shaped curve was replaced by a triangle or trapezoid, as it was easier to compute and provided satisfactory results. The triangular membership function, specifically, has become generally accepted as the standard by the fuzzy control design community. However, it is now time to challenge this train of thought. The tremendous increase in digital processing power during the last several years lends itself well for computing more complex membership functions without the negative aspect of delayed response time.

Diligent research has uncovered relatively little information in the area of alternative membership function shape. Albeit somewhat outdated now, an article written in 1985 by Maiers and Sherif [4] performed a state-of-the-art literature review related to applications of fuzzy set theory available at that time. Not one of the 56 references nor 450 of the bibliographic items mentioned anything on this subject. A data search from 1985 to present was also performed on the On-line Computer Library Center (OCLC), which is one of the most comprehensive national databases available. It only provided a handful of membership function-related references. According to Foslien et al., "Little is known about how the effects of the shapes of membership functions or the amount of division on the universe of discourse have upon the performance of the fuzzy controller ..." [5]. In addition, Boston states: "... but there have been few systematic studies of the effects of the membership function parameters on system performance" [6].

It is possible that an existing fuzzy system's performance can be enhanced by alternative membership function shape. This thesis will firstly explore four alternative input membership function shapes and compare them with the triangular standard. Secondly, the "best" shape will be applied to a fuzzy-controlled vehicular steering model created by Brown [7]. This will provide measurable results in which to assess the potential additional benefit through using alternative membership function shapes.

1.2. Thesis Organization

The work presented here attempts to explore the use of alternative input membership functions. This chapter provides an overview on the forms of imprecision as well as a scientific technique used to address this problem.

Furthermore, the chapter delineates the reasons for using fuzzy logic technology.

Chapter 2 uses a double-input single-output control system to explain fuzzy control theory and its associated properties. Fuzzification, inferencing and defuzzification are all addressed using the classical Mamdani approach. In addition to the overview, definitions applied to membership functions are also presented.

Definition of the triangular standard and four alternative input membership function shapes follow in Chapter 3. The shape, degree of overlap and placement within each universe of discourse are defined. The number of membership functions within these universes of discourse will not exceed seven.

Each input membership function will be utilized within a complete fuzzy module for evaluation purposes. Chapter 4 delimits the ramp input stimulus to be used and the five response functions to be mimicked by the fuzzy systems. Sugeno inference will be used with results accordingly displayed. By using Sugeno inference, only input membership function shapes are required. Output membership functions will be singletons. Reviewing these results, using statistical methods, will provide insight to which membership function shape yields the "best" results according to stipulated guidelines.

Application of the best membership function shape to a nonlinear front and rear independent steering vehicle model follows in Chapter 5. The fuzzy controller will be exposed to varying amounts of wheel slip and must provide front and rear steering compensation in order to minimize yaw error, which is defined as the angle from centerline of travel a vehicle spins. A car with minimum yaw rate error thus minimizes its spin on an icy surface.

This work ends with a conclusion in Chapter 6. The problem statement will be reiterated and pertinent facts and a final assessment of the work performed will be compiled into summary form. Other possible thesis topics are also stated.

2. FUZZY CONTROL THEORY

It is common engineering practice to transform time-domain data into another domain for ease of data computation. Such is the case for using the Laplace transform when analyzing electrical circuits. The time domain is transformed into the *s* domain so the effects of resistance, capacitance and inductance on a circuit may be straightforwardly computed. In fuzzy logic, data from the time domain are transformed into the fuzzy domain to be manipulated. After manipulation, the data are transformed back into time-domain data for system control. Performing discrete-time signal processing using the Fourier transform is an analogous mathematical method to using fuzzy logic. According to Brubaker [8], "Because the necessary processing is easier in the frequency domain than it is in the time domain, processing time-domain data warrants the expense of an FFT (editor's note: fast Fourier transform) and an inverse FFT." The same is true in many cases when applying fuzzy logic. The ability to describe a system with words instead of complex math is well worth the fuzzification and defuzzification processes.

A distinction between probability theory and fuzzy set theory must be noted. In general, probability theory deals with randomness of future events and fuzzy set theory deals with imprecision of current or past events. Maiers and Sherif [4] state, "Randomness deals with the uncertainty regarding the occurrence or nonoccurrence of some event, while the imprecision of fuzzy sets deals with the membership or non-membership of an object in a set with imprecise boundaries." Both of these techniques do address the issue of uncertainty. However, fuzzy set theory addresses uncertainty resulting from imprecise boundaries while probability theory addresses uncertainty resulting from random behavior which tends to dissipate with increasing information.

2.1. Fuzzy Logic Introduction

In general, a fuzzy logic controller (FLC) consists of a fuzzification interface, an inference engine and a defuzzification interface. More specifically for this work, only fuzzy rule-based controllers will be used, although other structures do exist. This is depicted in Figure 2-1.

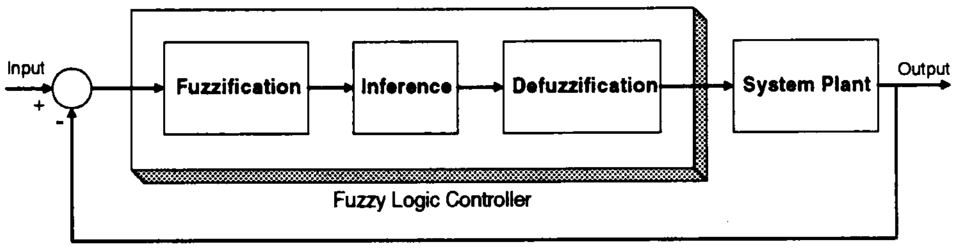


Figure 2-1. Fuzzy feedback controller

It should be noted that the first and third fuzzy logic controller blocks are technically referred to as "crisp to fuzzy" and "fuzzy to crisp" interfaces respectively. As fuzzification is one way to perform a crisp to fuzzy interface, defuzzification is one way to accomplish a fuzzy to crisp interface (another technique is called a “lambda-cut”). However, the “fuzzification” and “defuzzification” terms are generally accepted within the engineering community and, for purposes of this document, will be synonymous with the “crisp to fuzzy” and “fuzzy to crisp” conversion definitions.

2.1.1. Fuzzification

A crisp input is one which has a distinct value associated with a label, thus giving it meaning (i.e. 58 miles per hour, 60 degrees, etc.). In traditional bi-level logic, an element is completely a member of a pre-defined set or not a member at all. Using $z = 58$ miles per hour (mph) as an example, set A may be defined from 30 mph to 50 mph. Set B may be defined from 50 mph to 70 mph. This is shown below.

$$A=[30,50] \tag{2-1}$$

$$B=[50,70] \tag{2-2}$$

Therefore, the crisp value $z = 58$ mph is not a member of set A at all ($z \notin A$) and completely a member of set B ($z \in B$). If μ is defined as the degree of membership between variable z and set B, then $\mu_B(z)=[0,1]$ which can take the value of zero (0) or one (1), but nothing between. The degree of membership between variable z and set B is depicted in Figure 2-2.

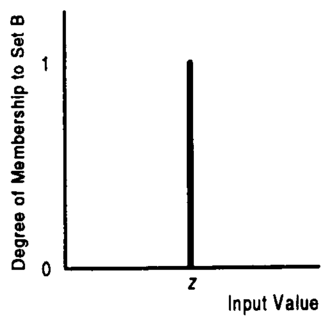


Figure 2-2. Crisp value degree of membership

A crisp value may have associated error bounds also. This can take the form of $z \pm y$. Therefore, a range of numbers would be considered members of a crisp set. However, as with any bi-level logic, null or complete membership is required, as shown in Figure 2-3.

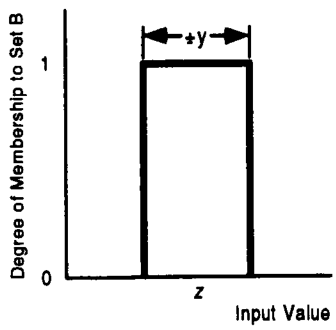


Figure 2-3. Crisp set degree of membership

A fuzzy set may be viewed as a generalization of a Boolean set whose membership function may take a value in the interval zero (0) and one (1)

$$\mu_B(z) \rightarrow [0,1] \qquad (2\text{-}3)$$

where zero (0) indicates null membership and one (1) indicates full membership. Partial membership may exist along with membership in one or more fuzzy sets. At this point, it is helpful to define the crisp system variable *z* linguistically as *speed*. In addition, each fuzzy set is defined by a label. This provides another linguistic aspect to the model. Labeling set B as "fast", Figure 2-4 depicts a fuzzy set.

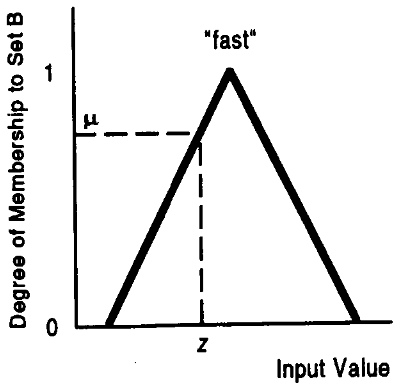


Figure 2-4. Fuzzy degree of membership

Therefore, μ_{fast}(*speed=58*)=0.8, means that *speed*=58 mph is 0.8 fast. Stated another way, "The speed 58 mph is fast" is true with a truth value of 0.8. The domain of a system variable (such as *speed)* denotes the extent of valid values. This extent, that is all possible values applicable to a system variable, is known as its "universe of discourse," U. Multiple membership functions are used to completely specify a system variable throughout its universe of discourse. This is shown in Figure 2-5 which delineates *speed's* U as 0 mph to 80 mph.

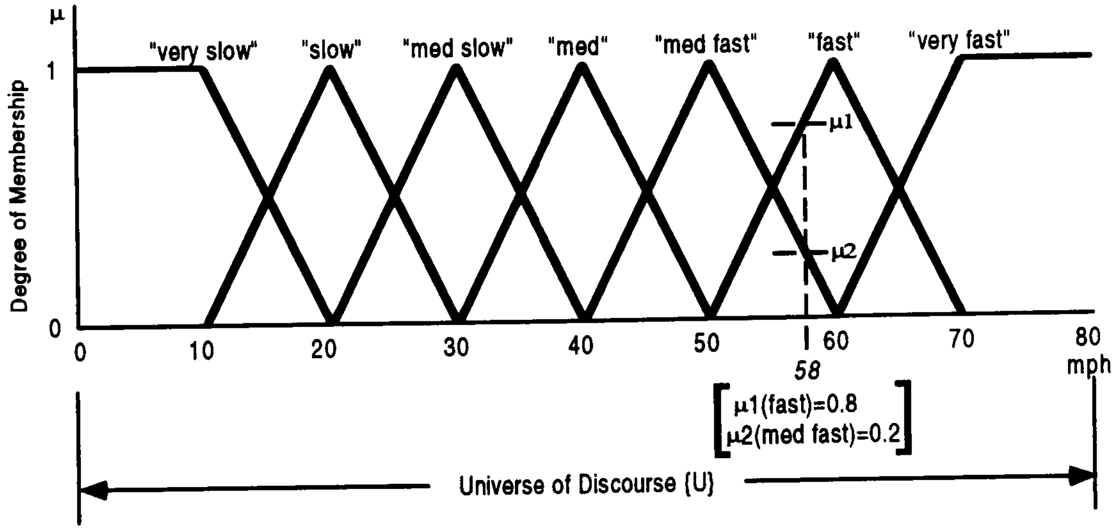


Figure 2-5. Universe of discourse for system variable "speed"

Thus, for *speed*=58 mph, the following membership of four selected fuzzy sets would be:

$\mu_{med}(speed=58mph)=0.0$;

$\mu_{med\ fast}(speed=58mph)=0.2$;

$\mu_{fast}(speed=58mph)=0.8$; and

$\mu_{very\ fast}(speed=58mph)=0.0$.

The number of fuzzy sets used to define a system variable may vary. Most designs currently use five to nine. Fuzzy sets within a universe of discourse do not have to be the same shape or be defined by the same width. In addition, the placement of these sets need not be equidistant. Note that a set's maximum degree of membership need not be unity, although this is the case in the above example.

2.1.2. Inference

Consider fuzzy sets A and B respectively on a given universe of discourse U. As with crisp sets, fuzzy sets are manipulated by the operators of union, intersection and complement. Other fuzzy operator definitions do exist. However, these are the three originally proposed by Zadeh [2].

The union of fuzzy sets A and B, represented by $A \cup B$, seeks to maximize the truth value between these two sets. Given an element z taken over U, the union of the two sets A and B will result in the larger value. This is analogous to an OR function.

$$\mu_{A \cup B}(z) = \max[\ \mu_A(z)\ ,\ \mu_B(z)\ ] \qquad (2\text{-}4)$$

The intersection of fuzzy sets A and B, represented by $A \cap B$, seeks to minimize the truth value between these two sets. Given an element z taken over U, the intersection of the two sets A and B will result in the smallest membership value. This is analogous to an AND function.

$$\mu_{A \cap B}(z) = \min[\ \mu_A(z)\ ,\ \mu_B(z)\ ] \qquad (2\text{-}5)$$

The complement of fuzzy set A, represented by $\bar{A}$, seeks to determine the inverse truth value of that set. Given an element z taken over U, the complement of set A is simply the value one (1) minus the membership value. This is analogous to a NOT function.

$$\mu_{\bar{A}}(z)=1-\mu_A(z) \tag{2-6}$$

In addition, with this set of operators, the classical properties of associativity, distributivity, transitivity, etc. also apply to fuzzy sets. In fact, the only basic axioms not common to both crisp and fuzzy set theory are the excluded middle law: $A \cup \bar{A} = U$, and the law of contradiction, $A \cap \bar{A} = 0$.

These above-mentioned properties support an earlier statement that fuzzy logic is a generalization of Boolean logic. Stated another way, Boolean logic is actually a subset of fuzzy logic. By inserting a zero or a one into the logical operators AND, OR, and NOT, one may obtain the same results as with crisp logic. This is known as the Extension Principle, which states that the classical results of Boolean logic are recovered from fuzzy logic operations when all fuzzy membership grades are restricted to the traditional set [0,1] [9]. This effectively establishes fuzzy subsets and logic as a true generalization of classical set theory and logic. By this reasoning, all crisp subsets are fuzzy subsets of this special type. Therefore, there is no conflict between these two methods.

Rule-based fuzzy systems consist of a group of IF-THEN statements.

Rule 1: IF z(t) is Z_1 AND d(t) is D_1, THEN t(t) is T_1

Rule 2: IF z(t) is Z_2 AND d(t) is D_2, THEN t(t) is T_2

.

.

.

Rule N: IF z(t) is Z_N AND d(t) is D_N, THEN t(t) is T_N

It should be noted that the AND function in the antecedent (input) block of this statement is completely optional. It is perfectly within fuzzy design guidelines to use a rule such as: IF z(t) is Z_1, THEN t(t) is T_1. However, design experience has shown that this AND function provides a key way to inter-relate input variables. Likewise, more than one AND statement may be used in any rule. Building on the example previously described in this chapter, the rule base may be applied to a fuzzy controller used for an intelligent cruise control (ICC) device in an automobile. This device's task is to acquire and "lock onto" the preceding vehicle traveling in the same lane. Under the fuzzy controller's guidance, the car then maintains a safe distance behind the preceding vehicle depending on the speed of travel. The driver is thereby not required to continually turn the car's cruise control on and off when traffic flow is altered. From previously, z(t) still denotes the speed of a vehicle. Functions d(t) would represent the relative distance between the two vehicles and t(t) would define the amount of throttle required to remain a safe distance behind the preceding car. Remaining consistent with the first fuzzy set, both d(t) and t(t) will be defined by seven fuzzy subsets over their respective universes of discourse. As delimited by the variable "t", these functions represent crisp, time-domain labels. An example rule would be:

IF *speed* is slow AND *delta_dist* is pos. med THEN *throttle* is pos. large.

In this example, positive *delta_dist* refers to an increasing distance between the cars and negative *delta_dist* is a decreasing distance. Likewise, a positive *throttle* results in acceleration, while negative *throttle* results in deceleration. A *throttle* of "zero" does not denote no throttle at all. It simply signifies that there is no change in throttle position. An example of this rule system is shown in Figure 2-6.

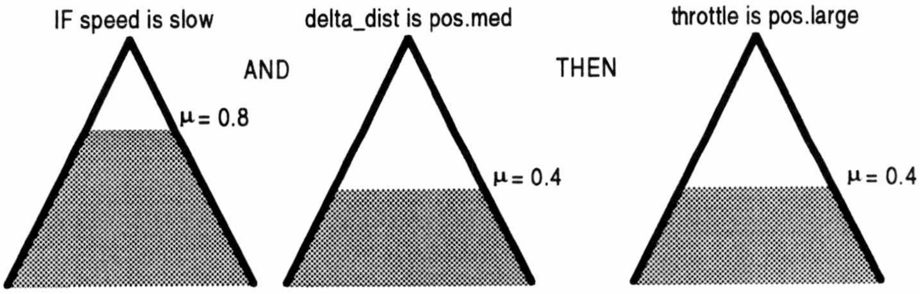


Figure 2-6. Fuzzy Rule Inferencing

The AND function determines the minimum membership grade value from the antecedent block. The THEN function applies this minimum value to the consequent (output) block. This process is performed on the entire set of rules used in the fuzzy logic controller.

Note that each membership function is dependent only on one criterion in determining its membership grade. For example, *speed* is only dependent on how fast (in mph) a vehicle travels. This is known as a one-dimensional membership function. It is possible that a membership function may be defined by two (or more) input variables. A speed of 80 mph may be extremely fast for a Hyundai Excel but not very fast for a Porsche 944. Therefore, a membership function may determine that a speed is fast or slow depending on the vehicle model. This is known as a two-dimensional (or "multi-dimensional") membership function, but will not be addressed in this work.

Of course, a fuzzy controller utilizes more than one rule to control the target system. Multiple rules are combined and used in the manner above to fully define system control. Each antecedent block is defined by seven fuzzy subsets. Accounting for all possible input combinations would require 49 (7x7) rules. This is not necessary in the vast majority of fuzzy applications [10]. For example, the ICC device above will be controlled using only seven rules. The *delta_dist* universe of discourse will be defined as 0 ± 40 feet per second (ft/s). The *throttle* universe of discourse is a bit more ambiguous from full negative throttle (maximum braking) to full positive throttle (maximum acceleration). This would be dependent on the actual vehicle used. Note that the second antecedent block and consequent block labels are shortened for convenience. Therefore "pos. med" will be shortened to "pm" and "negative large" will be shortened to "nl" etc. The rules are:

Rule 1: IF *speed* is slow AND *delta_dist* is pm THEN *throttle* is pl

Rule 2: IF *speed* is med. slow AND *delta_dist* is nm THEN *throttle* is nm

Rule 3: IF *speed* is med AND *delta_dist* is nm THEN *throttle* is ns

Rule 4: IF *speed* is med AND *delta_dist* is pm THEN *throttle* is ps

Rule 5: IF *speed* is med. fast AND *delta_dist* is ns THEN *throttle* is zr

Rule 6: IF *speed* is fast AND *delta_dist* is ps THEN *throttle* is pm

Rule 7: IF *speed* is fast AND *delta_dist* is nl THEN *throttle* is nl

Using system inputs of *speed* = 48 mph and *delta_dist* = -15 feet per second (ft/s), Figure 2-7 details the inference process This classical definition is known as the Mamdani inference method [11]. Rules 3 and 5 are said to "fire". That is, the system inputs result in a non-zero membership grade in the corresponding consequent blocks. For each rule, the minimum antecedent membership grade propagates through to the consequent block. The truncated membership functions for each rule are aggregated, thus

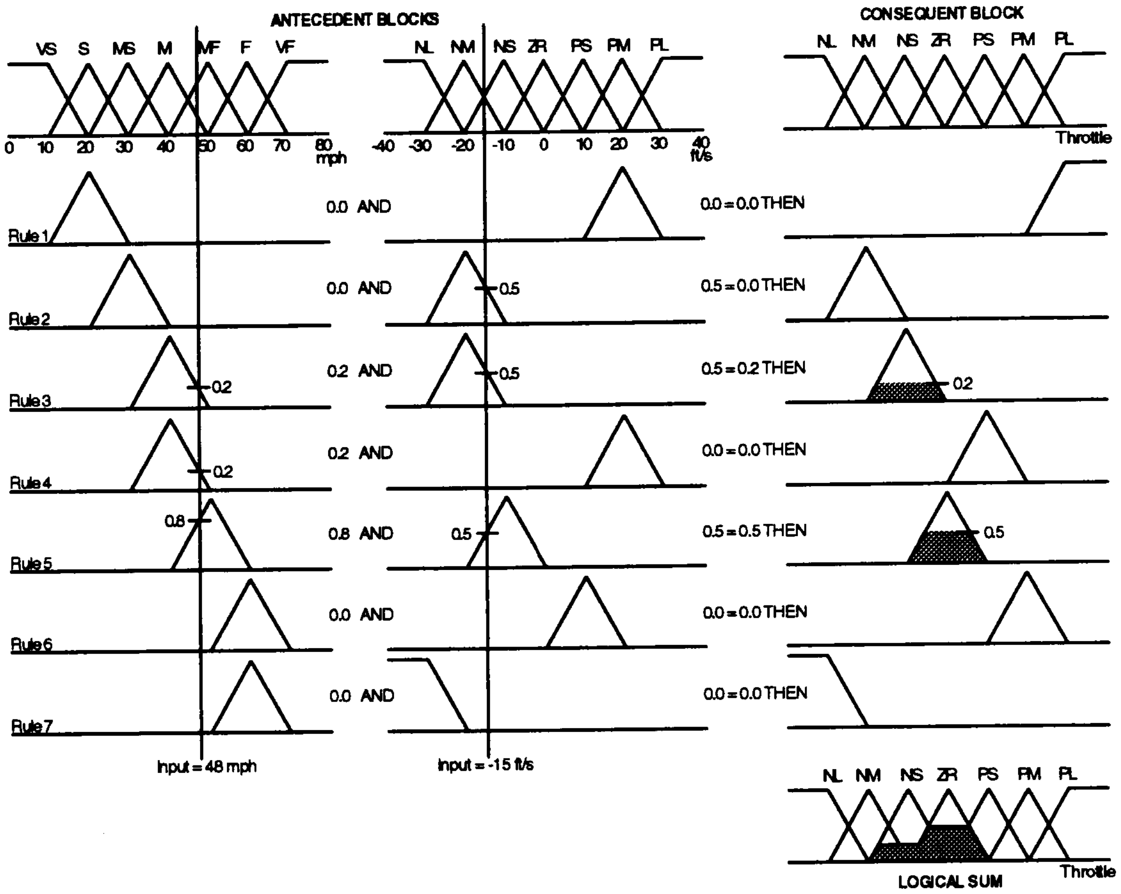


Figure 2-7. Mamdani Fuzzy Inference Process

creating an outer envelope of the individual truncated membership forms from each rule using Mamdani's (max-min) technique below.

$$\mu_{T(N)}(t)=\max[\min[\ \mu_{Z(N)}(z)\ ,\ \mu_{D(N)}(d)\ ]] \qquad N=1\text{-}7 \qquad (2\text{-}7)$$

Another technique is known as the max-product inference method. The effect of the max-product implication is that the consequent membership functions remain as scaled triangles, as opposed to the truncated triangles with the Mamdani method. This is presented for informational purposes only and will not be used in this work. In application, there is little system performance difference between the two in most control systems.

2.1.3. Defuzzification

The currently available defuzzification methods are too numerous to mention in this text. However, four of the more popular methods are described below [12]. Note that t^* denotes the actual output value.

The max-membership method, also known as the height method, uses the single largest membership grade as the output point. As the membership grade must be singular, this method is limited to peaked output functions. This method is given by the algebraic expression

$$\mu_{T(N)}(t^*) \geq \mu_{T(N)}(t) \qquad \text{for all } t \in T \qquad (2\text{-}8)$$

and is shown graphically in Figure 2-8. The example ICC output needs to be modified slightly to demonstrate this method. Therefore, the consequent membership grade for Rule 5 is set to one.

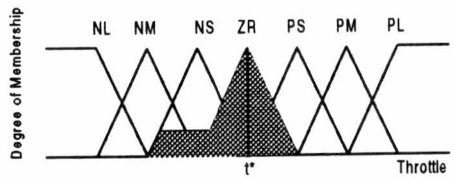


Figure 2-8. Max-membership defuzzification method

The centroid, or center of gravity, method is currently the most prevalent defuzzification method in use. It is given by the algebraic expression

$$t^* = \frac{\int \mu_{T(N)} \bullet t dt}{\int \mu_{T(N)} dt} \tag{2-9}$$

where $\int$ denotes an integration procedure. This method is illustrated in Figure 2-9.

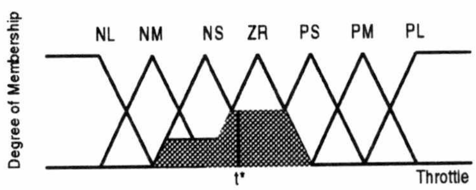


Figure 2-9. Centroid Defuzzification Method

Only valid for symmetrical membership functions, the weighted average method is given by the expression

$$t^* = \frac{\sum \mu_{T(N)}(\bar{t}) \bullet \bar{t}}{\sum \mu_{T(N)}(\bar{t})} \tag{2-10}$$

where Σ denotes an algebraic sum. This method is shown in Figure 2-10.

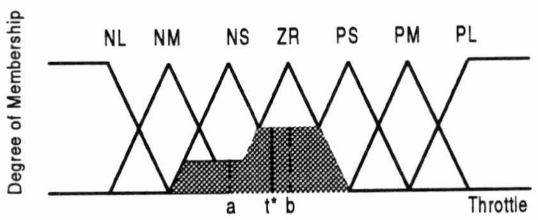


Figure 2-10. Weighted Average Defuzzification Method

The weighted average method is formed by weighting each membership function in the output by its respective membership grade. Therefore, the two functions shown in Figure 2-10 would result in the following defuzzified value:

$$t^* = \frac{a(0.2) + b(0.5)}{0.2 + 0.5} \tag{2-11}$$

Note that since this method is limited to symmetrical membership functions, the values of *a* and *b* are the means of their respective shapes.

The mean-max approach is closely related to the first defuzzification method defined in this section, except that the locations of the maximum membership grade may be non-unique. That is, the maximum membership grade may be a plateau rather than a single point. This method is shown in Figure 2-11.

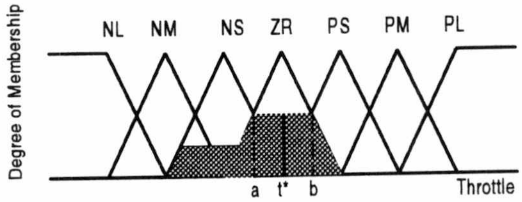


Figure 2-11. Mean-Max Defuzzification Method

This method is given by the equation

$$t^* = \frac{a+b}{2} \tag{2-12}$$

and is simply the midpoint of the maximum membership grade level.

2.2. Detailed Membership Function Description

Most things in nature can not be characterized by simple shapes or functions. And many real-world processes are non-linear at some point. Membership functions can be uniquely defined to mimic real-world data as closely as possible. That is, membership functions characterize the fuzziness in a fuzzy set. Due to this importance, membership function shape can be critical to the success of a fuzzy logic controller.

There is virtually an infinite number of membership function shapes that can be created. However, it is useful from a mathematical standpoint to define certain attributes a membership function may have for description purposes. Table 2-1 details the definitions illustrated in Figure 2-12.

Table 2-1. Membership Function Definitions

Attribute	Definition
Core	Region of the fuzzy set that is characterized by complete and full membership.
Support	Region of the fuzzy set that is characterized by non-zero membership.
Boundaries	Region of the fuzzy set that is characterized by non-zero membership, but not complete membership.
Overlap	Region of one side of the fuzzy set that overlays the adjacent fuzzy set.
Normal	A fuzzy set with at least one element whose membership grade is unity.
Convex	A fuzzy set which is strictly monotonically (SM) increasing, SM decreasing, or SM increasing then SM decreasing with increasing values for elements in the fuzzy set.

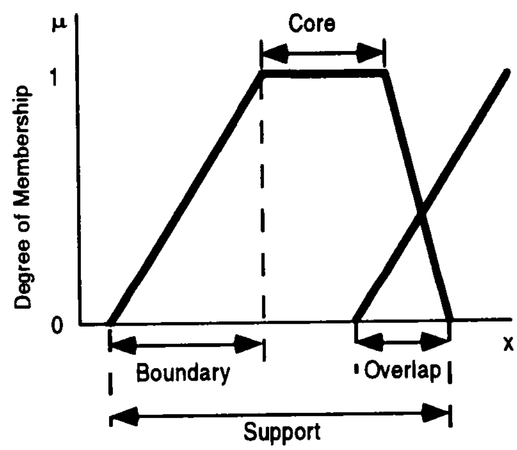


Figure 2-12. Graphical Membership Function Definitions

Membership functions may be created in one of several ways. Two of the more popular methods are described below.

Intuition is often used to create membership function shapes. This is because a person can successfully control a process without knowing the intimate details of the process itself. Once assigning membership function shapes to each variable, these shapes may be modified or moved within its universe of discourse to maximize system performance.

Inference is a method by which a conclusion is deduced or inferred, given a body of facts or knowledge. These data may be obtained through experimental procedures, documentation and even opinion polls and surveys.

Other means of generating membership functions include genetic algorithms, statistics and neural networks. These will not be addressed in this text. Rather, this effort will focus on normal, convex membership functions which will be generated through intuition. In addition, each membership function must have one normal point within its respective universe of discourse.

3. MEMBERSHIP FUNCTION SHAPES

This chapter describes the standard triangular shape and four alternative input membership function shapes which will be analyzed in this thesis effort. As stated in Section 2.2, only normal, convex membership functions will be considered. Thus, the degree of membership will map between zero (0) and one (1). In addition, each membership function shape will be restricted to being dependent on only two variables. This prevents a given membership function type from having an unfair advantage. Any generalized membership function shape which requires more than two variables will require these additional variables to be dependent on one of the first two.

3.1. Triangular Function

A generalized triangular membership function shape is defined by the equation:

$$f(x;a,b,c) = \begin{Bmatrix} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{Bmatrix} \tag{3-1}$$

In order to restrict this shape to two variables, lengths *ab* and *bc* will be equal, thus forming an isosceles triangle. Therefore, triangular parameter *b* will be in the center of support *ac*, and length *ab* will denote the width from null membership to full membership as depicted in Figure 3-1.

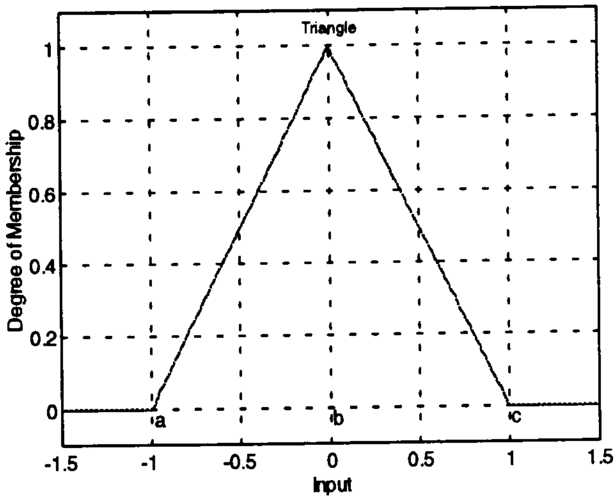


Figure 3-1. Triangular Membership Function Shape

3.2. Trapezoidal Function

A generalized trapezoidal membership function shape is defined by the equation:

$$f(x;a,b,c,d)=\begin{Bmatrix} 0, & x\le a \\ \dfrac{x-a}{b-a}, & a\le x\le b \\ 1, & b\le x\le c \\ \dfrac{d-x}{d-c}, & c\le x\le d \\ 0, & d\le x \end{Bmatrix} \qquad (3\text{-}2)$$

In order to define this function's shape with only two variables, certain restrictions must apply. Firstly, lengths *ab* and *cd* must be equal, creating a symmetric shape. Secondly, length *bc* must be one-fourth the length of *ad*. Thus, the core-to-support ratio is 1:4. As defined by the author, this allows for a trapezoidal shape without the region of complete membership being relatively too large or too small. Figure 3-2 depicts this membership function.

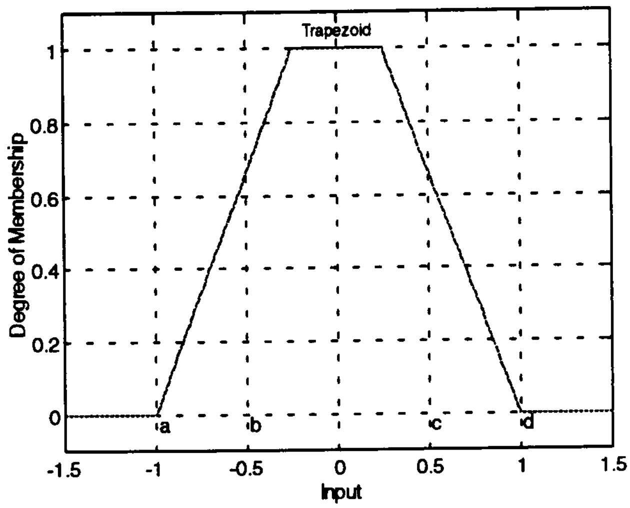


Figure 3-2. Trapezoidal Membership Function Shape

3.3. Gaussian Function

A generalized Gaussian membership function shape is defined by the equation:

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{3-3}$$

The variable c represents the curve's center point while σ can be statistically defined as a set's standard deviation. As this function is only governed by two variables, no additional restrictions need apply. A representative curve is shown in Figure 3-3.

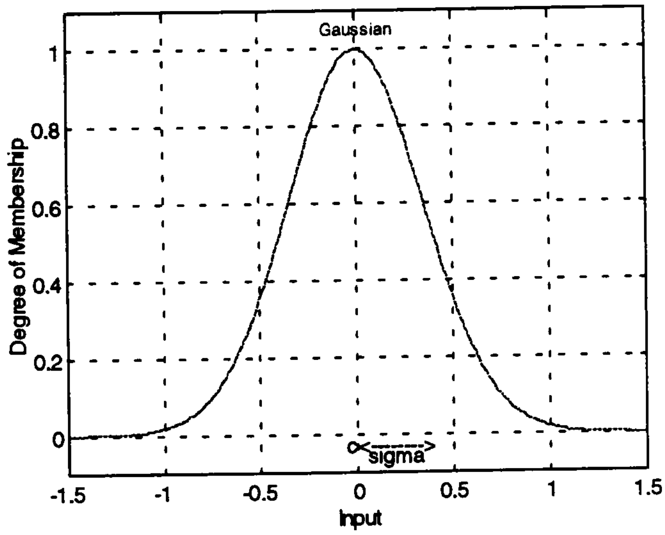


Figure 3-3. Gaussian Membership Function Shape

3.4. <u>Sigmoidal Function</u>

A generalized sigmoidal membership function shape is defined by the equation:

$$f(x;a,c) = \frac{1}{1+e^{-a(x-c)}} \tag{3-4}$$

Depending on the sign of parameter *a*, this S-shaped curve pictured in Figure 3-4, is inherently open-left or open-right, thus being appropriate for representing concepts such as “very positive” or “very negative”.

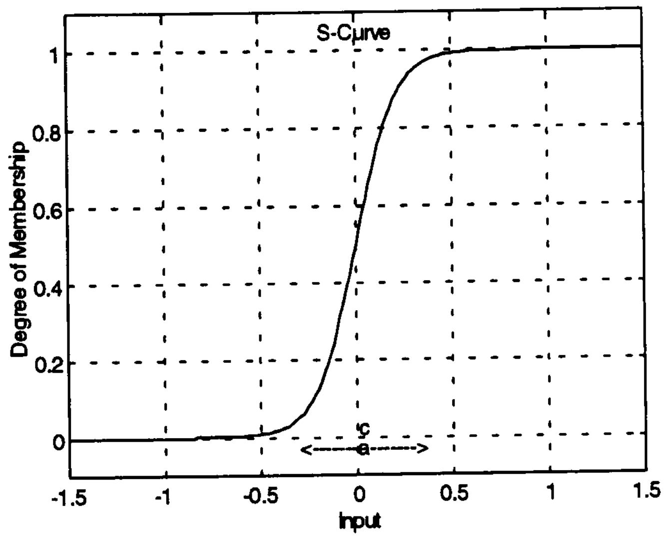


Figure 3-4. S-Curve Membership Function Shape

More conventional-looking membership functions may be created by multiplying or taking the difference of two sigmoidal functions provided that the a_n parameters are of opposite signs (multiplication case only).

$$MBF=f_1(x;a_1,c_1)*f_2(x;a_2,c_2) \quad (3\text{-}5)$$

$$MBF=f_1(x;a_1,c_1)-f_2(x;a_2,c_2) \quad (3\text{-}6)$$

As either resultant is similar to the trapezoidal shape, the same symmetry and core-to-support ratio restrictions will apply. One minor caveat to the second restriction must be noted. As the sigmoid function is defined by an exponential term, the shoulders between the core and boundary areas will be rounded. Thus, the core will be defined as any part of the shape which has a membership grade of 0.98 or above. Figure 3-5 illustrates the difference of two sigmoidal functions.

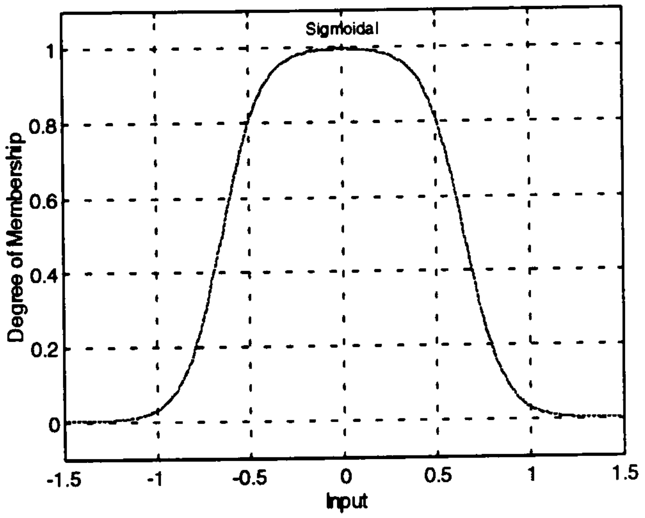


Figure 3-5. Sigmoidal Membership Function Shape

3.5. Cubic Spline Function

Cubic splines have an interesting history. Up until the earlier parts of this century, curves needed in aircraft or ship design were created using sweeps (large French curves) or by bending thin laths of metal or wood around pegs hammered into the floor. The curves were then drawn by tracing chalk around the laths or by scribing on sheets of aluminum. The laths naturally assumed a shape that was both aesthetically pleasing and mechanically sound. Resulting in a curve of minimum strain energy, they became known as splines, an East Anglican dialect word [13]. During the 1940s, a gentleman by the name of Shoenberg, developed the notion of a mathematical spline. It was derived from the physical spline by observing that, for small deflections, the shape assumed by the physical spline is closely approximated by a piecewise cubic polynomial.

Polynomials themselves have long been used to interpolate between data points. Unfortunately, it is often difficult for one polynomial to accurately approximate a data set with varying characteristics. Data sets with large amounts of variance require higher-order polynomials for approximation. This has a negative side effect as Greville [14] notes: "... it is a common observation that a polynomial of moderately high degree fitted to a fairly large number of given data points tends to exhibit more numerous, and more severe undulations than a curve drawn by a spline or a French curve." The power behind the cubic spline is that it is a *piecewise* polynomial, and thus may have different polynomial definitions between different data points.

Given a function f defined on interval [a,b] and a set of numbers, called knots, $a = x_0 < x_1 < ... < x_n = b$, a cubic spline interpolant S, for f is a function that satisfies the following conditions [15]:

1. S is a cubic polynomial, denoted by S_j, on the subinterval $[x_j, x_{j+1}]$ for each $j = 0,1,...,n-1$;

2. $S(x_j) = f(x_j)$ for each $j = 0,1,...,n$;

3. $S_{j+1}(x_{j+1}) = S_j(x_{j+1})$ for each $j = 0,1,...,n-2$;

4. $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1})$ for each $j = 0,1,...,n-2$;

5. $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1})$ for each $j = 0,1,...,n-2$; and

6. one of the following set of boundary conditions must be satisfied:

   a) $S'(x_0) = f'(x_0)$ and $S'(x_n) = f'(x_n)$ (clamped boundary)

   b) $S''(x_0) = S''(x_n) = 0$ (free or natural boundary)

   c) $S'''(x_1) \neq 0$ and $S'''(x_{n-1}) \neq 0$ (not-a-knot boundary)

This effort will utilize not-a-knot cubic splines with the restrictions of symmetry and equidistant knot placement. The not-a-knot boundary condition specifies that there be no discontinuity in the third derivative at the first and last interior data points. Therefore, the first and second polynomial pieces are derived from the same polynomial, as are the second-last and last polynomial pieces. A cubic spline representing a triangle is depicted in Figure 3-6. Note that a cubic spline is not necessarily restricted to this shape, but may take any form depending on the knot placement (i.e. trapezoidal, Gaussian and sigmoidal). However, for purposes of this effort, the cubic splines will be defined in the same manner as the triangular membership function for two reasons. Firstly, the cubic spline must be restricted to a consistent shape. Secondly, system control may be compromised at a triangular or trapezoidal membership function's transition points where the boundary regions reach zero (0) or unity

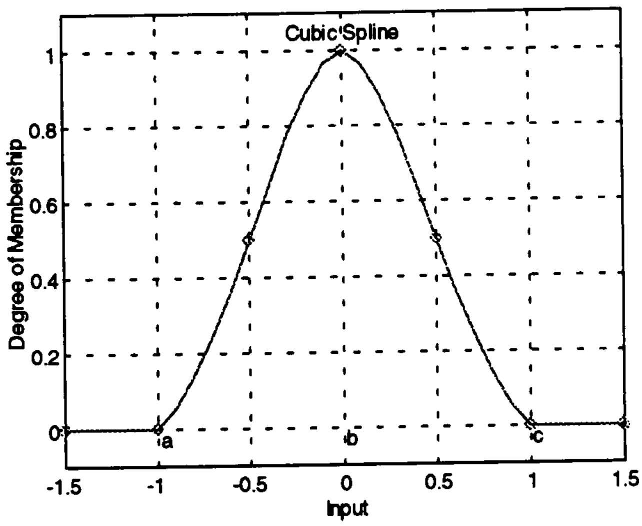


Figure 3-6. Cubic Spline Membership Function Shape

(1) and the function's derivative is discontinuous. It is possible that system control can be improved by a smooth transition at these points, which would result in a continuous membership function derivative. Cubic splines provide this smooth transition.

4. MEMBERSHIP FUNCTION ANALYSIS

This chapter details the work to be performed and contains the majority of fuzzy inferencing and analysis. Results of this chapter will be applied to the vehicle control model covered in Chapter 5.

4.1. Approach

Given a one-input, one-output system of the form shown in Figure 4-1, x and y are defined as the system input and output respectively, and f(x) the response function.

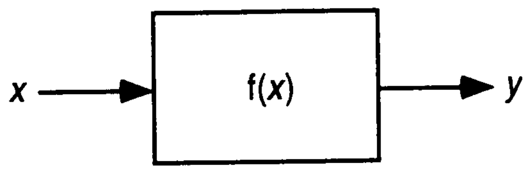


Figure 4-1. Input-Output System Definition

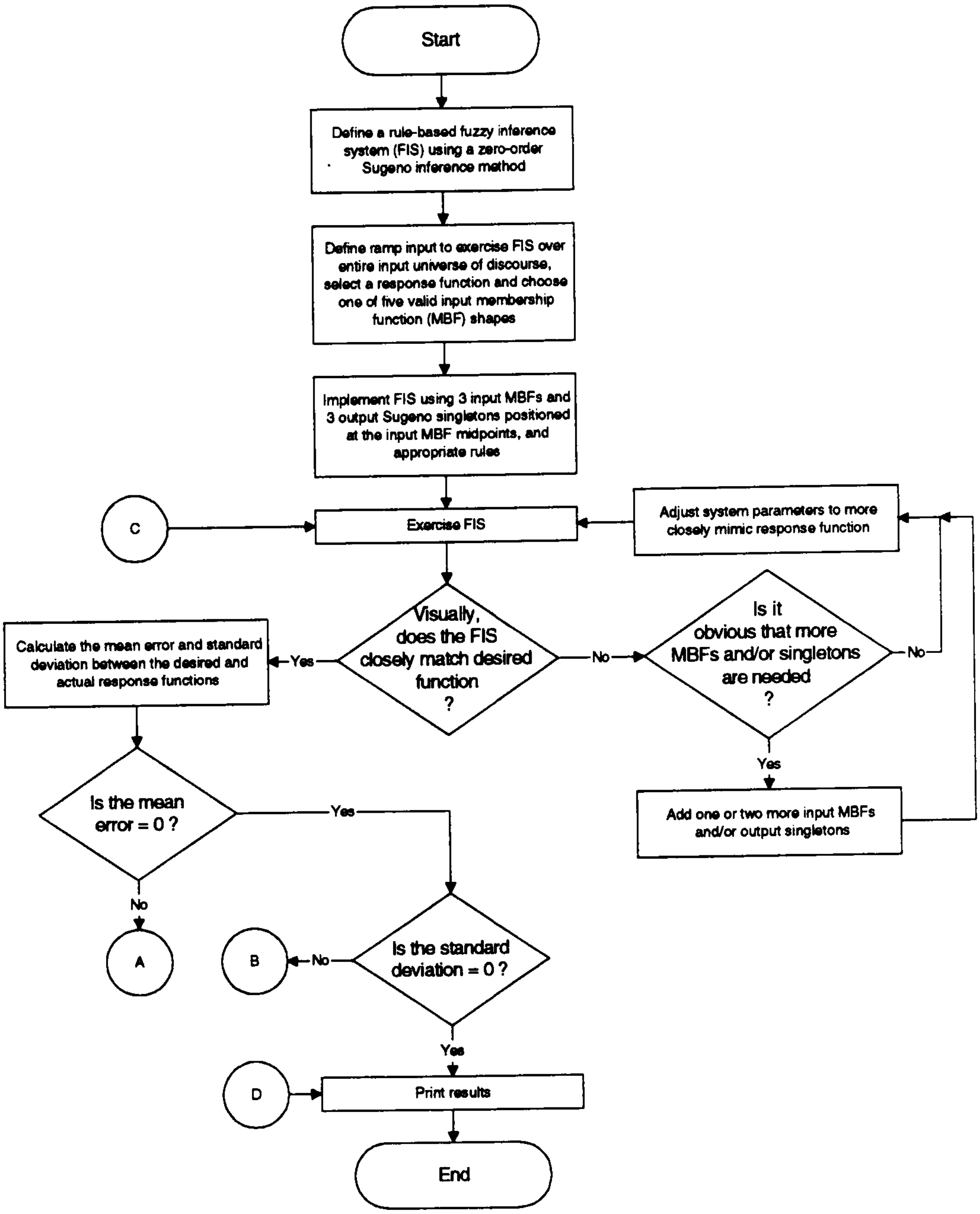
Let f(x) be defined over the interval $-10 \leq x \leq 10$. By implementing the inference method explained below in Section 4.2, a fuzzy rule-based system will be created to approximate f(x). Stated in another fashion, applying a given x in the allowed input range, it is expected that the inference process would produce a value quite close to the one expected by mathematically solving y = f(x). How well the fuzzy rule-based system approximates each response function is a measure of the quality of that system. In addition to the designer's competence, this quality is directly dependent on the membership function shapes used.

Each fuzzy system will be subjected to an error analysis. As an error analysis can be defined by using the combination of mean error and standard deviation, some ambiguity may arise by questioning the relative importance of each. For purposes of this work, the mean error will be zeroed-out ($\leq |0.01|$) by the inference process and only standard deviation will be addressed. This provides an objective standard in which to assess each membership function shape. A detailed explanation is provided in Section 4.6.

As many fuzzy inference systems will be created, a defined procedure and format must be followed for consistency and quality assurance purposes. Figure 4-2 delineates the design process to be followed.

4.2. Inference Method

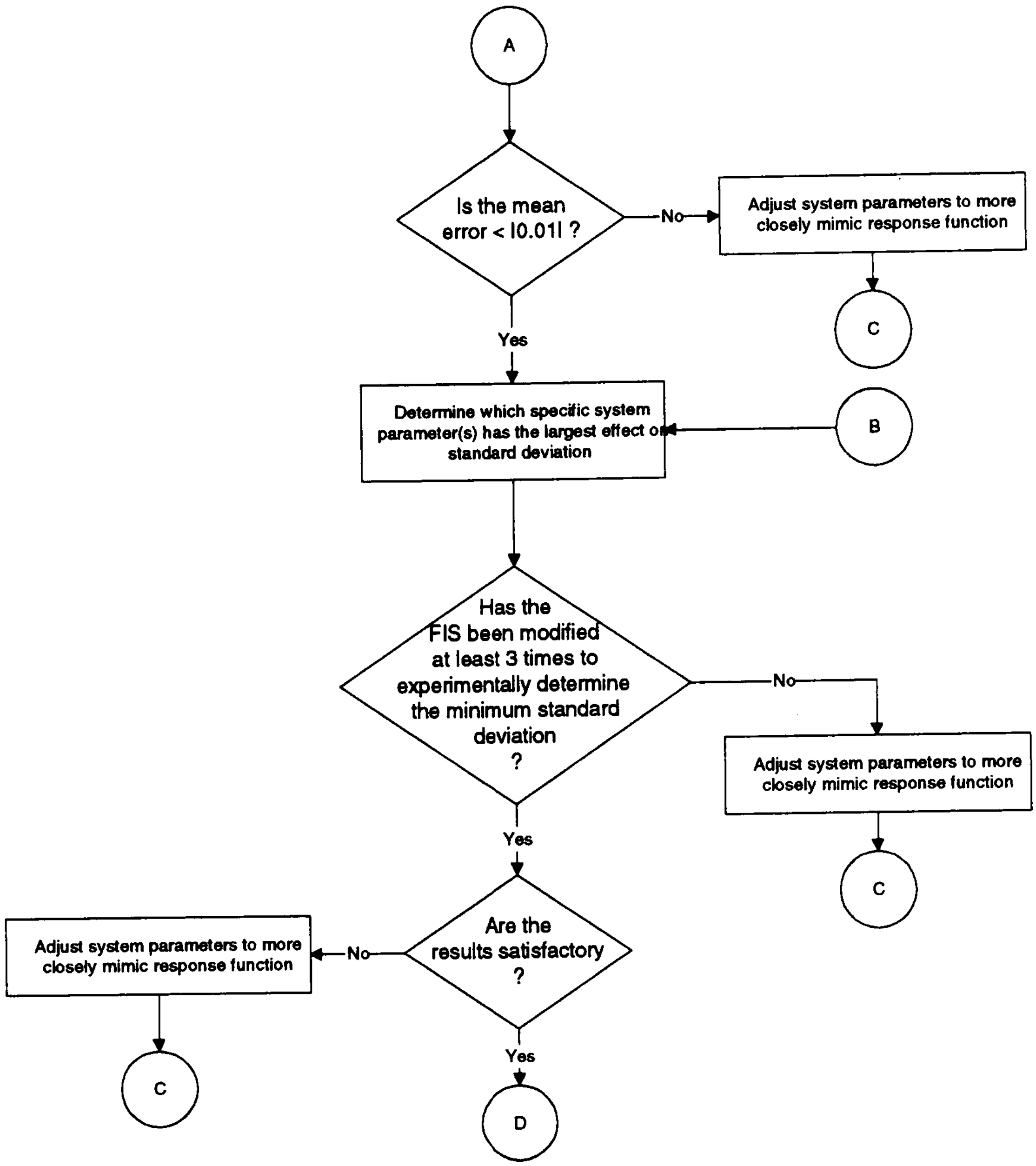
The fuzzy inference process referred to so far is known as Mamdani's fuzzy inference method. Based on Zaheh's pioneering work [2], Mamdani attempted to control a steam engine by synthesizing a set of linguistic control rules obtained by experienced human operators [11]. One characteristic of Mamdani's method is that it expects the output membership functions to be fuzzy sets, therefore requiring a defuzzification step in the inference process. Therein lies the major difference between the Mamdani inference method and



(a)

Figure 4-2. Fuzzy Inference Design Flowchart

(a) Main flowchart (continued on next page).



(b)

Figure 4-2. Fuzzy Inference Design Flowchart (cont.)

(b) Flowchart continuation.

the Sugeno inference method. First introduced in 1985, the zero-order Sugeno (or Takagi-Sugeno-Kang) method [16] utilized crisp singletons as output membership functions for fuzzy parking control using a model car. This singleton use enhances the efficiency of the inference process because it greatly simplifies the computation required to find the centroid of a two-dimensional shape. Rather than integrating across a varying two-dimensional shape to find the centroid, a weighted average over a few data points may be employed to calculate the crisp output. Therefore, no defuzzification is necessary. Adapting this inference method to the ICC example given in Chapter 2, Figure 4-3 illustrates the complete zero-order Sugeno inference process.

As before, rules 3 and 5 fire to the truth values of 0.2 and 0.5 respectively. Note that the fuzzification and inference procedures are the same as Mamdani's method. Denoting the NS and ZR singleton spikes by *a* and *b* respectively, the output value is:

$$t^* = \frac{a(0.2) + b(0.5)}{0.2 + 0.5} \qquad (4\text{-}1)$$

For purposes of analyzing the effects of membership function shape on system performance, Sugeno inference is ideal. Only the input membership function shapes are fuzzy, as the output shapes are simply singletons.

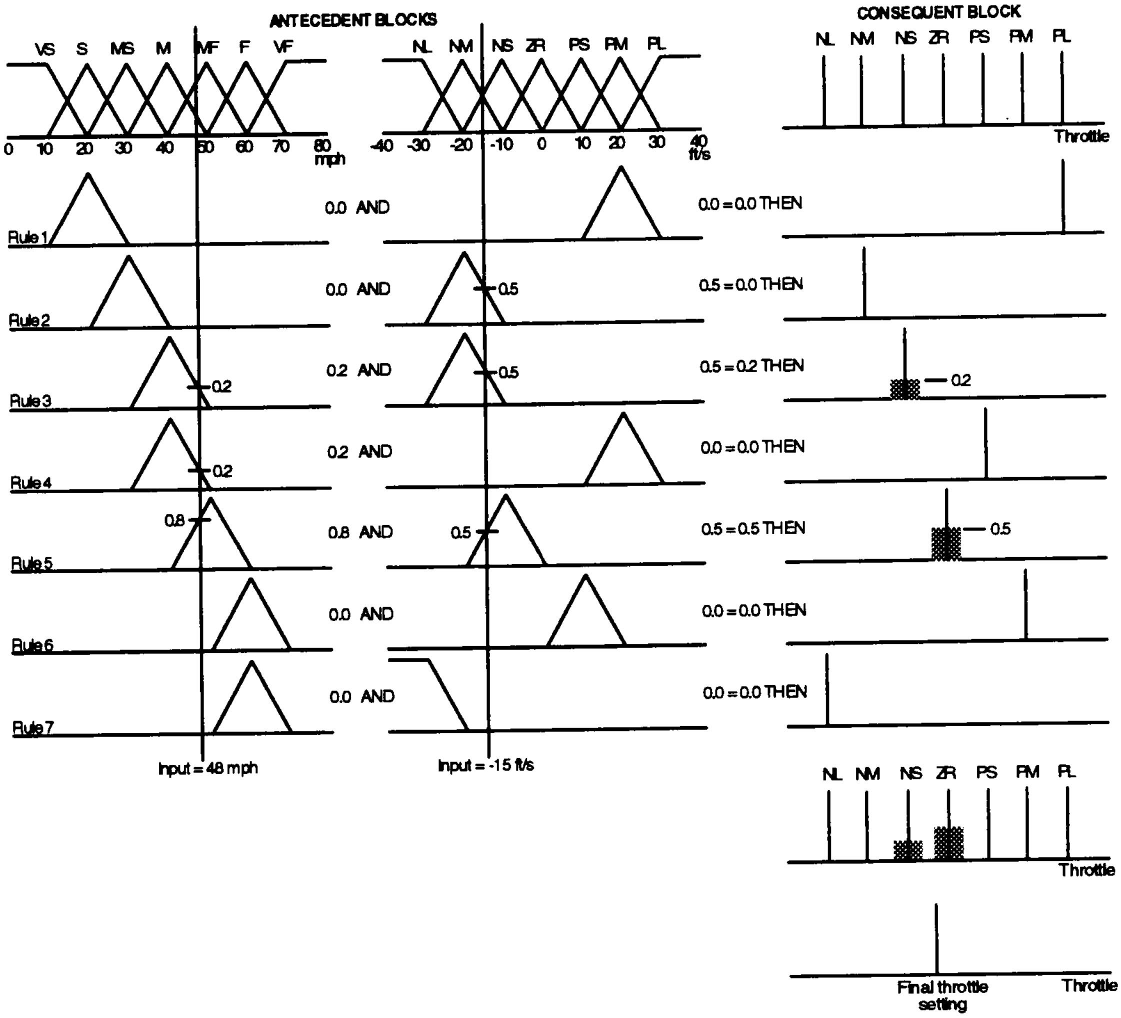


Figure 4-3. Zero-order Sugeno Fuzzy Inference Process

Therefore, Sugeno inference (requiring no integration in the defuzzification step) allows the removal of as much ambiguity as possible from the inference process. Results can not be skewed by having fuzzy sets in the consequent block. This results in the most accurate assessment of input membership function shape possible.

4.3. Rule Base

Modifying the rule-base format from Section 2.1.2 to account for zero-order Sugeno inference, a one-input, one-output system will contain a group of IF-THEN statements of the form

IF $z(t)$ is Z_n THEN $t(t)$ is k_n

where Z_n is a fuzzy set in the antecedent and k_n is a crisply defined constant (singleton) in the consequent. When the output of each rule is a constant, the similarity to the Mamdani inference method is striking. The only distinctions are the fact that all output membership functions are singleton spikes and the implication and aggregation methods are fixed, and can not be edited.

The more general first-order one-input, one-output Sugeno inference method contains a group of IF-THEN statements of the form

IF $z(t)$ is Z_1 THEN $t(t)$ is $m_1{*}z(t) + k_1$

where m_n and k_n are constants. Each rule can be thought of as defining the location of a "moving singleton". Thus, the output spikes can "walk around" in the output space. Mathematically stated, the output of each rule is a function of t for the given input region. This method allows for very accurate control and compact system notation. However, it is more difficult to analyze the effects of input membership function shape on system performance, and therefore, this method will not be used.

A single-input fuzzy system must contain as many rules as there are membership functions. The rules may use any antecedent/consequent relationships and the singleton spikes may be placed anywhere within the output universe of discourse.

4.4. Stimulus

To exercise each system over the entire input universe of discourse, a ramp function, shown in Figure 4-4, will provide the input stimulus.

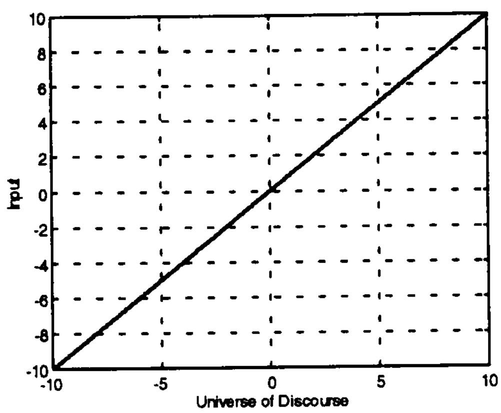


Figure 4-4. Ramp Input Stimulus

Since no feedback is used, a more complex input stimulus is not required.

4.5. Response Functions

Five response functions will be mimicked by each membership function shape. In all, 25 fuzzy inference systems will be created. The desired fuzzy system response is annotated with a "+" symbol. In subsequent sections, the actual fuzzy system response will be annotated with an "o" symbol.

4.5.1. First-order Response Function

The first-order response function, defined by: $y = -0.67x + 1$, is shown in Figure 4-5.

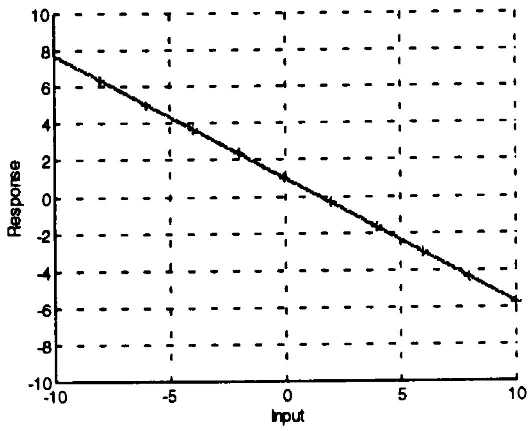


Figure 4-5. First-order Response Function: $y = -0.67x + 1$

This provides the most simple response function for the fuzzy system to track. Although simple in nature, a linear response such as this provides great insight into the nature of the membership function shapes and the transitioning between fuzzy sets.

4.5.2. Second-order Response Function

Actual control systems are often non-linear over some part of their universe of discourse. The inference mechanism's ability to handle these non-linearities requires different characteristics than for a linear response. Figure 4-6 illustrates the second order response function of: $y = 0.09x^2 - 0.75x - 8$.

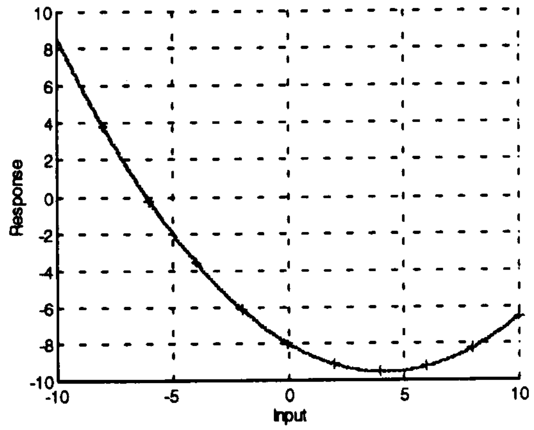


Figure 4-6. Second-order Response Function: $y = 0.09x^2 - 0.75x - 8$

4.5.3. Third-order Response Function

A more rigorous test is provided by approximating the third-order response function depicted in Figure 4-7. In this case, the fuzzy logic controller must be able to account for two fairly sharp changes in slope at $x = -5.5$ and at $x = 4.8$. This may be difficult to accomplish with the restriction of using no more than seven membership functions within the universe of discourse. The response to be approximated is defined by: $y = 0.035x^3 + 0.03x^2 - 2.75x - 1$.

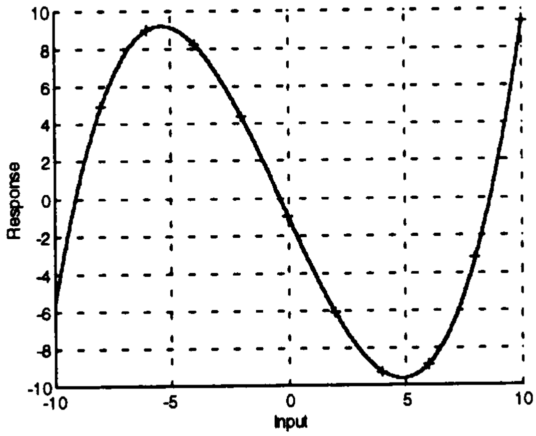


Figure 4-7. Third-order Response Function:

$$y = 0.035x^3 + 0.03x^2 - 2.75x - 1$$

4.5.4. Piecewise-linear Response Function

Discontinuities in the response function's first derivative provide the fuzzy system a different challenge in attempting to mimic a piecewise-linear response. This challenges each membership function shape to mimic sharp edges, which is unique to the five membership response functions. A total of eleven separate regions were needed to create the desired function. They are defined in Table 4-1.

In creating this function, there was a conscious attempt to require the system to act differently depending on the input's location within the universe of discourse. Except for two small regions, the inference system must produce a positive output independent of system input. This is shown in Figure 4-8.

Table 4-1. Piecewise-linear Function Definition

Region	Range	Definition
1	$-10 \le x \le -7.5$	$y = 6x + 53$
2	$-7.5 < x \le -6.5$	$y = 8$
3	$-6.5 < x \le -4.0$	$y = -4x - 18$
4	$-4.0 < x \le 0$	$y = 2.5x + 8$
5	$0 < x \le 1.5$	$y = -5x + 8$
6	$1.5 < x \le 3$	$y = 6x - 8.5$
7	$3 < x \le 5$	$y = 9.5$
8	$5 < x \le 6$	$y = -4x + 29.5$
9	$6 < x \le 7$	$y = 2x - 6.5$
10	$7 < x \le 8.5$	$y = -3x + 28.5$
11	$8.5 < x \le 10.0$	$y = x - 5.5$

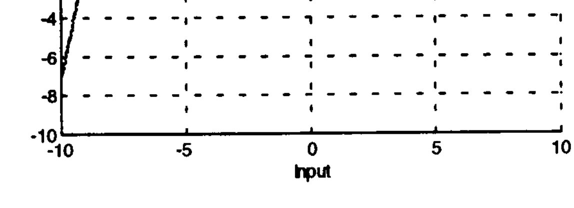


Figure 4-8. Piecewise-linear Response Function: $y = m_n x + b_n$

4.5.5. Sinusoidal Response Function

The greatest challenge will be provided by a multi-spectral sinusoidal response function. Fluctuations in the sine wave's amplitude and shape, not to mention varying period widths, make this a very difficult response to track. Defined by: $y = 5\sin(0.475\pi x) + 1.5\sin(1.125\pi x) + 0.5\sin(1.875\pi x)$, this function is pictured in Figure 4-9.

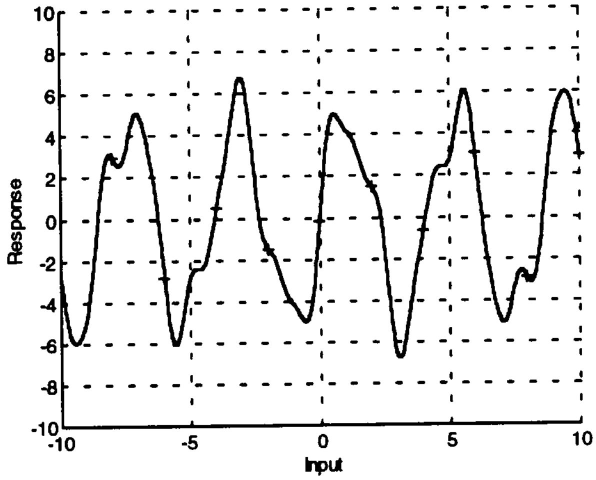


Figure 4-9. Sinusoidal Response Function:

$$y = 5\sin(0.475\pi x) + 1.5\sin(1.125\pi x) + 0.5\sin(1.875\pi x)$$

4.6. <u>Error Analysis</u>

When performing an error analysis, it is often desirable to summarize the differences between the theoretical output and the actual output by a few pre-defined numbers. One of these numbers is the average amount of error. This value is calculated by

$$\mu_s = \frac{1}{N}\sum_{i=1}^{N} error_i \qquad (4\text{-}2)$$

where the subscript *s* is a reminder that μ_s is the average of a set [17]. Given by way of example, Table 4-2 delimits a desired output and an actual output of a theoretical system.

Table 4-2. Error Analysis Data Table

i	Desired	Actual	Error
1	1	1.6	-0.6
2	2	1.9	0.1
3	3	3.3	-0.3
4	4	3.45	0.55
5	5	4.8	0.2

Using the equation above, the average error is calculated as:

$$\frac{(-0.6+0.1-0.3+0.55+0.2)}{5} = -0.01$$

While the average error provides the most-likely error value, or "center of gravity" of the set, it does not provide insight as to how much the numbers spread or deviate from this average. A value that provides this type of information is known as the standard deviation of a set, σ_s, computed by [17]:

$$\sigma_s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(error_i - \mu)^2} \qquad (4\text{-}3)$$

The summation is computed as follows:

$$
\begin{aligned}
(-0.6 - {}^{-}0.01)^2 &= 0.3481 \\
(\ 0.1 - {}^{-}0.01)^2 &= 0.0121 \\
(-0.3 - {}^{-}0.01)^2 &= 0.0841 \\
(0.55 - {}^{-}0.01)^2 &= 0.3136 \\
(\ 0.2 - {}^{-}0.01)^2 &= \underline{0.0441} \\
& 0.8020
\end{aligned}
$$

This results in a standard deviation of 0.4478 as computed by:

$$\sigma_s = \sqrt{\frac{0.8020}{4}} = 0.4478$$

Therefore, assuming a Gaussian distribution of output values centered about the mean (-0.01), 69% of the output values fall between plus and minus sigma ($\pm\sigma_s$). Figure 4-10 illustrates this point.

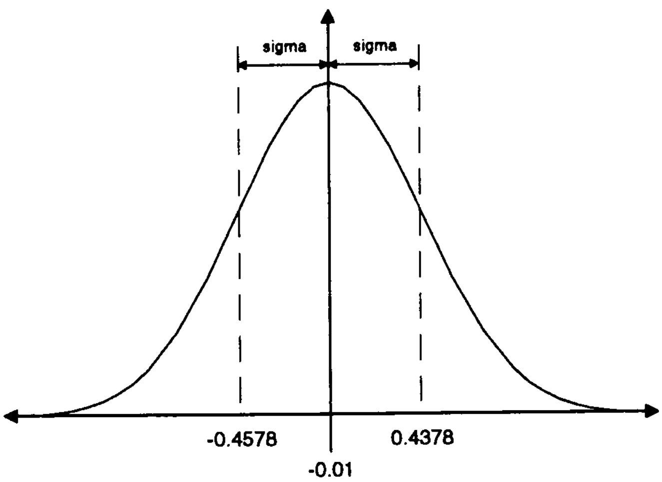
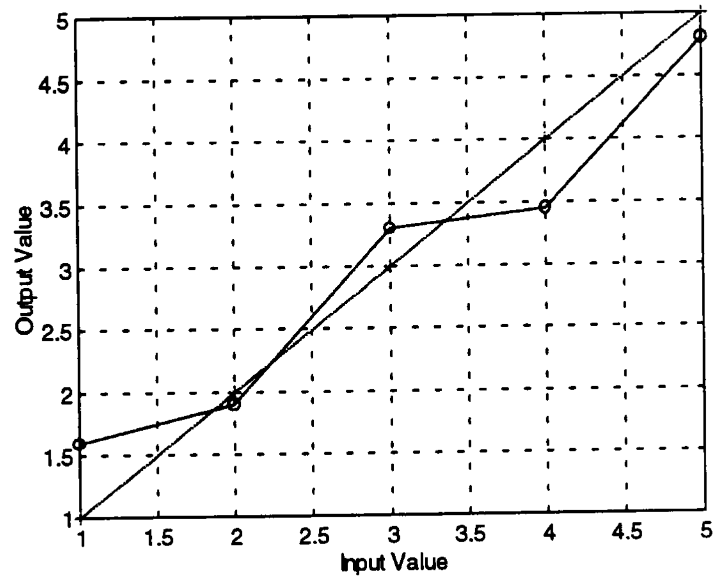
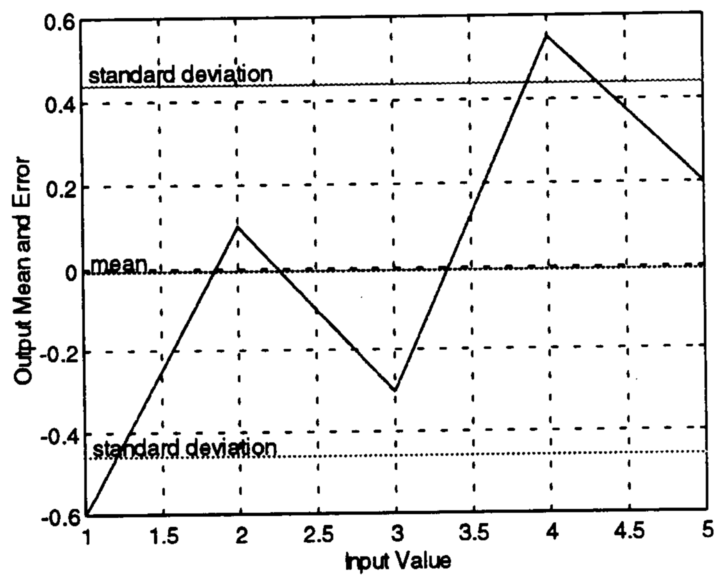


Figure 4-10. Gaussian Distribution Error Example

Applying these concepts to the given example, a plot of the desired versus actual data points appears in Figure 4-11 (a). Note that the two plots



(a)



(b)

Figure 4-11. Standard Deviation Example

(a) desired versus actual results, (b) error graph actually cross at x = 1.9, 2.2 and 3.4. This denotes when the desired response and actual response equal each other, thus resulting in zero error at those points. This can be shown by plotting the error between these two as illustrated in Figure 4-11 (b). This graph depicts the mean error, standard deviation ($\pm\sigma$), and actual error at each point within the universe of discourse. Note the three above-mentioned x values, where the error is equal to zero. At x = 1 and 4, the large amounts of error cause these values (and ones close to them) to be outside the standard deviation area. However, most fall within the $\pm\sigma$ boundaries.

For all intents and purposes, the mean error will be considered zero when the value is less than or equal to $|0.01|$. Upon driving the error to within that range, the standard deviation will be recorded. The compilation of all standard deviations for each membership function shape will then be averaged. The membership function with the lowest average will be considered the "best" membership function.

As with all simulation results presented in Sections 4.7 through 4.11, the actual results and error graphs will be presented in the body of the text. Input membership function and output singleton graphs are included in Appendix D. Appendix E provides a complete definition of each fuzzy system used.

4.7. Triangular MBF Simulation Results

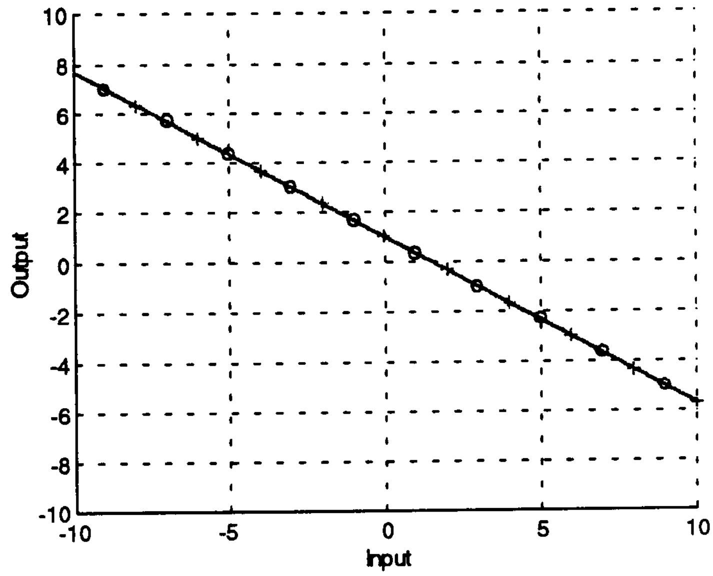
This section documents and details the simulation results for the triangular membership function. In general, this shape was easy to work with and provided a straightforward means of transforming crisp values into the fuzzy domain. However, limitations of this shape and restrictions defined in this scope of work did result in large amounts of error mimicking the more complex response functions.

4.7.1. First-order Response

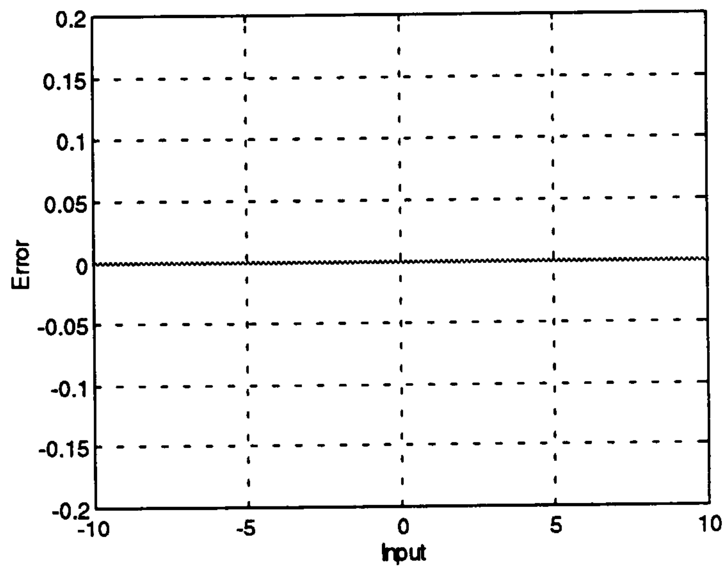
Using only three membership functions, this shape was able to duplicate the desired response function, resulting in zero standard deviation. This is shown in Figure 4-12. The membership functions were equidistant with 50 percent (%) overlap. Singletons were assigned values corresponding to each shape's normal point. With this configuration, a smooth transition between singleton values occurred as the ramp input was applied. Given that triangular membership function boundaries are linear by nature, it was very easy to reproduce a linear response function.

4.7.2. Second-order Response

Seven membership functions were used in two distinct ways to replicate the desired second-order response function. In the universe of



(a)



(b)

Figure 4-12. Triangular MBF First-order Response

(a) actual results, (b) error graph

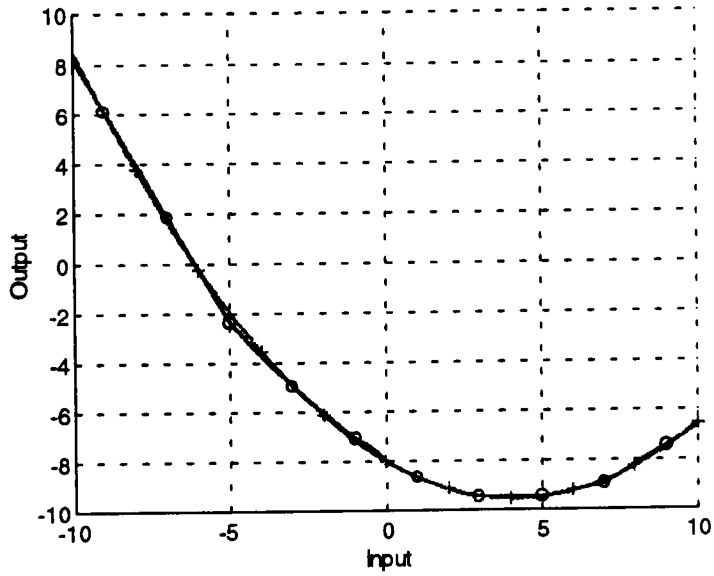
discourse's negative region, three shapes were placed similarly to the first-order response by not having more than two membership functions define any region in the universe of discourse and by placing the foot of one shape in the same "x" location as the core of the adjacent shape. This provided a smooth, linear response, albeit resulting in a small amount of error about x = -5 as the desired response function is of a second order.

The remaining four shapes were used to mimic the parabola's vertex and surrounding region. In particular, three membership functions with almost 60% overlap enabled reproduction of the parabola's minimum section. Twice in the universe of discourse's positive region, three different membership functions defined a particular region within the universe of discourse. This scheme provided excellent results, with a standard deviation equal to 0.1225, and proved that triangular membership functions can indeed be used to create a non-linear output. Figure 4-13 summarizes the fuzzy output.

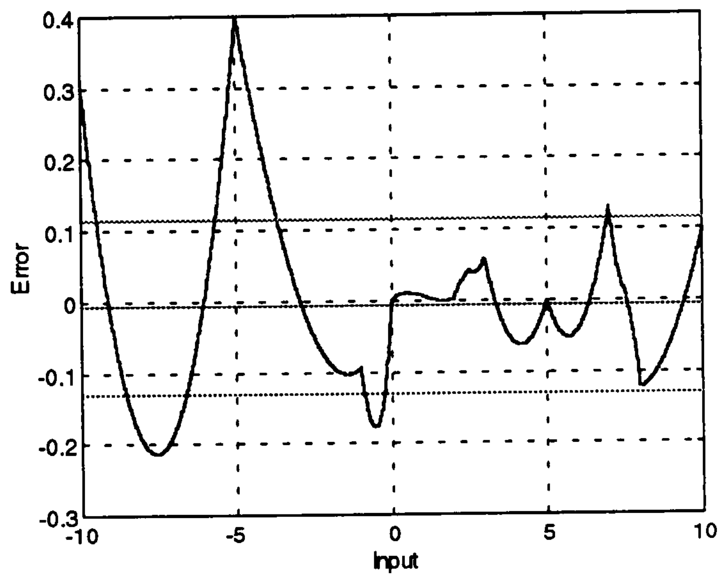
4.7.3. Third-order Response

This response function provided a challenge in mimicking its shape while placing several restrictions on the membership function placement due to its third-order nature. The restrictions are:

1. Center point of MBF "vlow" must equal -10;
2. Left foot of MBF "low" must equal -10;
3. MBF "med" must be defined from -4 to 4 to reproduce linear region;
4. Right foot of MBF "high" must equal 10; and
5. Center point of MBF "vhigh" must equal 10.



(a)



(b)

Figure 4-13. Triangular MBF Second-order Response

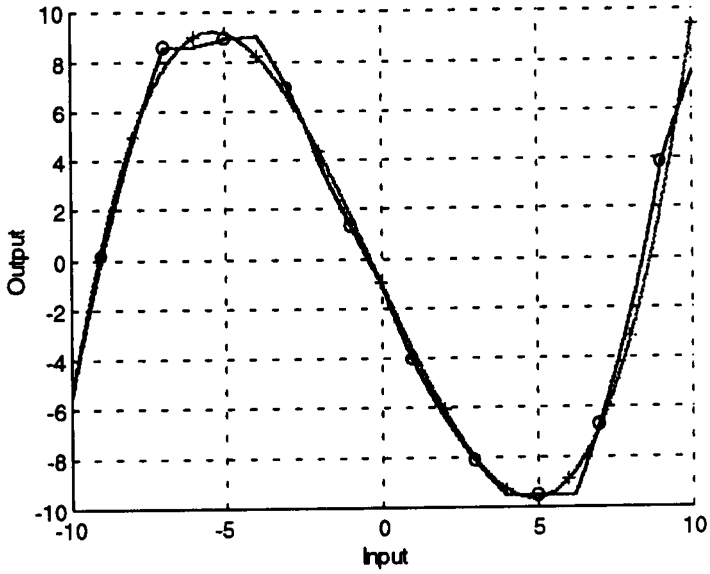
(a) actual results, (b) error graph

Thus, only the widths of membership functions "vlow" and "vhigh" and the membership function definitions "mlow" and "mhigh" may vary.

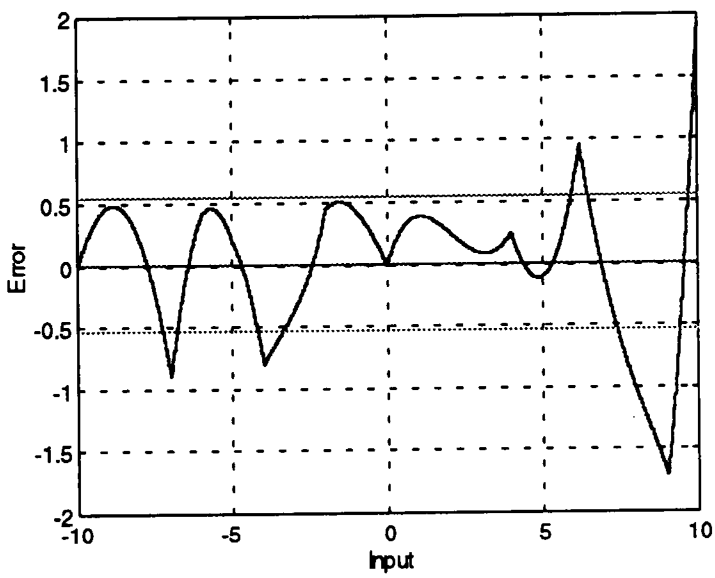
Using the non-linearity knowledge gained previously, Figure 4-14 displays the obtained results with a standard deviation of 0.5385. As only two membership functions could be used to mimic each vertex, a considerable amount of overlap provided the best results. An attempt was made to define each vertex with two narrow triangular-shaped membership functions, but this scheme resulted in a "boxier" output with a larger standard deviation. The trade-off, however, was a significant error between $9 \leq x \leq 10$. With the given restrictions, seven membership functions provided admiral results. Using an additional membership function at each vertex or non-symmetrical membership functions would each provide a small error reduction.

4.7.4. Piecewise-linear Response

Given enough membership functions, the fuzzy system could almost exactly reproduce the piecewise-linear response. The rationale in defining the input universe of discourse membership function placement was to position each membership function's core where there was a change of slope in the response function. Constant output areas would simply have one membership function defining that region (e.g. $-7.5 \leq x \leq -6.5$). The results are displayed in Figure 4-15.



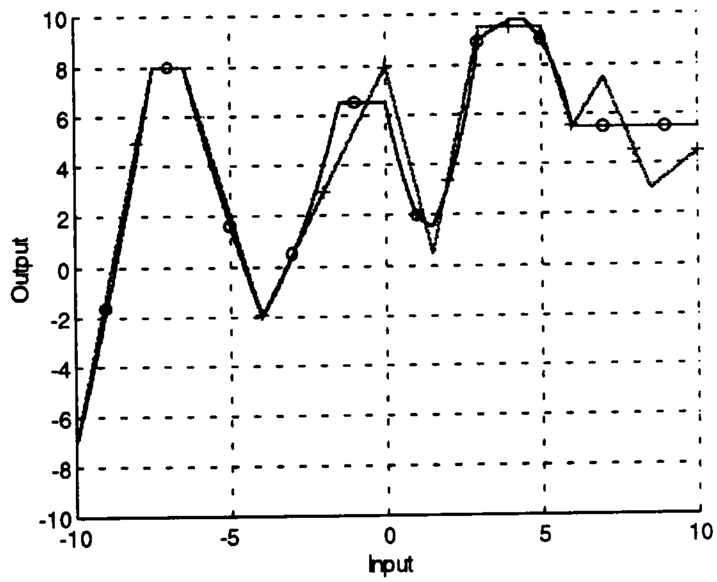
(a)



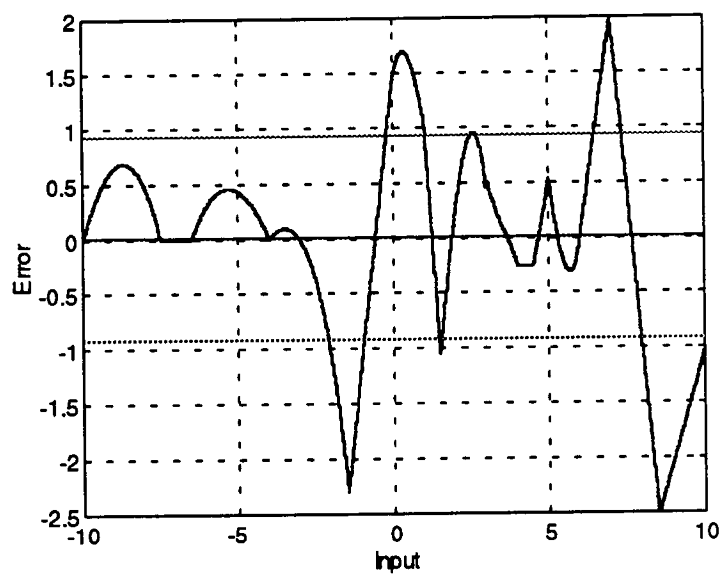
(b)

Figure 4-14. Triangular MBF Third-order Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-15. Triangular MBF Piecewise-linear Response

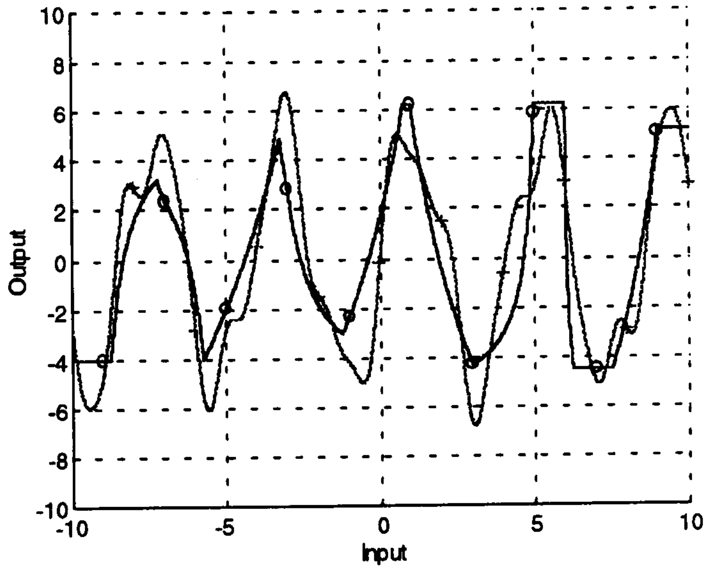
(a) actual results, (b) error graph

In order not to exceed seven membership functions while minimizing error, a triangular shape was required between $0 \leq x \leq 3$, instead of using it in the area $7 \leq x \leq 9$. Without this membership function placement, no vertex would have occurred at $x = 1.5$, thus resulting in more error than with the shape's original position. This begins to show some limitations in using this triangular shape.

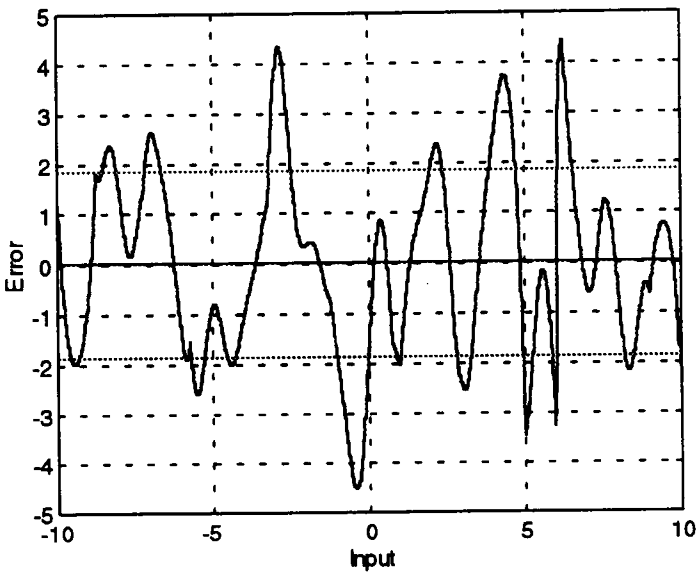
The middle region of $-2 \leq x \leq 2$ proved to be an interesting area in which to analyze the interplay of membership functions. The cores of membership functions "mlow", "med" and "mhigh" needed to be defined at $x = -4$, $x = 0$ and $x = 1.5$ respectively. Adjusting the membership function widths and corresponding output singleton values provided widely varying results. The challenge came from the slope between $-4 \leq x \leq 0$ was of a different absolute value than the slope between $0 \leq x \leq 1.5$. This proved to be a difficult task for symmetrical shapes to reproduce. The minimum standard deviation was experimentally obtained and equaled 0.9217.

4.7.5. Sinusoidal Response

Surprisingly enough, the triangular shape did a remarkable job in transforming the system input into the desired sinusoidal response. With Figure 4-16 depicting the final results, a unique shape placement was needed.



(a)



(b)

Figure 4-16. Triangular MBF Sinusoidal Response

(a) actual results, (b) error graph

The biggest challenge was how to make the limited number of membership functions create the large number of slope changes in the desired response. Inserting narrow triangles within a larger one provided the answer. This is best seen within the universe of discourse region of $-4 \leq x \leq 5$. A large "med" membership function centered about $x = 0$ provides the local maximum while the "mlow" and "mhigh" membership functions provide the local minima. The outer portions of the "med" membership function then work to raise the value of the output for the next required maxima. A large percentage of overlap even introduced a certain amount of non-linearity. A restriction in the number of membership functions precluded using this technique in the more positive universe of discourse region.

Driving the mean error to $\leq |0.01|$ proved no easy matter. This was best accomplished by level-shifting or "biasing" the fuzzy output up and down in different sections of the universe of discourse. A minimum standard deviation of 1.8547 was obtained.

4.8. Trapezoidal MBF Simulation Results

Trapezoidal membership functions provide the fuzzy system designer with easy-to-use linear boundary regions and a non-zero core to denote full membership over a range of values. This non-zero core proved very useful in many situations. As mentioned previously, the core-to-support ratio is 1:4.

4.8.1. First-order Response

Three membership functions incorporating approximately a 65% overlap proved to result in the minimum standard deviation for this first-order response.

As seen from Figure 4-17, there was a small amount of error in the regions of $-10 \leq x \leq -8$, $-2 \leq x \leq 2$ and $8 \leq x \leq 10$. This is due to the non-zero cores of each membership function. The best approach to achieve a linear output is for any two adjacent membership function boundary's slopes to be equal in magnitude and opposite in sign. Unfortunately, a small amount of error will occur when a boundary area overlaps the core of an adjacent membership function. However, a standard deviation of 0.0621 is reasonable.

Five membership functions were used to define the universe of discourse; however, this attempt actually resulted in a larger standard deviation. Each membership function's core was smaller, but the boundaries were defined by a larger slope, thus causing a larger standard deviation.

4.8.2. Second-order Response

Seven membership functions were used in much the same way as the triangular membership functions to mimic this second-order response resulting in a standard deviation of 0.1485. Using this shape in this scenario proved to be a mixed bag of advantages and disadvantages. The vertex of

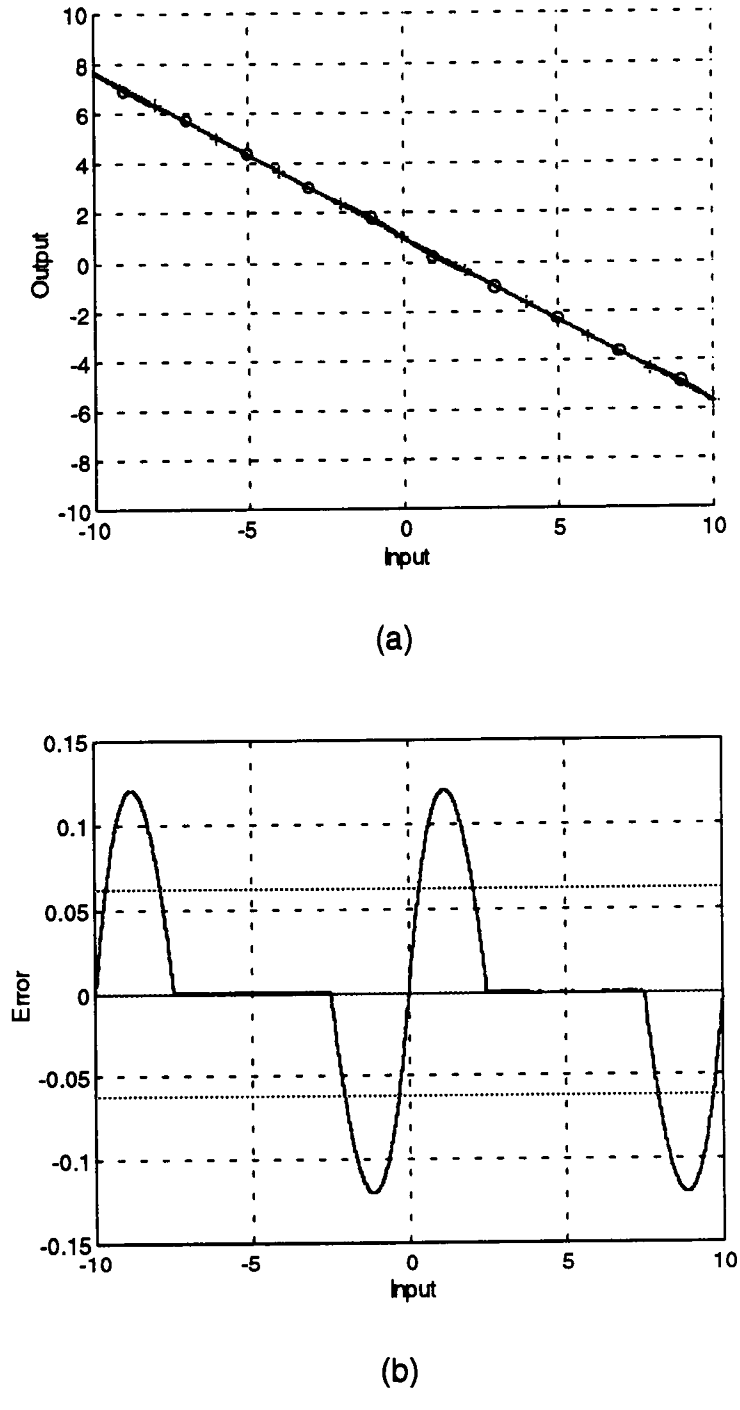


Figure 4-17. Trapezoidal MBF First-order Response

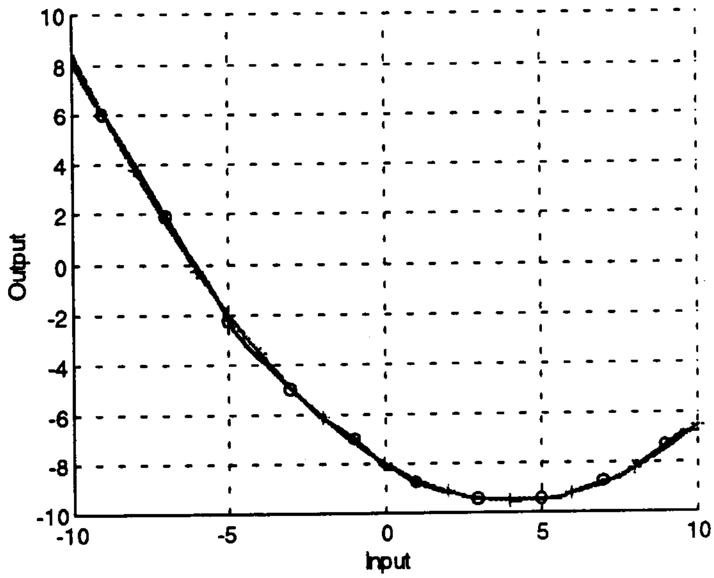
(a) actual results, (b) error graph

the parabola was nearly replicated simply by defining the "mhigh" membership function in the surrounding region of $x = 5$. In addition, the overall response function was straightforward to reproduce. Unfortunately, the core-to-support ratio caused some problems. Optimum placement of each membership function's cores caused excessive overlap, thus creating significant error in the areas about $x = -5$ and $x = -1$. This was minimized as much as possible.

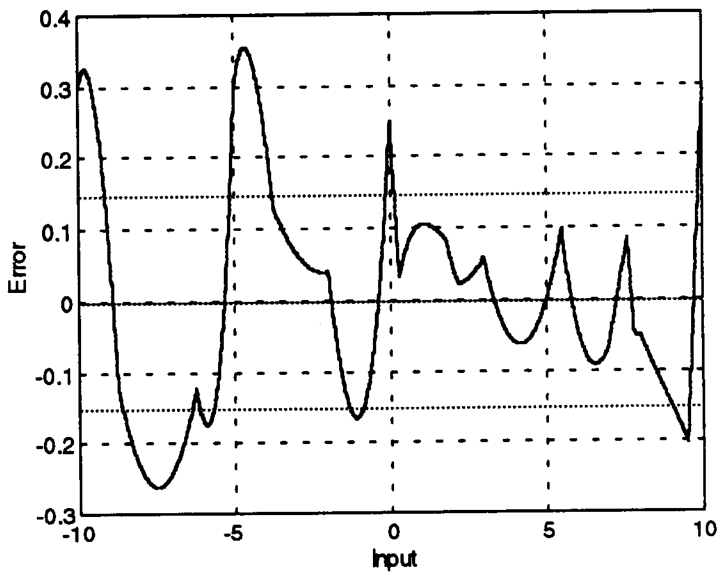
As an alternative to excessive overlap, the universe of discourse was defined by only five membership functions. While this eliminated some amount of error about the two above-stated points, the overall system did a poor job of reproducing a smooth, second order function. The best results are depicted in Figure 4-18.

4.8.3. Third-order Response

As the response functions became more complex, the non-zero core of a trapezoidal shape became more useful. Greater flexibility resulted and regions defined by three membership functions were easily created. Five membership functions actually did a respectable job re-creating this third-order response function. One was placed in the middle, one at each vertex, and one on each side. While producing a "boxy" output, it proved that much could be done with just five membership functions.



(a)



(b)

Figure 4-18. Trapezoidal MBF Second-order Response

(a) actual results, (b) error graph

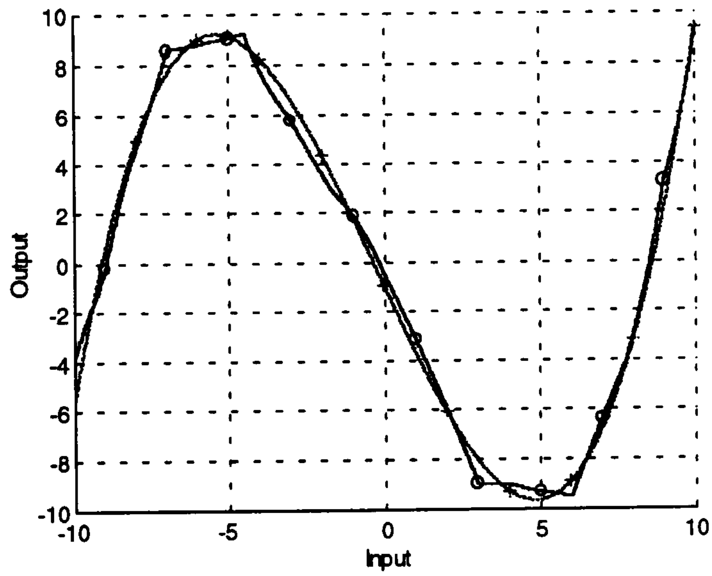
Seven membership functions were ultimately used to account for the non-linearity, resulting in a standard deviation of 0.5024. The output is shown in Figure 4-19. In order to mimic each vertex, two trapezoidal shapes were implemented with large amounts of overlap. Placement was critical in this case as several regions were defined by three membership functions. Moving any of the three resulted in widely varying results.

As can be seen in the membership function diagram, the "med" membership function was shifted to the left with its left-most support touching the "low" membership function. This caused the area between $3 \leq x \leq 4$ to be defined only by the "mhigh" membership function. While not desirable, this configuration resulted in the lowest standard deviation while adhering to the core-to-support ratio.

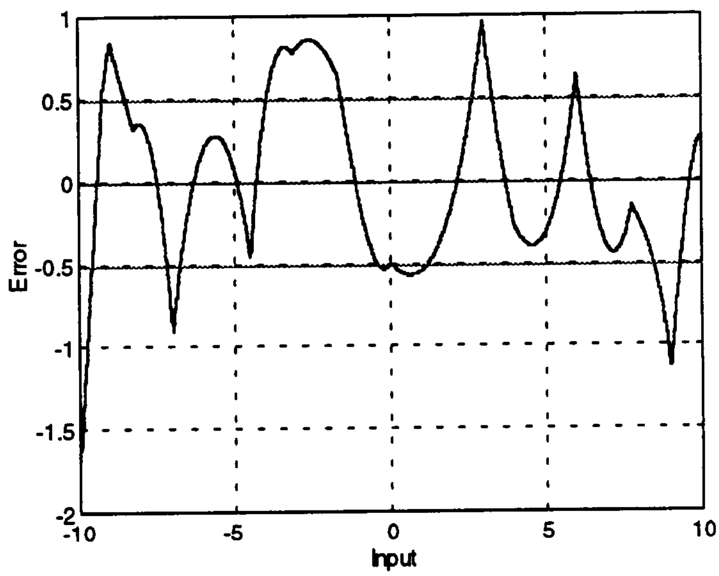
4.8.4. Piecewise-linear Response

A unique approach was taken to replicate this piecewise-linear response function with surprising results. The nemesis of the triangular-shaped membership function was that seven membership functions were just not enough to replicate the numerous linear regions in the universe of discourse.

The answer lied within using one membership function to account for multiple regions within the universe of discourse. A very wide "mhigh"



(a)



(b)

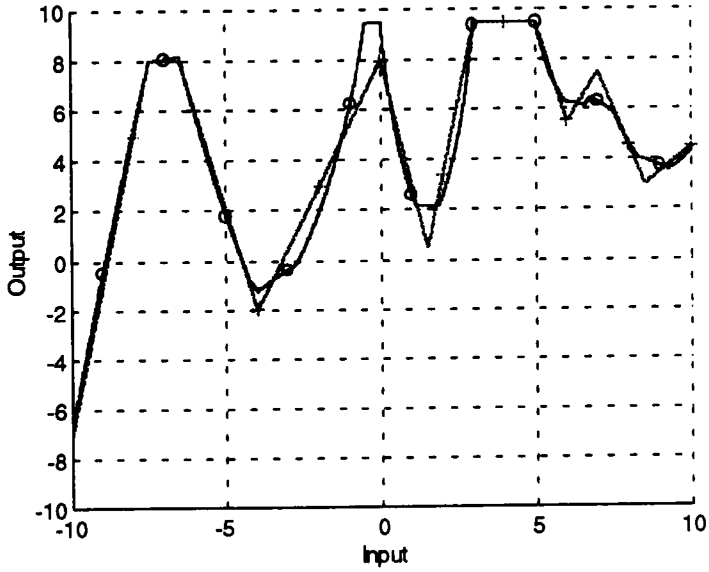
Figure 4-19. Trapezoidal MBF Third-order Response

(a) actual results, (b) error graph

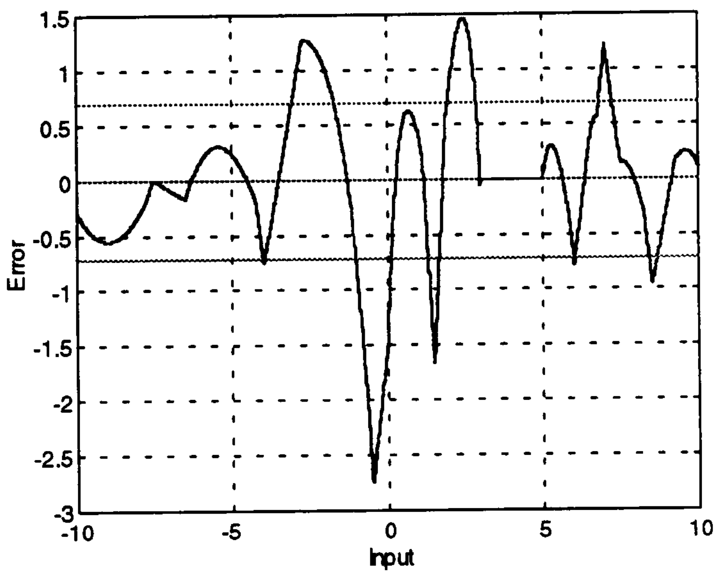
membership function was used not only to duplicate a constant-valued region, but also created some non-linearities which minimized the standard deviation. In the region of $0 \leq x \leq 3$, the "med" membership function was inserted within the larger MBF to replicate a local minimum. As both membership functions had similar boundary region slopes, a nearly linear output resulted. Choosing the "mhigh's" output singleton to be the same value as the $3 \leq x \leq 5$ region, a duplication of the desired response function was achieved. The interesting part lies in the region of $6 \leq x \leq 10$ using the "high" and "vhigh" membership functions. As the slopes of these two functions are very different from the "mhigh" membership function, a non-linearity was achieved by choosing the correct singleton values. This resulted in a standard deviation of 0.7045 with results accordingly displayed in Figure 4-20.

4.8.5. Sinusoidal Response

Once again, a very wide membership function was used to define multiple universe of discourse regions. This design procedure proved useful in several cases. The high-valued singleton associated with the "mhigh" membership function provided the positive fuzzy output. Narrower membership functions were then inserted within the wide membership function to provide output slope changes. Figure 4-21 displays the results which produced a standard deviation of 1.4848.



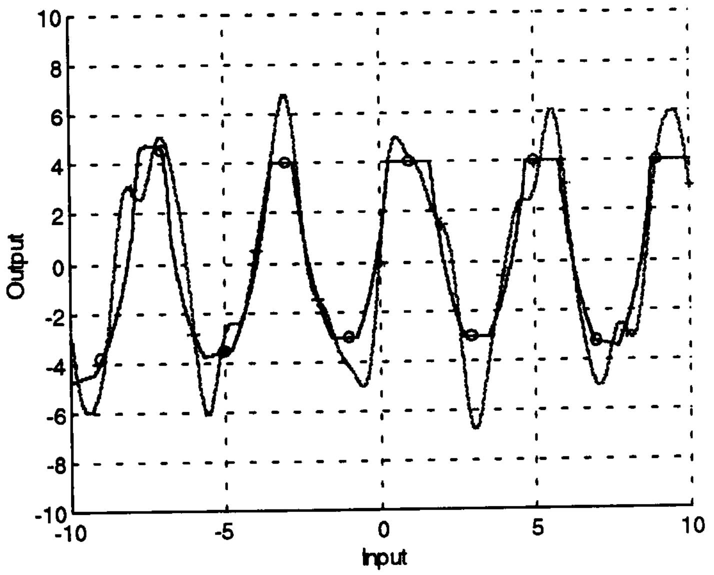
(a)



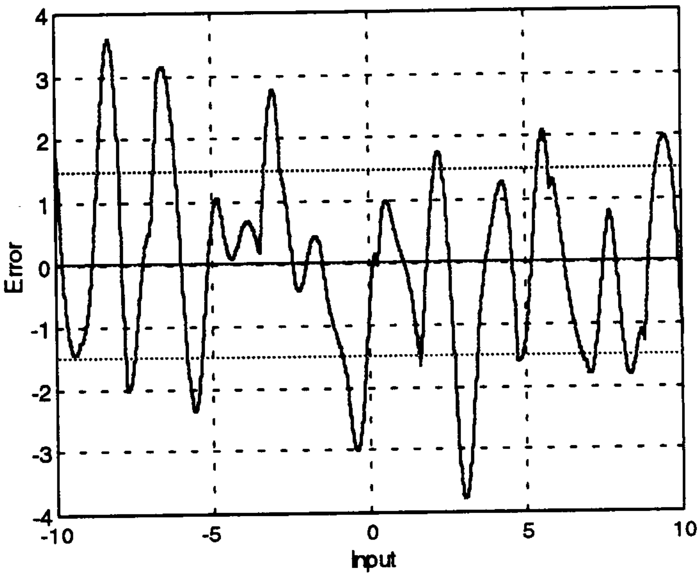
(b)

Figure 4-20. Trapezoidal MBF Piecewise-linear Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-21. Trapezoidal MBF Sinusoidal Response

(a) actual results, (b) error graph

Four of five sinusoidal maxima were simply created by only defining those regions with the "mhigh" membership function. While producing a flat-topped result, this allowed the other membership functions to be used for the sinusoidal minima. A non-zero core proved beneficial as the sinusoidal shape was composed of varying-width minima and maxima. The mean was driven to zero by using the "mhigh" output singleton as a level-shifting mechanism.

There are two areas of note. Firstly, the actual results were never able to achieve the full amplitude variation of ± 6 defined by the response function. Inserting one MBF within another precluded this. With the singletons defined between 0 and 10, a best-case scenario would result in a ± 5 variation. Secondly, the boundary slope differences between the "mhigh" and "low" membership functions created a nicely-curved sinusoidal peak. This helped minimize standard deviation.

4.9. Gaussian MBF Simulation Results

Gaussian shapes proved to be powerful, yet frustrating at times, to work with. As the shapes do not have strict regions of support, but tail off in either direction exponentially, movement of one membership function affects (to some degree) the entire universe of discourse. Due to this fact, rough estimates of the response function shapes were easily created, but close replication was much more difficult to attain.

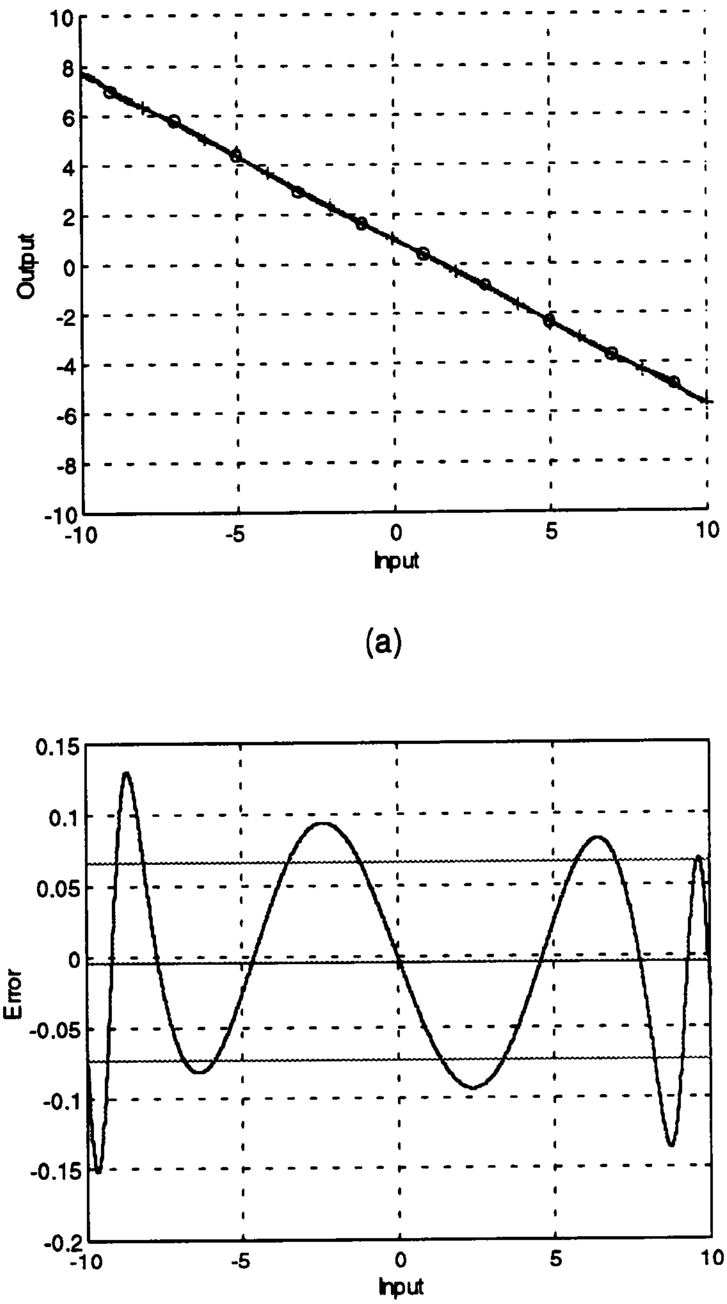
4.9.1. First-order Response

Reproducing a linear response was much harder than originally anticipated. In fact, a different train of thought was required to fully utilize this shape's characteristics. With the first difference already stated in Section 4.9, another paradigm shift was also required when overlapping these functions. Due to their exponential nature, a Gaussian membership function rapidly decreases in value near its endpoints. Likewise, the membership function rapidly increases in value near its center point. These two phenomena naturally create a curved, or non-linear output when overlapping Gaussian functions.

In this case, a large amount of overlap allowed the functions to cross each other in near-linear regions. The endpoints proved to be a problem however. Thus, "linearizing" membership functions with very small σ values were used to create a more linear response. Also, large overlap about $x = 0$ proved to negate the rounded response of the "med" membership function. Figure 4-22 depicts the response resulting in a standard deviation = 0.0701.

4.9.2. Second-order Response

Using the information gained from the first-order response, the second-order minimum standard deviation equaled 0.1211 with the results displayed in Figure 4-23. This effort utilized seven membership functions.

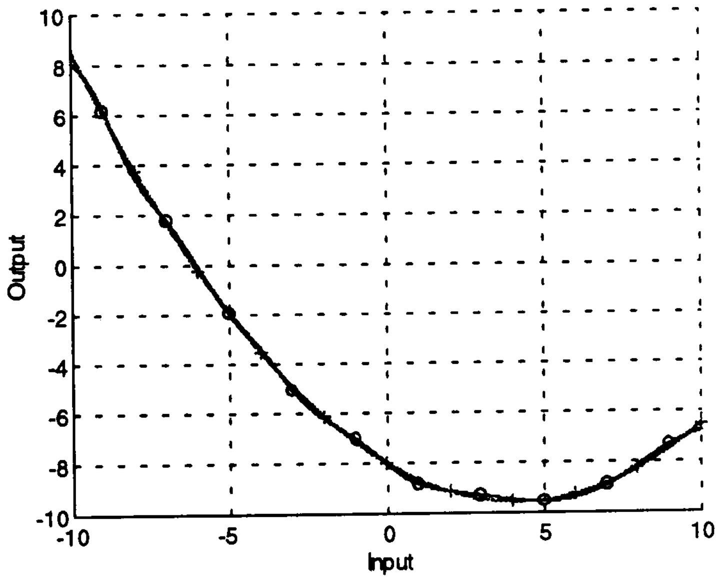


(a)

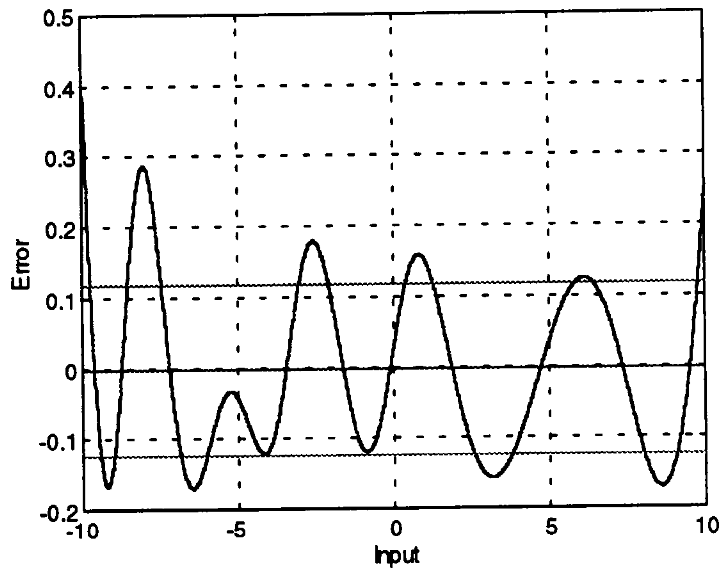
(b)

Figure 4-22. Gaussian MBF First-order Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-23. Gaussian MBF Second-order Response

(a) actual results, (b) error graph

A comparison of this input membership function's universe of discourse and that of the triangular or trapezoidal's proves to be very interesting. As expected, the triangular or trapezoidal membership functions are clustered about the response function's vertex. Three membership functions were used for the near-linear region and four were used about this minimum point. But as the interaction of the Gaussian functions naturally creates a curved response, the exact opposite design philosophy must apply to the Gaussian fuzzy system. Thus, four membership functions were needed in the near-linear region and only three were needed to approximate the vertex portion.

Not being able to take advantage of any "linearizing" membership functions, a small amount of error occurred around the endpoints. However, the Gaussian shape proved to produce the lowest standard deviation for this second-order response.

4.9.3. Third-order Response

Since the Gaussian shape performed so well in the previous case, original expectations were that once again, this membership function shape would enable the lowest standard deviation for a third-order response. However, this assumption proved to be incorrect.

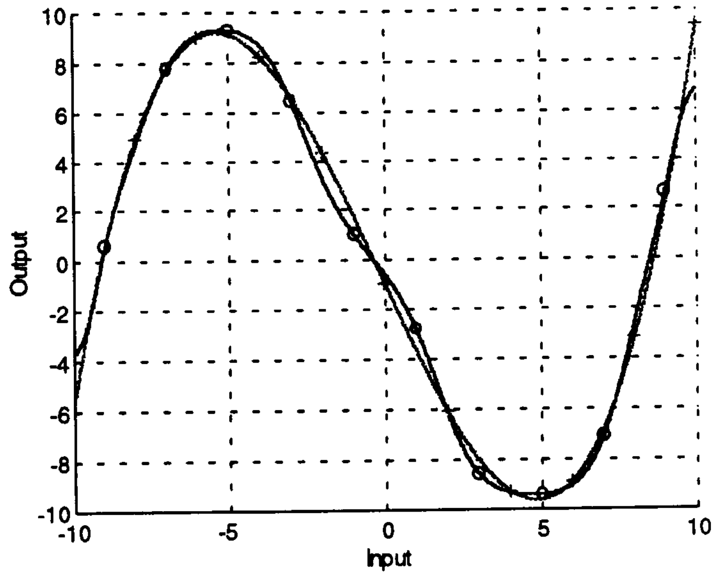
This response function contains three near-linear regions. As the Gaussian membership functions naturally provide a curved response, a unique design approach was again undertaken. Instead of using two membership functions each to define the corresponding vertices as previously used, one membership function proved adequate. A narrow "med" membership function was used in the center of the universe of discourse to mimic the center region. Having only used five membership functions to this point, the extra two were implemented as "linearizing" membership functions. This approach provided a lower standard deviation than using the extra two to minimize the error between $-3 \leq x \leq 3$. The response resulting in the standard deviation of 0.5168 is shown in Figure 4-24.

The trapezoidal-shaped fuzzy system provided the lowest standard deviation. This is due to that system's ability to effectively reproduce the near-linear regions. Nevertheless, the Gaussian shape proved to be very effective in reproducing the desired third-order response.

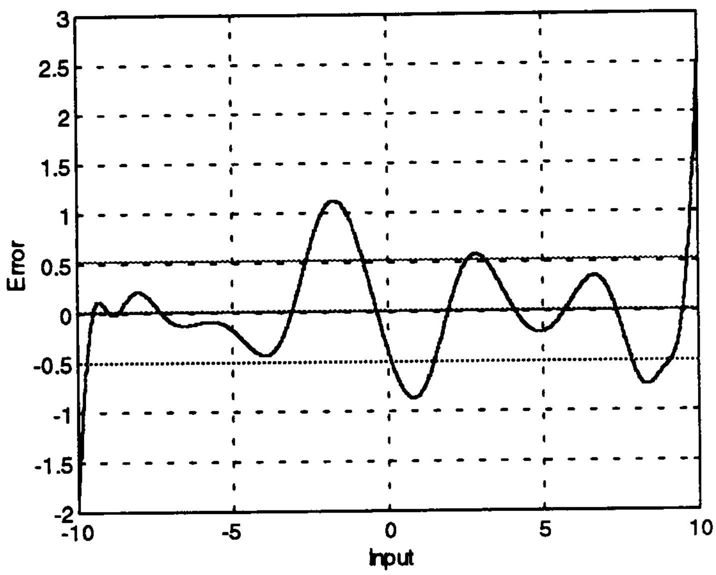
4.9.4. Piecewise-linear Response

The original expectation was that a Gaussian shape would not reproduce a piecewise-linear response function very well as the response's linear regions were obviously better suited to be mimicked by linear-boundaried triangular or trapezoidal shapes. Once again, an original assumption proved wrong.

As opposed to the two previous response functions, which required a different design philosophy, the input membership function universe of



(a)



(b)

Figure 4-24. Gaussian MBF Third-order Response

(a) actual results, (b) error graph

discourse was defined in a similar fashion as the trapezoidal system. Several narrow membership functions were inserted within one wide membership function to best recreate the desired response function.

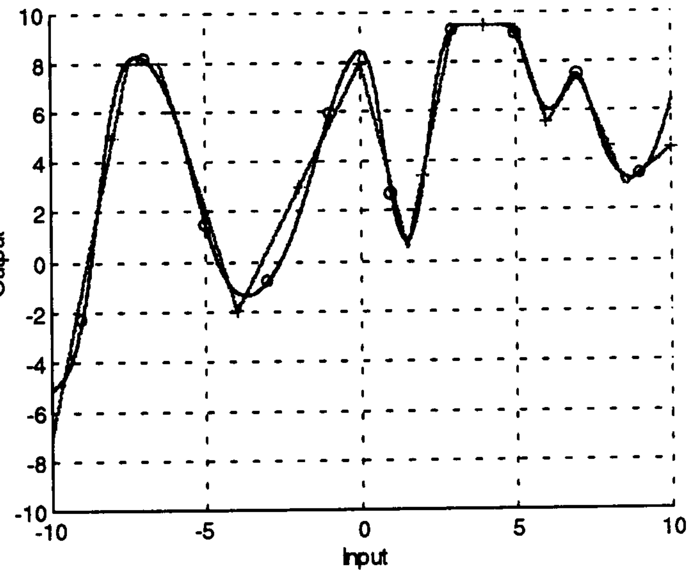
The driving factor enabling this shape to yield the minimum standard deviation result was the exponential nature of the Gaussian shape. As opposed to the trapezoidal shape, the degree of membership to the wide "mhigh" membership function decreased rapidly as the distance increased from the center point. This allowed the inserted membership functions to better shape the desired output.

As before, the endpoints proved troublesome. However, the Gaussian membership function shape performed an extraordinary job in reproducing the desired response function as can be seen from Figure 4-25. The minimum standard deviation achieved was 0.6710.

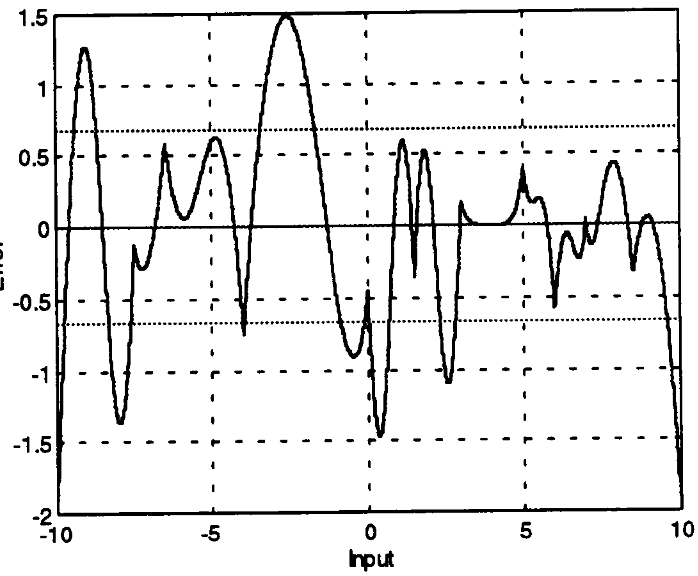
4.9.5. Sinusoidal Response

An original assumption finally proved correct. The Gaussian membership function shape imitated the desired sinusoidal response function better than any other shape with a minimum standard deviation of 1.2389. Figure 4-26 displays the results.

The varied membership function crossover points proved interesting in this exercise. Accepted fuzzy control design practices recommend a 40% to



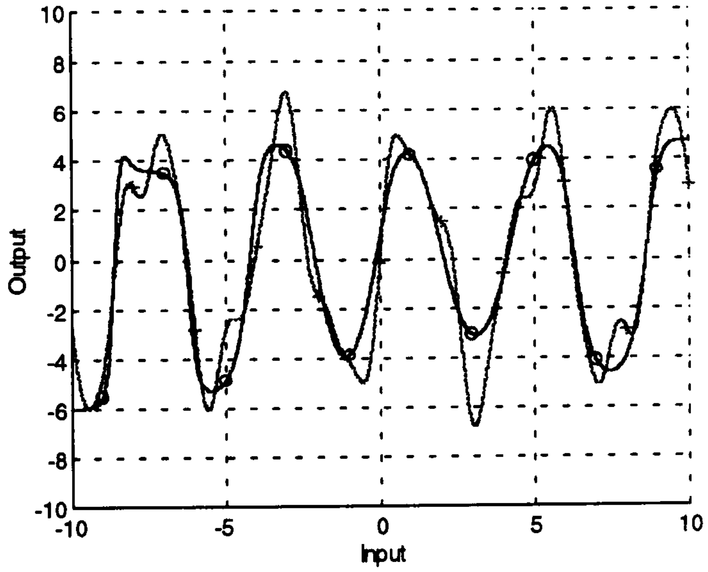
(a)



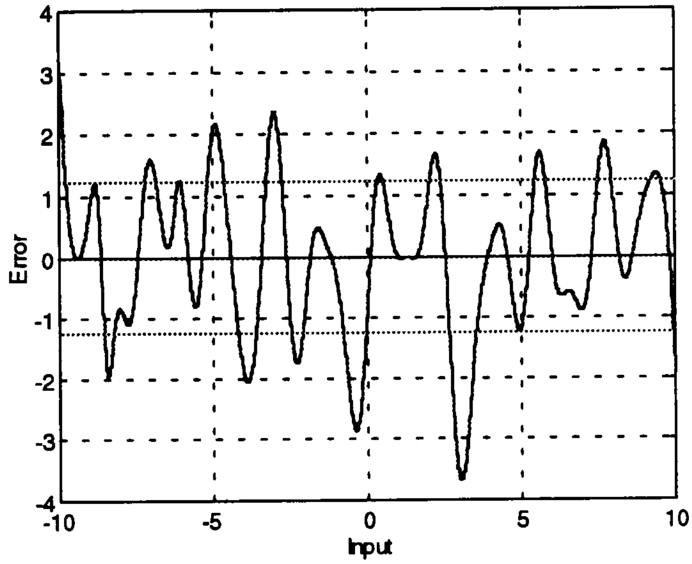
(b)

Figure 4-25. Gaussian MBF Piecewise-linear Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-26. Gaussian MBF Sinusoidal Response

(a) actual results, (b) error graph

60% overlap between membership functions. Two cases, in particular, varied from this practice, but yielded excellent results. Firstly, the "low" and "mlow" membership functions only overlap each other by approximately 25%. A more linear output was expected, but due to each membership function's narrowness and rounded tails, a tightly-curved, smooth vertex output resulted. Secondly, it was not known how the "mhigh" and "vhigh" membership functions would interact in the region of $9 \leq x \leq 10$. The Gaussian shape proved to adequately reproduce the desired response function, even with an overlap of less than 10%.

As with the piecewise-linear response, the wide membership function "mhigh" decreased rapidly from its center point and allowed the inserted membership functions to better affect the output. Therefore, peak-to-peak replication occurred better at each end of the output space than in the middle.

4.10. Sigmoidal MBF Simulation Results

The sigmoidal membership function proved challenging to work with. Using the difference of two oppositely-signed sigmoidal S-curves, a membership function could be formed with either a small core and large boundary, or a large core and small boundary. Unfortunately, with the normal shape restriction, a small core / large boundary membership function could not be created. The inability to create such a shape resulted in unwanted error in many situations.

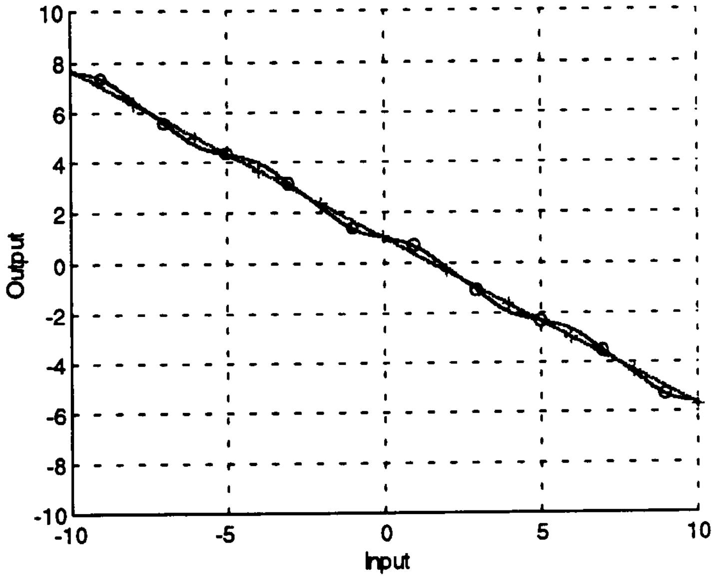
Strictly speaking, the core-to-support ratio was maintained at 1:4. However, the term "core" mentioned in this section will be loosely used to refer to the membership function's center section. This allows reference to pertinent features within the input universe of discourse without the drawback of verbosity. One important note must be made. Without the normality restriction, a sigmoidal membership function could be made to look and behave Gaussian, but all membership functions in that input space would need to be subnormal.

4.10.1. First-order Response

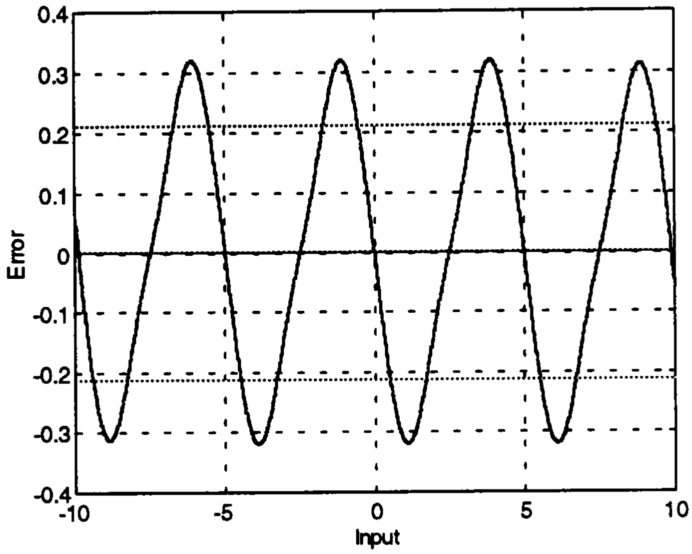
Three membership functions did not provide enough overlap between functions and seven membership functions led to the boundary regions being too steep and producing a highly fluctuating result. Both cases resulted in excessively large error figures. Five membership functions with greater than 60% overlap proved to produce the minimum standard deviation of 0.2129. The results are depicted in Figure 4-27.

4.10.2. Second-order Response

As this membership function is defined by an e^{-x} term, the general Gaussian rule of membership functions interacting in a non-linear fashion applies to this shape. As such, four membership functions defined the near-linear region and three membership functions defined the rounded vertex



(a)



(b)

Figure 4-27. Sigmoidal MBF First-order Response

(a) actual results, (b) error graph

portion of this second-order response. The results displayed in Figure 4-28 yielded a minimum 0.3939 standard deviation.

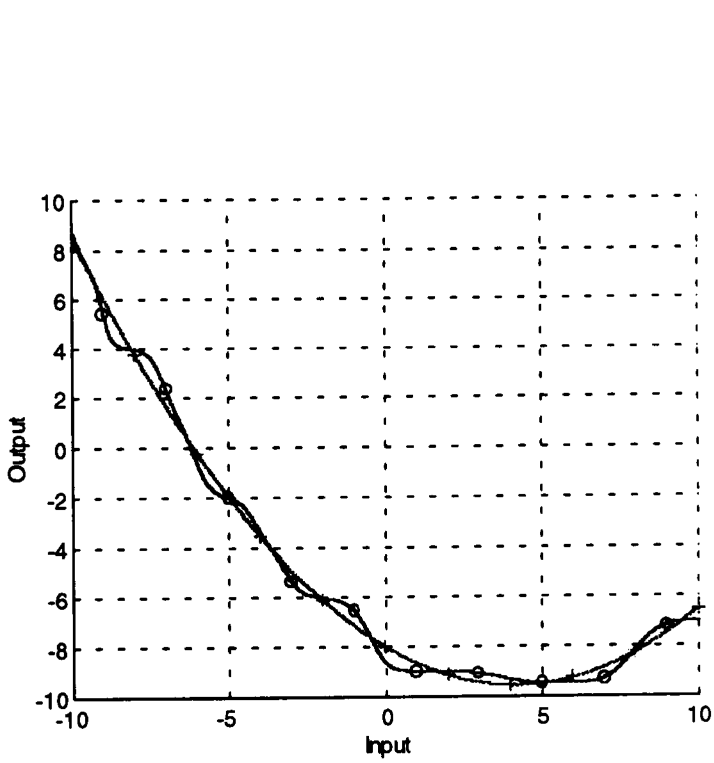
Overall, results were disappointing. Not only was the vertex area not formed well, but the near-linear portion did poorly as well. Also, the amount of overlap seemed to have little effect on minimizing output fluctuations. The only thing that worked well was the "linearizing" membership function, as defined in the Gaussian section, in the area of $-10 \leq x \leq -8$.

4.10.3. Third-order Response

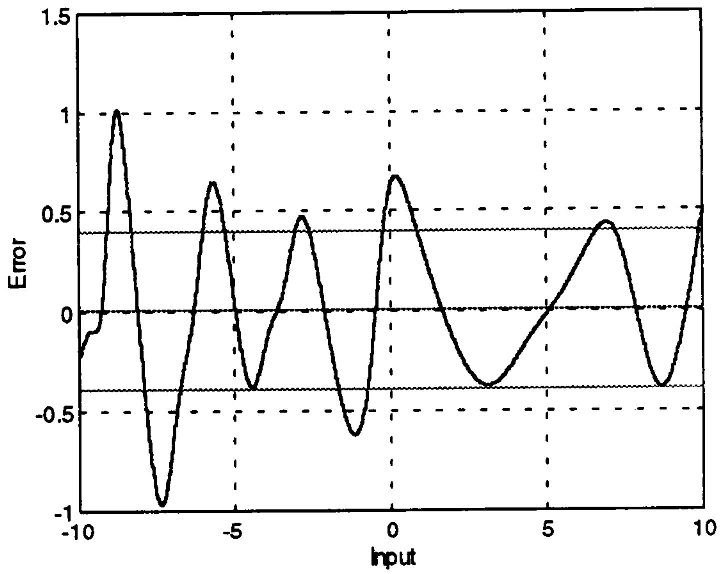
A similar design philosophy as used with the Gaussian shape was employed for the sigmoidal third-order response resulting in a standard deviation of 1.0769. As is the downfall for each sigmoidal result, the large core area caused unwanted error. This is especially true in the region of $-4 \leq x \leq 4$ as seen in Figure 4-29. With a 50% overlap, and only two membership functions defining any one point within this region, it was interesting to note the non-linearity in the result. The linearizing membership functions did provide some benefit, but overall, the results were once again disappointing.

4.10.4. Piecewise-linear Response

The piecewise-linear output generated by this sigmoidal fuzzy system did provide some encouraging results. The standard deviation was within



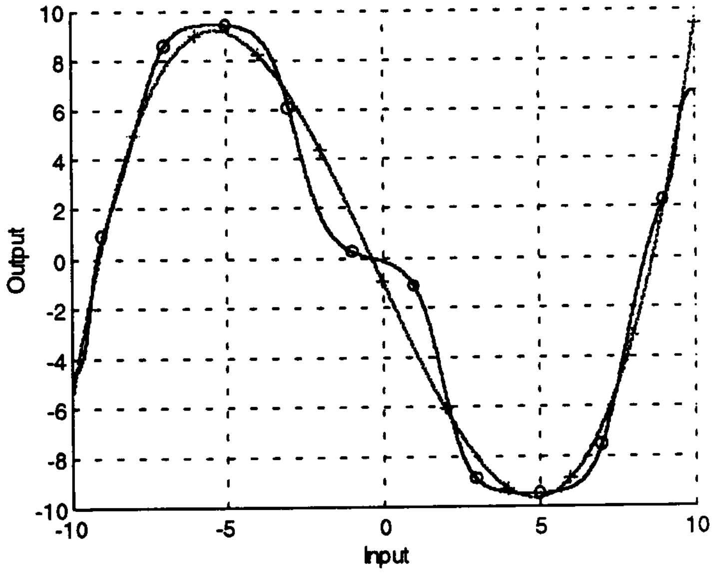
(a)



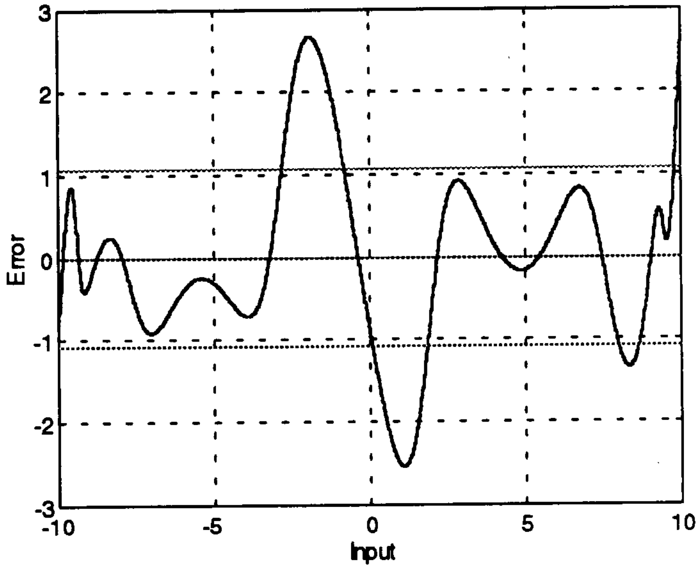
(b)

Figure 4-28. Sigmoidal Second-order Response

(a) results, (b) error graph



(a)



(b)

Figure 4-29. Sigmoidal MBF Third-order Response

(a) actual results, (b) error graph

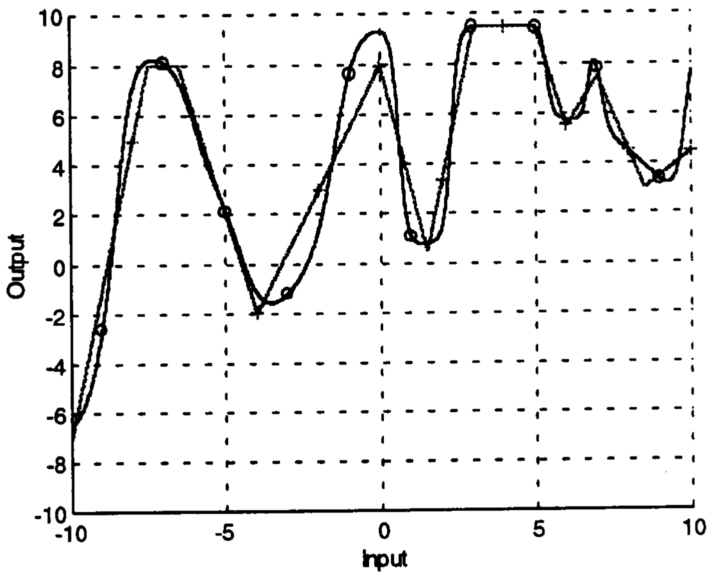
several tenths of the other membership function shapes with a minimum result of 1.1064. Figure 4-30 details the actual output.

Two very narrow membership functions provided interesting results. Centered at x = 1.5 and x = 6 respectively, their effect on the fuzzy system output is clearly seen. In both cases, a rounded vertex area with highly-sloped boundary areas resulted. Even narrow sigmoidal shapes came with their own set of challenges.

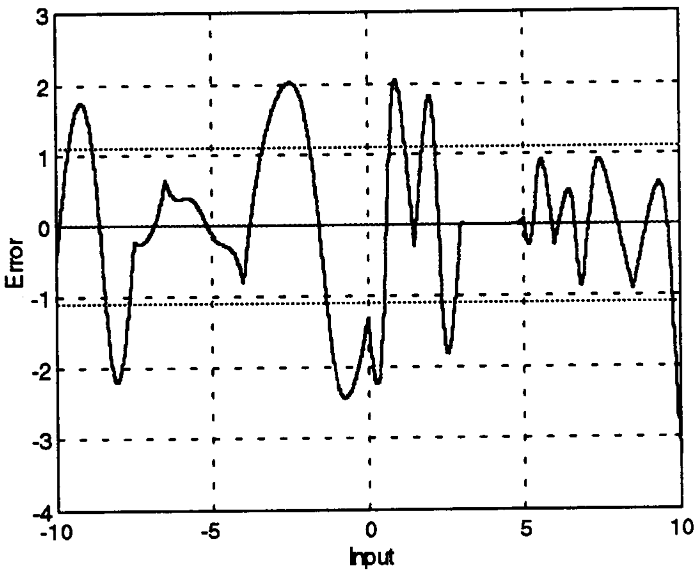
4.10.5. Sinusoidal Response

The one bright spot regarding the sigmoidal shape came with attempting to reproduce the last desired response. The rounded core shape lent itself well for use in replicating the minimum and maximum areas of the sinusoid. In fact, the sigmoid was only second to the Gaussian shape in standard deviation at 1.4345. The narrow-shaped membership functions proved more useful here than for the previous response.

Figure 4-31 displays the results using the same design philosophy as the trapezoidal and Gaussian membership functions. This output proved better than the trapezoidal scenario due to the exponential decay of the boundary regions. Unfortunately, the large core once again caused a larger error than the Gaussian scenario. Overall, the response function was the only one that the sigmoidal shape replicated well.



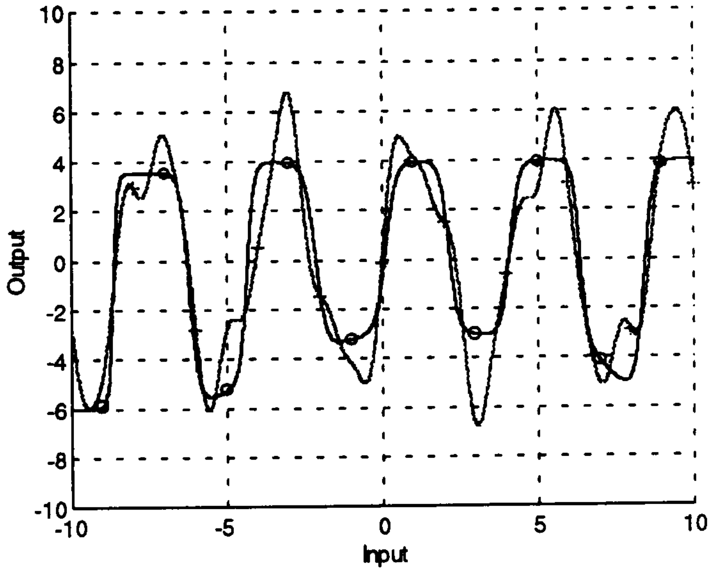
(a)



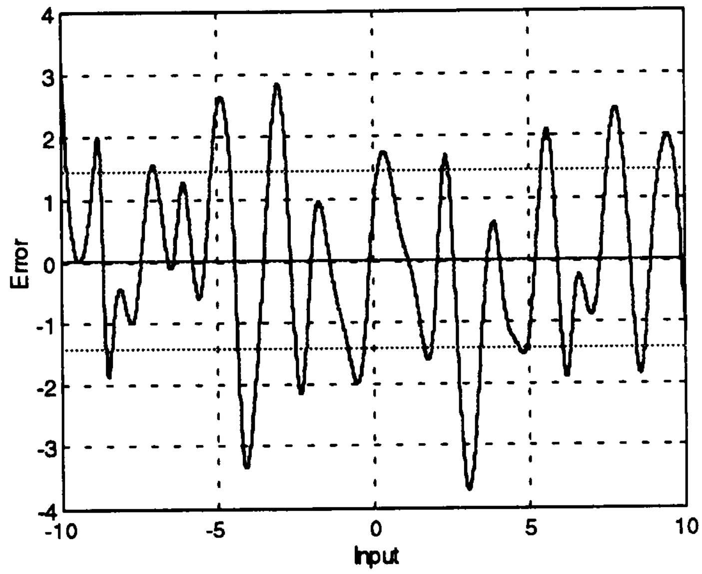
(b)

Figure 4-30. Sigmoidal Piecewise-linear Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-31. Sigmoidal MBF Sinusoidal Response

(a) actual results, (b) error graph

4.11. Cubic Spline MBF Simulation Results

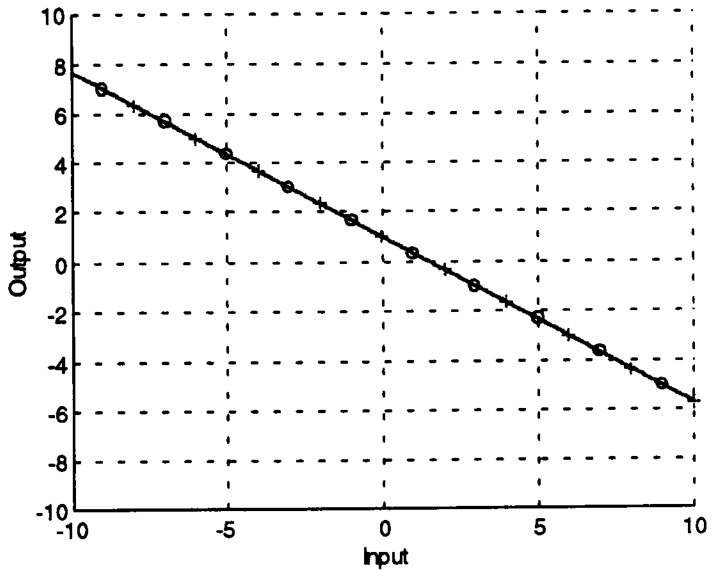
Given that the only real functional role of fuzziness and overlapping membership functions in a fuzzy rulebased system is to interpolate between output values, the cubic spline presents a way to compare the effects on system control between the triangular-shaped spline's rounded null and unity points and the triangular shape's sharp null and unity points. While overall results were similar, the cubic spline did have a slightly lower overall standard deviation.

4.11.1. First-order Response

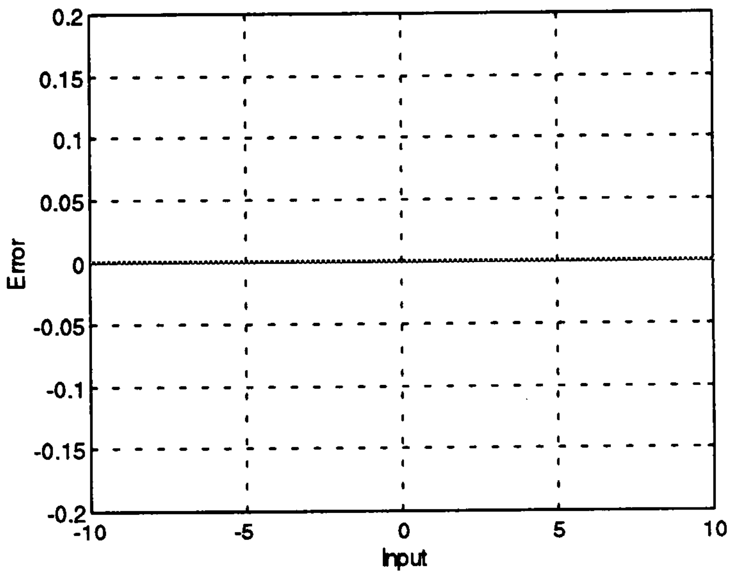
The cubic spline shape was the only other membership function beside the triangular shape to result in zero standard deviation. The results are displayed in Figure 4-32. It was expected that the rounded core area would produce a small amount of error. However, the adjacent membership functions had oppositely-rounded null points, thus canceling any error which may have occurred. As with the triangular fuzzy system, only three membership functions were required to define the universe of discourse.

4.11.2. Second-order Response

The rounded edges did not seem to provide any additional benefit for this second-order response function as Figure 4-33 shows. In fact the minimum standard deviation of 0.1312 was slightly worse than its triangular



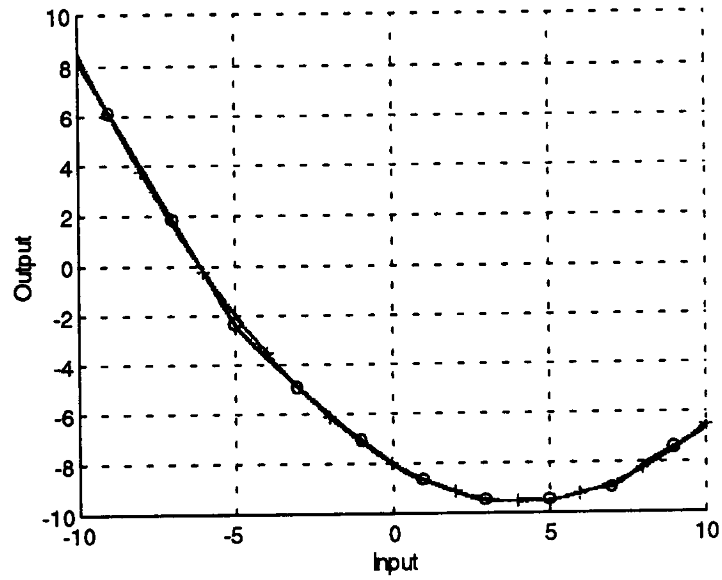
(a)



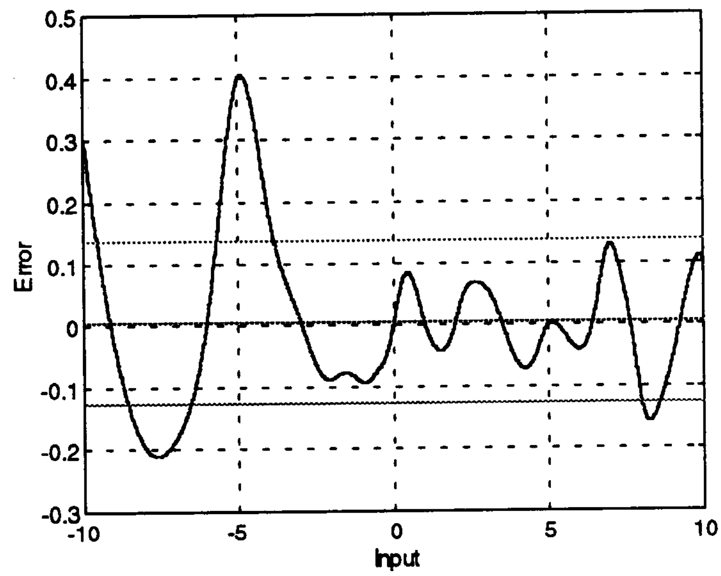
(b)

Figure 4-32. Cubic Spline MBF First-order Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-33. Cubic Spline MBF Second-order Response

(a) actual results, (b) error graph

counterpart. But overall, system functioning proved to be more than adequate.

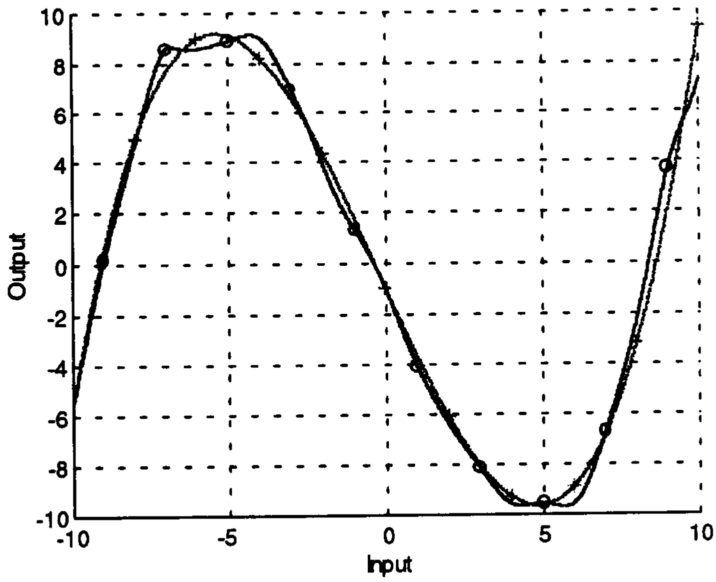
4.11.3. Third-order Response

Compared with the triangular membership function, a much smoother output from this shape resulted in replicating this third-order response. This is seen as desirable for a feedback control system as it would be less stressful on the machinery being controlled and would produce fewer harmonics and other detrimental system signals.

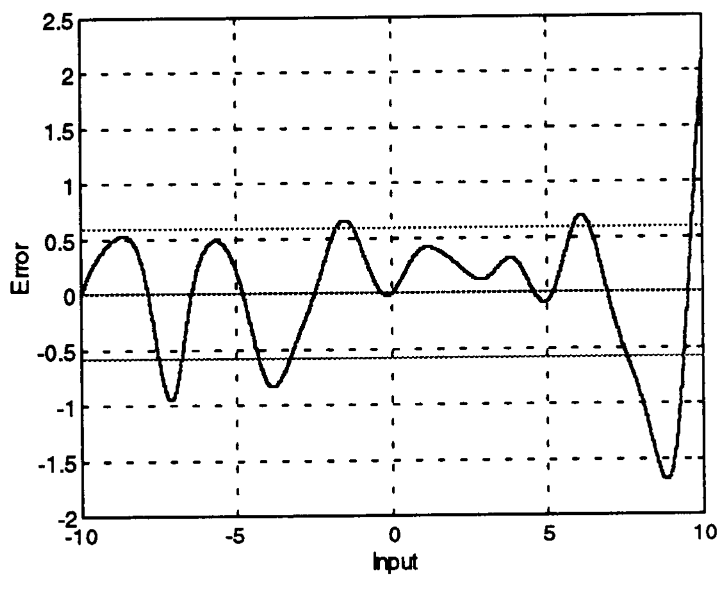
A design approach similar to the triangular one, similar results were obtained with a standard deviation of 0.5782 and an output which is depicted in Figure 4-34. At this point, no major advantages for using a cubic spline over a triangle in fuzzy rulebased system were encountered.

4.11.4. Piecewise-linear Response

Expecting worse results when compared with the corresponding triangular-shaped fuzzy system, it was pleasantly surprising to see a lower standard deviation of 0.7869 resulting from this effort. While providing a somewhat undulating output, the membership function boundary areas seem to be consistently wider than their triangular counterparts due to the rounded null points. This resulted in a better universe of discourse definition about the



(a)



(b)

Figure 4-34. Cubic Spline MBF Third-order Response

(a) actual results, (b) error graph

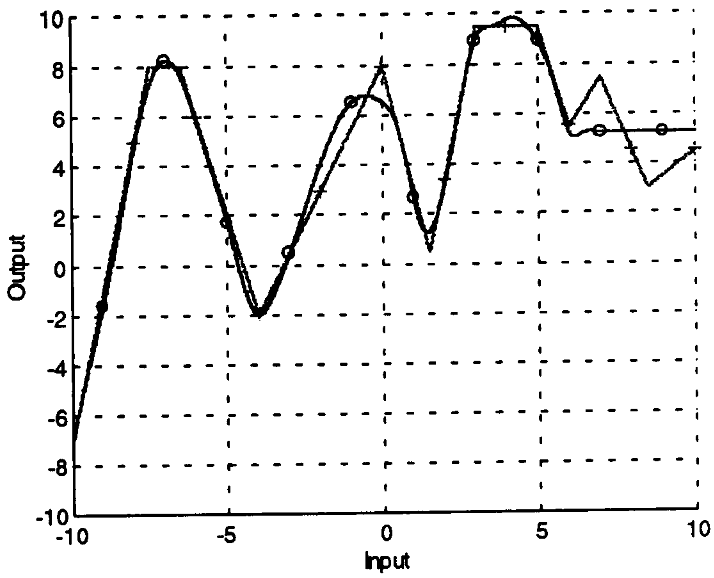
points of x = -7 and x = 4, where only one membership function defined fuzzy functioning with the triangular system. Therefore, these rounded areas proved useful in this case. The final results are depicted in Figure 4-35.

4.11.5. Sinusoidal Response

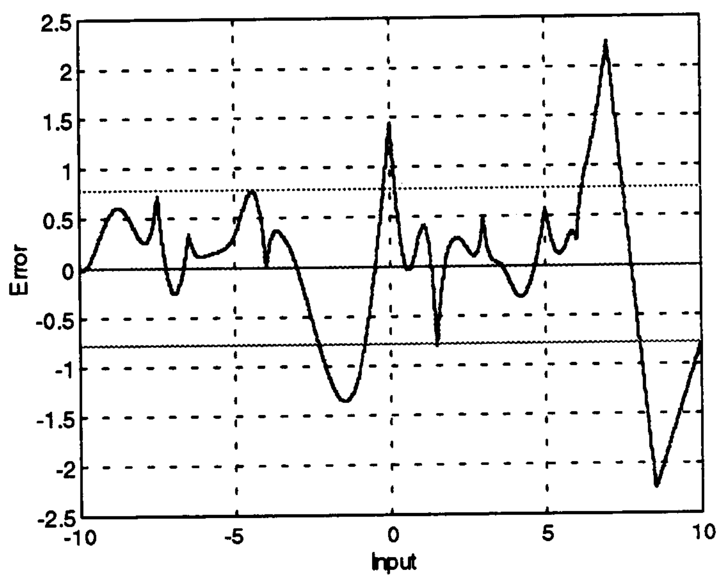
Rounded unity and null points proved their usefulness in mimicking a sinusoidal response. Not only were all areas defined by two or more membership functions, which was not the case for the triangular system, but the cubic spline shape was better suited to the rounded sinusoid minimum and maximum points. A minimum standard deviation of 1.6305 resulted with results accordingly displayed in Figure 4-36. For this response function, the cubic spline system produced excellent results.

4.12. Calculation of "Best" Membership Function

The shape with the lowest average standard deviation after use in all five response functions will be considered to be the "best" membership function. Results are collated and displayed in Table 4-3. With the this effort's given restrictions, the Gaussian shape proved to be the best membership function in recreating the desired response functions. Somewhat surprisingly, the trapezoidal shape came in second, with the cubic spline, triangle and sigmoidal shapes finishing third, fourth and last respectively.



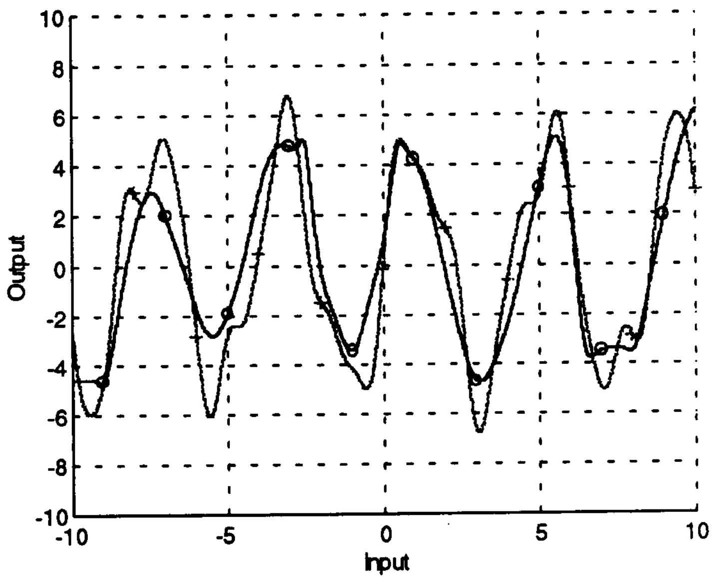
(a)



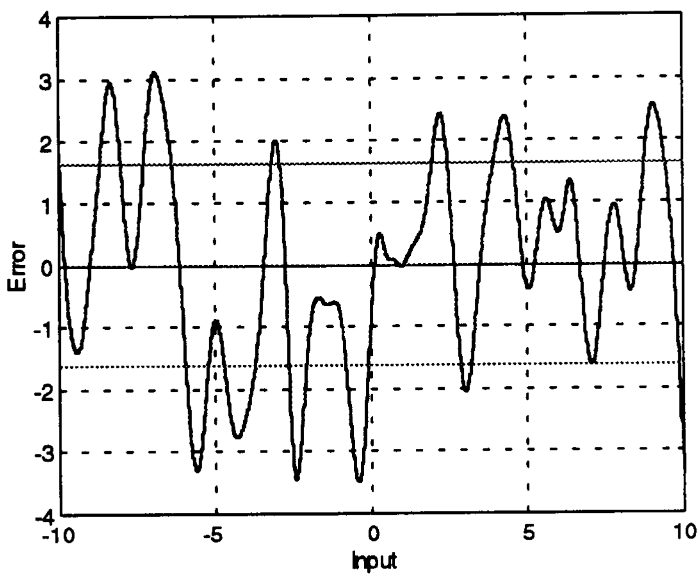
(b)

Figure 4-35. Cubic Spline MBF Piecewise-linear Response

(a) actual results, (b) error graph



(a)



(b)

Figure 4-36. Cubic Spline MBF Sinusoidal Response

(a) actual results, (b) error graph

Table 4-3. Standard Deviation Results

	Response Function					
MBF Shape	**First-order**	**Second-order**	**Third-order**	**Piecewise-linear**	**Sinusoidal**	**Ave.**
Triangle	0.0000	0.1225	0.5385	0.9217	1.8547	0.687
Trapezoid	0.0621	0.1485	0.5024	0.7045	1.4848	0.580
Gaussian	0.0701	0.1211	0.5168	0.6710	1.2389	0.524
Sigmoid	0.2129	0.3939	1.0769	1.1064	1.4345	0.845
Spline	0.0000	0.1312	0.5782	0.7869	1.6305	0.625

4.13. Analysis of Results

Summarized below are general opinions and recommendations regarding the five membership functions gleaned from this work. It needs to be understood, however, the restrictions placed on each membership function. In addition, actual use of these shapes in other fuzzy systems may vary due to different system requirements.

Although finishing somewhat lower on the average standard deviation scale, the triangular-shaped membership function provided several useful benefits. With its fast computation times and well-defined null points, this shape was quickly manipulated and performed an adequate job of replicating the desired response functions. It is easy to see why it is so popular and undoubtedly will be used in many fuzzy systems to come.

At first, the trapezoidal-shaped membership function was not expected to perform very well. However, as the response functions became more complex, it became obvious why it may be necessary to define a range of full membership values, rather than a single point. Almost as easy to work with as the triangle, the trapezoid proved versatile and very accurate in mimicking the desired response functions.

An exponentially-defined boundary region proved to be the strongest point with the Gaussian shape. As versatile as the trapezoidal membership function, this shape also had the distinct advantage of using the exponentially-defined boundary regions to not affect nearby membership functions nearly as much as the triangular and trapezoidal shapes. It should prove to do well in the subsequent fuzzy car model and needs to be a shape that is seriously considered for any fuzzy control system.

Frustration is the word that sums up the experience with the sigmoidal shape. There were really no cases encountered where the sigmoidal shape provided any significant advantage. The restriction of membership function normality was the reason for the disappointment. Subnormal sigmoidal shapes proved to look and act Gaussian in nature. However, this was not acceptable in the scope of this work. One very strong point can be seen for using this shape in other fuzzy systems. Using the sigmoid shape as just an S-curve, it would perform an excellent job defining the lower and upper boundaries of a Gaussian fuzzy system, as both shapes are defined by exponential terms. But sigmoid use for the entire universe of discourse can not be recommended.

The cubic spline, while rounded at its null and unity points, did not give enough of an advantage to replace using the triangular shape in a fuzzy system. The average standard deviation was slightly lower, but it was harder to work with and the rounded effects of the null points required special attention. However, there is a definite use for cubic spline membership functions. A spline shape would be ideal if one were to create membership functions by curve-fitting a set of data points.

5. APPLICATION TO CAR MODEL

With the best membership function shape determined to be Gaussian, it will be very interesting to apply this shape to a fuzzy-controlled vehicular model, thus providing data to enforce, or refute, the theoretical calculations performed so far. It is very important to note that the intent of this exercise is not to improve the fuzzy-controlled model. Rather, it is to determine how much an alternative shape can affect the overall simulation results. As the existing car model employed non-symmetrical membership functions, the Gaussian shape will also be non-symmetrical. In addition, as the Gaussian shape provides no distinct null points, the triangular shape definitions will be adhered to as closely as possible.

5.1. Introduction to Car Model

Developed by Brown [7], a nonlinear, three degree-of-freedom (3-DOF) front and rear independent steering (FRIS) vehicular model was developed. The intent of the effort was to determine if front-wheel and rear-wheel fuzzy controllers could be used to minimize yaw error (spin) on an icy surface. Specifically, four scenarios were determined to produce the most severe situations. These scenarios are repeated below in Table 5-1.

Table 5-1. Vehicular Simulation Scenario Descriptions

Scenario	Wheels	Disturbance
1	front	t = 1 sec. front wheel slip ratio begins to linearly increase (acceleration). t = 2 sec. front wheel slip ratio peaks at 0.95 and begins to linearly decrease. t = 3 sec. front wheels regain full traction.
	rear	t = 1 sec. rear wheel slip ratio begins to linearly increase (acceleration). t = 2 sec. rear wheel slip ratio peaks at 0.95 and begins to linearly decrease. t = 3 sec. rear wheels regain full traction.
2	front	t = 1 sec. front wheel slip ratio begins to linearly decrease (skidding). t = 2 sec. front wheel slip ratio peaks at -0.95 and begins to linearly increase. t = 3 sec. front wheels regain full traction.
	rear	t = 1 sec. rear wheel slip ratio begins to linearly decrease (skidding). t = 2 sec. rear wheel slip ratio peaks at -0.95 and begins to linearly increase. t = 3 sec. rear wheels regain full traction.
3	front	t = 1 sec. front wheel slip ratio begins to linearly decrease (skidding). t = 2 sec. front wheel slip ratio peaks at -0.30 and begins to linearly increase. t = 3 sec. front wheels regain full traction.
	rear	t = 1 sec. rear wheel slip ratio begins to linearly decrease (skidding). t = 2 sec. rear wheel slip ratio peaks at -0.30 and begins to linearly increase. t = 3 sec. rear wheels regain full traction.
4	front	t = 1 sec. front wheel slip ratio begins to linearly decrease (skidding). t = 2 sec. front wheel slip ratio peaks at -0.95 and begins to linearly increase. t = 3 sec. front wheels regain full traction.
	rear	No disturbance on rear wheels (slip ratio = 0).

The vehicular model included lateral, longitudinal and yaw motions about the center of mass. Nominal simulation parameters were chosen and input disturbances were generated by varying front and rear wheel slip ratios over time. The fuzzy controllers will simultaneously steer both the front and rear wheels, while monitoring wheel traction and requested steering wheel angles in order to achieve the driver's desired vehicle path. In each scenario, it was assumed that the vehicle was turning in a clockwise rotation.

5.2. Front and Rear Fuzzy Logic Controllers

Asymmetrical Gaussian membership function shapes simply replaced the existing fuzzy models' input triangular shapes. Both controllers were two-input, one-output systems with yaw error and corresponding wheel-slip ratios as inputs. The fuzzy controller outputs were the front and rear-wheel steering compensations respectively. Membership function descriptions may be found in Appendix D.

5.3. Simulation Results

As fuzzy-controlled, vehicular-model simulation packages were not readily available in the market at that time, Brown [7] created a custom fuzzy inference system. The system was modeled after the Mamdani inference approach and used the height method for defuzzification. The universe of discourse for the yaw error was from -5.0 to 5.0 radians per second. In addition, -1.0 to 1.0 defined the front-slip and rear-slip ratios' universe of discourse. As with this effort, all membership function definitions and placements were intuitively defined.

Two error figures are used for comparison purposes. As the desired yaw error equals zero, defined by the x-axis, any non-zero x-value is considered error. This provides a mean error analysis Also, the standard deviation of each result is an insightful value as it defines the "spread" of output values and provides another direct comparison measure.

With the Sugeno method utilized in this effort, exact results could not be attained. However, Sugeno "product" inferencing provided closer results than "minimum" inferencing. Detailed in Table 5-2, are the mean and standard deviation simulation differences for the four scenarios.

Table 5-2. Vehicular Model Simulation Deltas

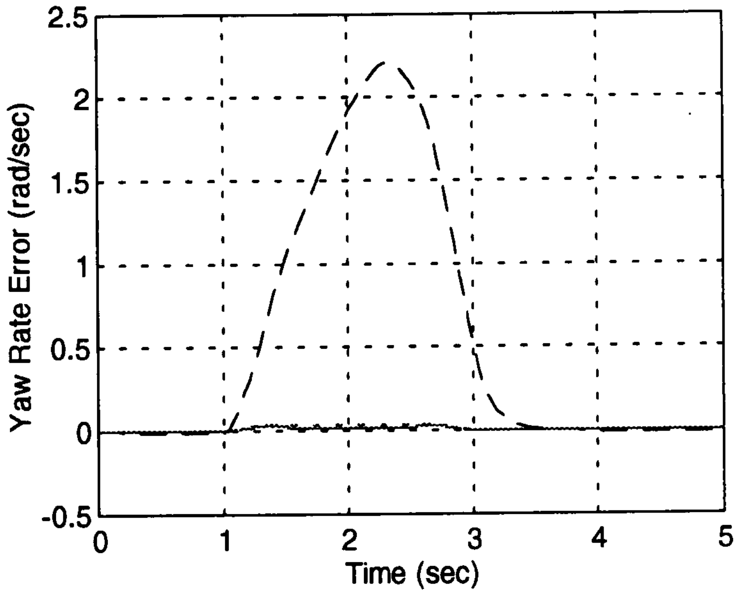
Scenario	Simulation Results	Baseline Model	Thesis Model	Percentage Improvement
1	mean	0.0132	0.0132	0.000%
	std dev	0.0161	0.0159	1.242%
2	mean	-0.0227	-0.0197	13.216%
	std dev	0.0597	0.0570	4.523%
3	mean	-0.0395	-0.0378	4.304%
	std dev	0.1150	0.1141	0.783%
4	mean	-0.0147	-0.0147	0.00%
	std dev	0.0259	0.0256	1.158%

These percentages will be removed from below calculations in order to achieve a direct comparison between the two membership function shapes.

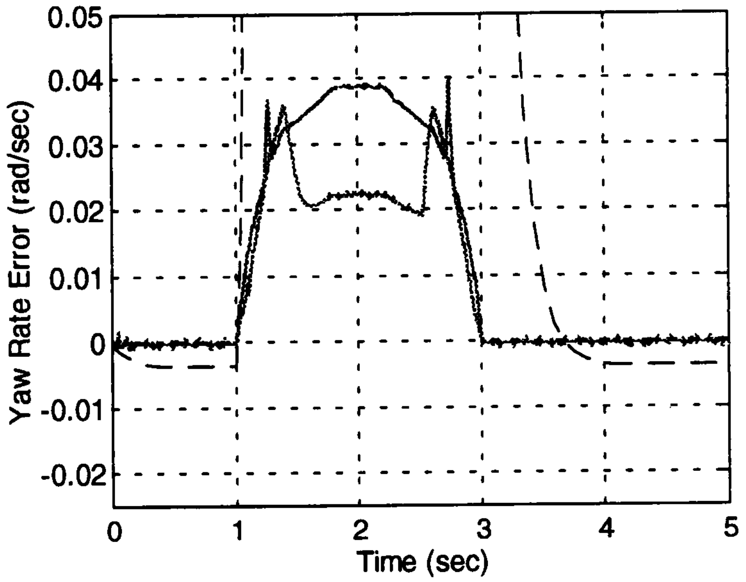
Figure 5-1 through Figure 5-4 depict all four simulations scenarios described above. The long-dashed line displays the uncompensated vehicular model results and the short-dashed line depicts Brown's triangular-shaped membership function fuzzy-controlled vehicular results. The solid line shows this effort's Gaussian-shaped membership function results using the same vehicular model. Overall and expanded views are provided for each scenario.

It is very interesting to note that in all four cases, the choice of a Gaussian membership function shape had a direct, positive impact on system performance. Although this impact was slight for the first scenario, nevertheless, the Gaussian shape provided a 34.09% improvement in mean error and a 26.09% improvement in standard deviation. Both fuzzy-controlled results were very-much improved over the uncompensated model.

The second scenario provided the exact opposite input stimulus as the first. In this case, the vehicle went skidding into the spin, instead of accelerating into it. A dual-humped output response resulted at the times that the wheels both lost and regained traction. A 44.05% and 39.70% improvement in error figures resulted respectively.



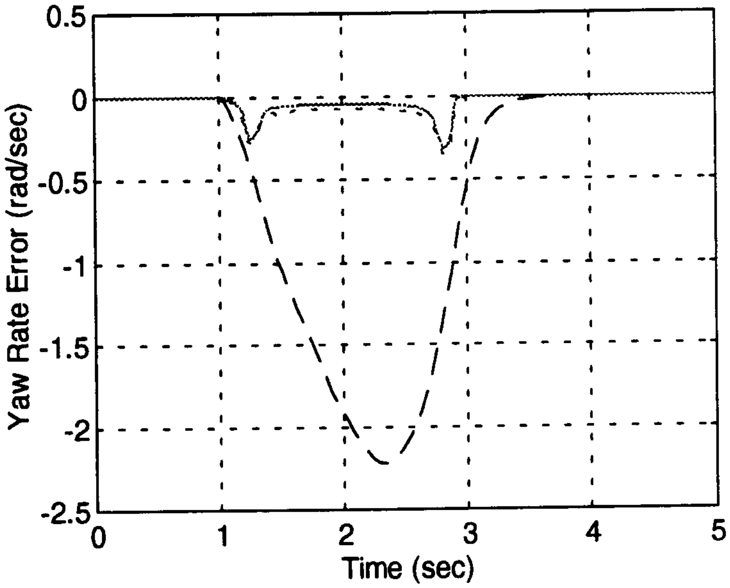
(a)



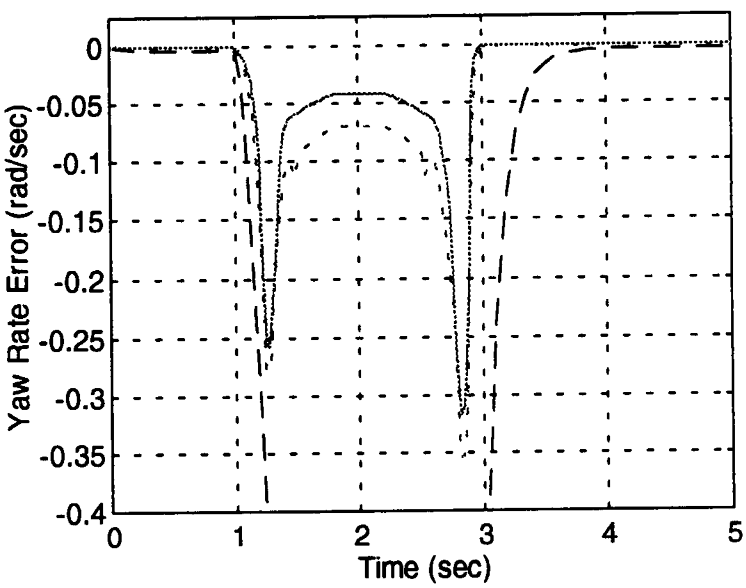
(b)

Figure 5-1. Scenario 1 Results

(a) scenario 1 results, (b) expanded view



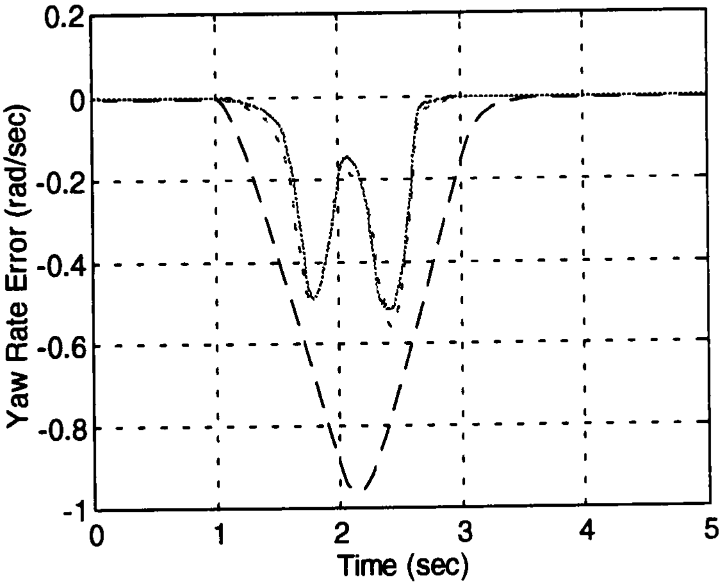
(a)



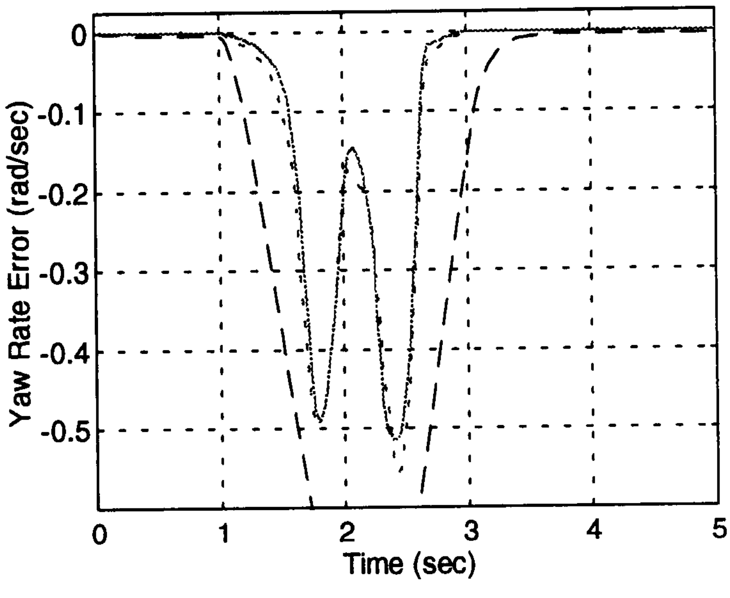
(b)

Figure 5-2. Scenario 2 Results

(a) scenario 2 results, (b) expanded view



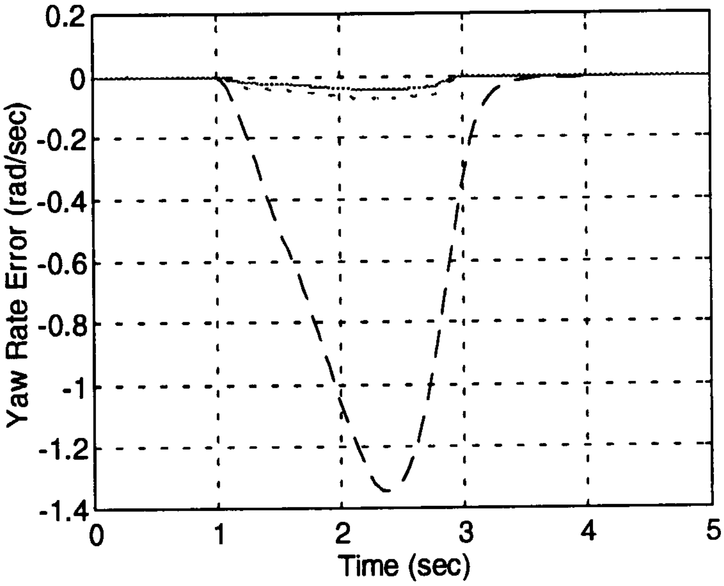
(a)



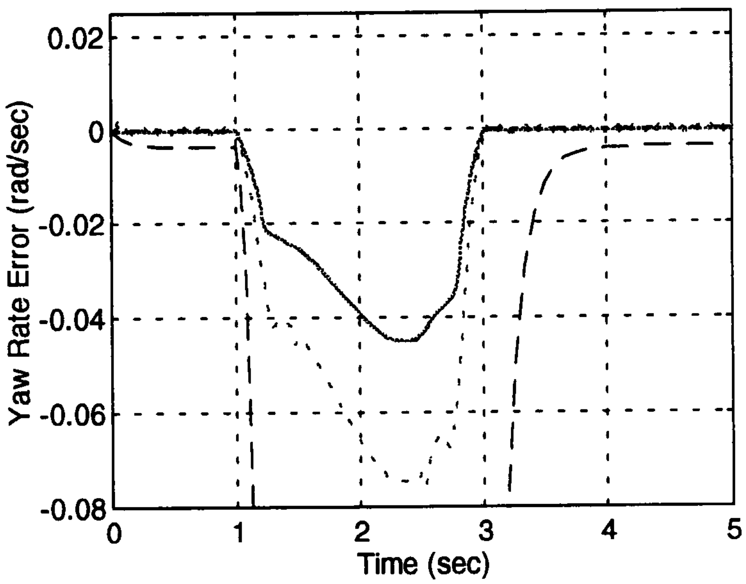
(b)

Figure 5-3. Scenario 3 Results

(a) scenario 3 results, (b) expanded view



(a)



(b)

Figure 5-4. Scenario 4 Results

(a) scenario 4 results, (b) expanded view

Scenario three, though defined by lower front and rear wheel slip ratios proved to be the most demanding on the fuzzy controllers. For this case, the front wheels need to be turned into the direction of skid, with the rear wheels turned away from the direction of skid to gain control of the car. Then, all wheels needed to be turned away from the direction of skid for the vehicle to return to zero yaw error. The large error points depicted on the graph result from these two occurrences. A mean error improvement of 57.22% and standard deviation improvement of 36.70% were recorded.

A large front-wheel slip ratio with no rear-wheel slip ratio defines the final scenario. In this case, the full traction afforded to the rear wheels allowed the rear-wheel fuzzy controller to effectively compensate for the loss of traction at the front wheels. Both triangular and Gaussian-defined fuzzy controllers performed well, with a 44.22% and 41.70% respective improvement using the Gaussian membership functions.

It is often useful to compile data into a table for comparison purposes. Table 5-3 below combines the data for all four scenarios and presents an average increase in system performance due to using the Gaussian membership function shape in place of the triangular shape. Note that the simulation deltas were removed from all percentage improvement figures. The final improvement figures are simply the averages of the corresponding percentage calculations.

Table 5-3. Vehicular Simulation Error Analysis

Scenario	Simulation Results	Triangular Model	Gaussian Model	Percentage Improvement
1	mean	0.0132	0.0087	34.09%
	std dev	0.0161	0.0117	26.09%
2	mean	-0.0227	-0.0097	44.05%
	std dev	0.0597	0.0333	39.70%
3	mean	-0.0395	-0.0152	57.22%
	std dev	0.1150	0.0719	36.70%
4	mean	-0.0147	-0.0082	44.22%
	std dev	0.0259	0.0148	41.70%
Average performance improvement				**44.9%**
				36.0%

5.4. Analysis of Results

It is amazing that a 44.9% improvement in mean error and 36.0% improvement in standard deviation could occur just by changing the membership function definition. Through experimentation, it became obvious that the major improvement of vehicular control ocurred when the yaw error membership functions were changed to Gaussian shapes. While changing the wheel slip ratio membership functions to Gaussian also improved system performance, its effect was much smaller. Though not an expert on vehicular control, it is hypothesized that since these are extreme situations, it appears that the exponential rise of the “nl” and “pl” yaw rate error input membership functions have the most impact on system performance improvement. Negative-large and positive-large steering compensations are therefore executed more aggressively than their triangular counterparts. As a final note, symmetrical Gaussian functions also improved system performance, but not to the degree of asymmetrical Gaussian shapes.

6. CONCLUSION

This study performed an analysis of five fuzzy logic membership function shapes by mimicking five desired response functions. Using a one-input, one-output system of the form $y = f(x)$, a fuzzy rule-based system was created to approximate $f(x)$. After driving each system's mean error to $\leq |0.01|$, the standard deviation of each system's output was recorded. The shape with the lowest average standard deviation was determined to be the "best" membership function shape given this effort's defined constraints.

The Gaussian shape proved to produce the lowest average standard deviation, with the trapezoidal, cubic spline, triangular and sigmoidal shapes finishing second through last respectively. Its exponential nature allowed the shape to rise quickly from null to full membership without adversely affecting adjacent membership functions. This shape was fairly easy to use and provided excellent results.

Application of this shape to a 3-DOF, FRIS vehicular model provided simulation data in which to compare this Gaussian function to the triangular standard. Replacing the triangular shape with the Gaussian shape provided an average of 44.9% and 36.0% improvement in mean error and standard deviation using four worst-case driving scenarios on an icy surface.

Continued studies are recommended to further increase the knowledge regarding membership function shapes and their affect on system performance. Using genetic algorithms to create membership function shapes, asymmetrical membership function shapes and using first-order Sugeno inference in replicating the five response functions are all matters worth pursuit.

REFERENCES

LIST OF REFERENCES

[1] Ribeiro, R.A., Powell, P.L., and Baldwin, J.F., "Uncertainty in decision-making: An abductive perspective," Decision Support Systems, Vol. 13, pp. 183-193, 1995.

[2] Zadeh, L.A., "Fuzzy sets," Information and Control, Vol. 8, No. 3, pp. 338-353, June 1965.

[3] Anderson, G., "Applying fuzzy logic in the real world," Sensors, pp. 15-25, September 1992.

[4] Maiers, J. and Sherif, Y.S., "Applications of fuzzy set theory," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 1, pp. 175-189, 1985.

[5] Foslien, W.K., Ray, J. and Wood, W.T., "An introduction to fuzzy control," Proceedings of 42nd International Appliance Technical Conference, pp. 251-263, May 1991.

[6] Boston, J.R., "Effects of the shape of fuzzy membership functions on fuzzy inference," IEEE Proceedings of ISUMA-NAFIPS, pp. 32-37, 1995.

[7] Brown, G.L., "Vehicle steering assistance in extreme situations using a fuzzy logic controller," Thesis, The University of Tennessee, pp. 1-104, 1994.

[8] Brubaker, D.I., "Fuzzy-logic basics: intuitive rules replace complex math," Engineering Design News, pp. 111-116, June 18, 1992.

[9] Kantrowitz, M., Horstkotte, E. and Joslyn, C., "Answers to Frequently Asked Questions about Fuzzy Logic and Fuzzy Expert Systems," comp.ai.fuzzy, pp. 5-7, March 1996.

[10] Anderson, G., "Fundamentals of fuzzy logic: Part I," Sensors, pp. 40-49, March 1993.

[11] Mamdani, E.H., and Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, pp. 1-13, 1975.

[12] Ross, T.J., Fuzzy Logic with Engineering Applications, McGraw-Hill, Inc., International Edition, pp. 134-137, 1995.

[13] Bartels, R.H., Beatty, J.C., and Barsky, B.A., An Introduction to Splines for use in Computer Graphics and Geometric Modeling, Morgan Kaufman Publishers, Inc., Los Altos, California, pp. vii-x, 1987.

[14] Greville, T.N., Editor, Theory and Applications of Spline Functions, Proceedings of an advanced seminar conducted by the Mathematics Research Center, United States Army, at the University of Wisconsin, Madison, Academic Press, New York, New York, pp. 1-35, 1969.

[15] Burden, R.L., and Faires, J.D., Numerical Analysis, PWS Publishing Company, Boston, Massachusetts, pp. 130-148, 1993.

[16] Sugeno, M., Editor, Industrial Applications of Fuzzy Control, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, pp. 125-138, 1985.

[17] Stark, H. and Woods, J., Probability, Random Processes, and Estimation Theory for Engineers, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 111-113, 1986.

[18] Beyer, W.H., Editor, CRC Standard Mathematical Tables - 28th Edition, CRC Press, Boston, Massachusetts, pp. 519, 1987.

APPENDICES

APPENDIX A

ACRONYMS AND ABBREVIATIONS

APPENDIX A

3-DOF	three degree of freedom
FFT	fast Fourier transform
FLC	fuzzy logic controller
FRIS	front and rear independent steering
ft/s	feet per second
ICC	intelligent cruise control
max	maximum
MBF	membership function
min	minimum
mph	miles per hour
SM	strictly monotonically

APPENDIX B

LIST OF SYMBOLS

APPENDIX B

$\in$	member of a set
$\notin$	non-member of a set
j	imaginary portion of a variable
s	Laplace variable ($s=\sigma+j\omega$)
σ	constant used to ensure Laplace integral convergence
U	universe of discourse
ω	radian frequency (radians per second)
$\cap$	intersection
$\cup$	union
-	logical complement: Not A = $\bar{A}$

APPENDIX C

FUZZY LOGIC DEFINITIONS

APPENDIX C

Aggregation - The combination of the consequents of each rule in a Mamdani fuzzy inference system for defuzzification purposes, or the combination of output values in a Sugeno inference system.

Antecedent - The "if" part of a fuzzy rule.

Boolean logic - Named after George Boole (1815-1864). An algebraic combinatorial system treating variables through operators AND, OR, NOT, IF, THEN, and EXCEPT. A variable in Boolean logic takes the value of true or false, but not both.

Center of gravity - Also known as the centroid method. Most commonly adopted defuzzification method which produces the expected value for a consequent variable by calculating the center of gravity (or its first moment of inertia) of the consequent fuzzy region.

Consequent - The "then" part of a fuzzy rule.

Crisp input - Distinct and sharply divided inputs.

Defuzzification - A process in which the expected value of an output variable is derived by isolating a crisp value in the universe of discourse of the output fuzzy sets.

Degree of membership - The amount to which a crisp value is compatible with a membership function (from 0 to 1). Also referred to as membership grade, truth value or fuzzy input.

Fire - A non-zero membership grade within the antecedent of a fuzzy logic IF-THEN rule.

Firing Strength - The degree to which the antecedent part of a fuzzy rule is satisfied.

Fuzzification - Domain transformation for an input from a crisp value to a fuzzy value.

Fuzzy output - The value assigned to each output membership function label after evaluating the rule strengths of all rules specifying that consequent action.

Fuzzy set - A set with fuzzy boundaries. A fuzzy set is represented in a fuzzy rulebased system with a membership function.

Label - A descriptive name used to identify a membership function.

Membership function - Defines a fuzzy set by mapping crisp inputs from its domain to degrees of membership.

Rule evaluation - Also called fuzzy inference. A technique used to calculate a numerical value representing the truth for a certain consequent action based on a set of rules bearing the consequent.

Scope/domain - Width of the membership function. The range of concepts, usually numbers, over which a membership function is mapped (i.e., the domain is from 60-80. The scope is 20).

Singleton - A special type of fuzzy set that contains only one element in the set.

Universe of discourse - Range of all possible values applicable to a system variable.

APPENDIX D

INPUT AND OUTPUT MEMBERSHIP FUNCTIONS

APPENDIX D

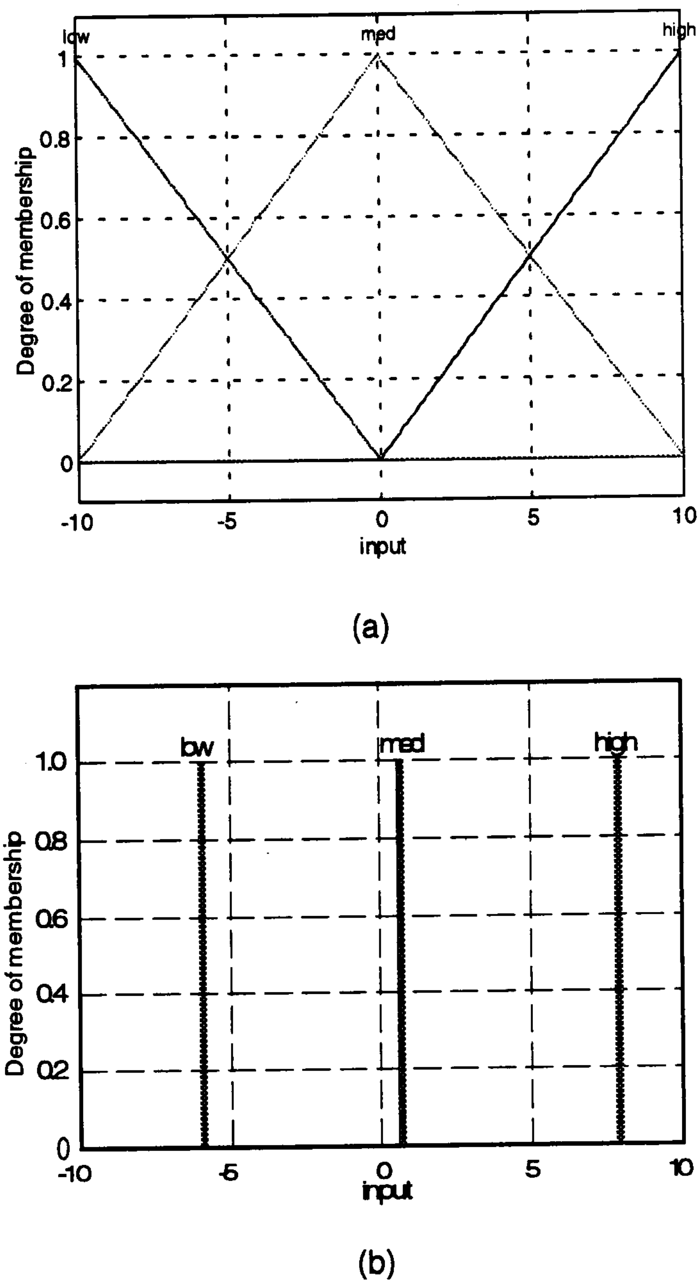


Figure D-1. Triangular MBF First-order Definition

(a) input membership functions, (b) output membership functions

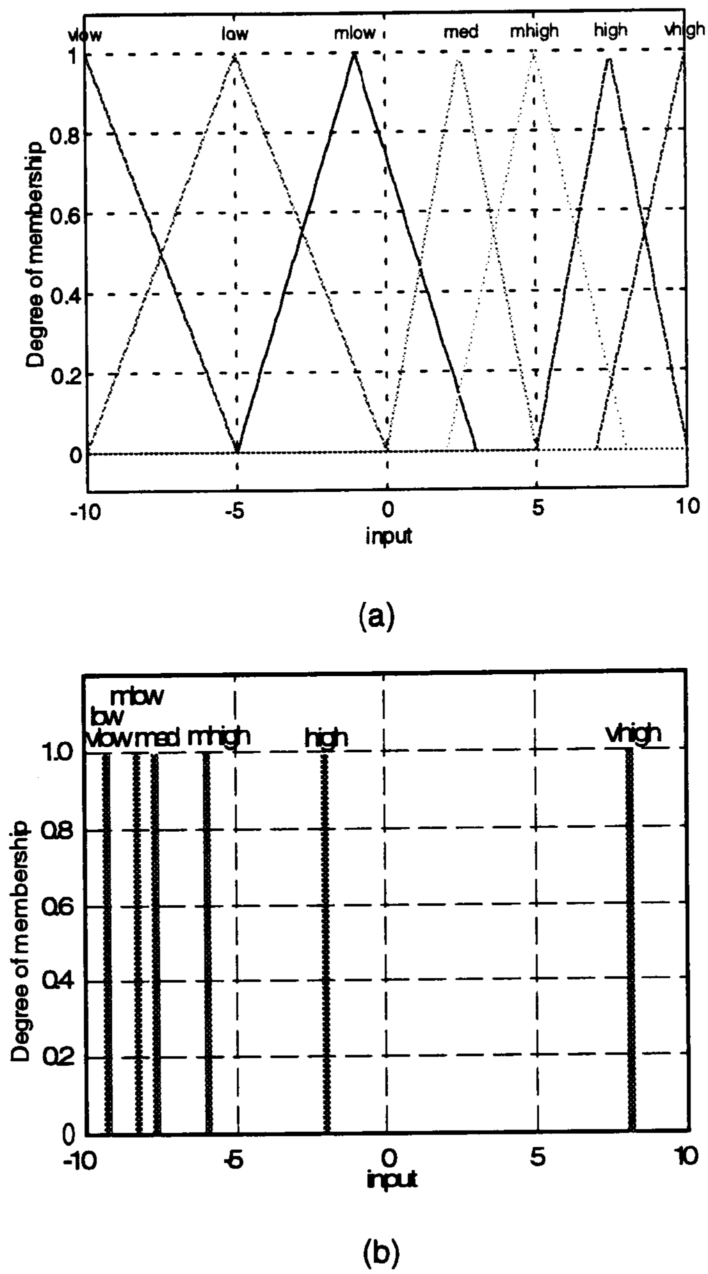
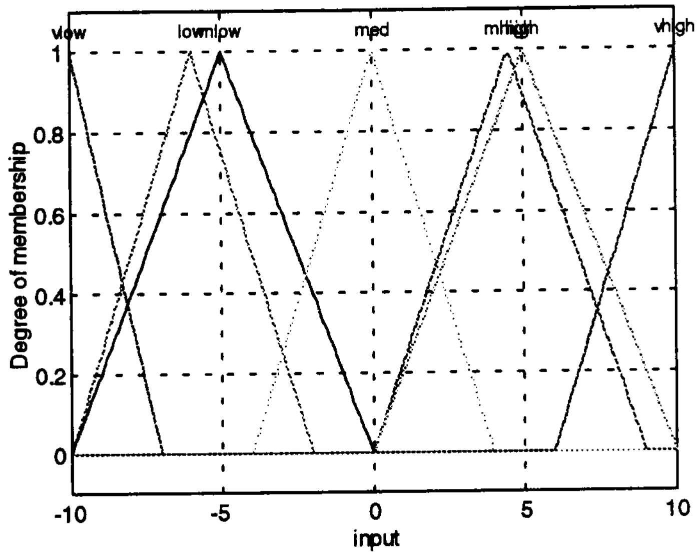
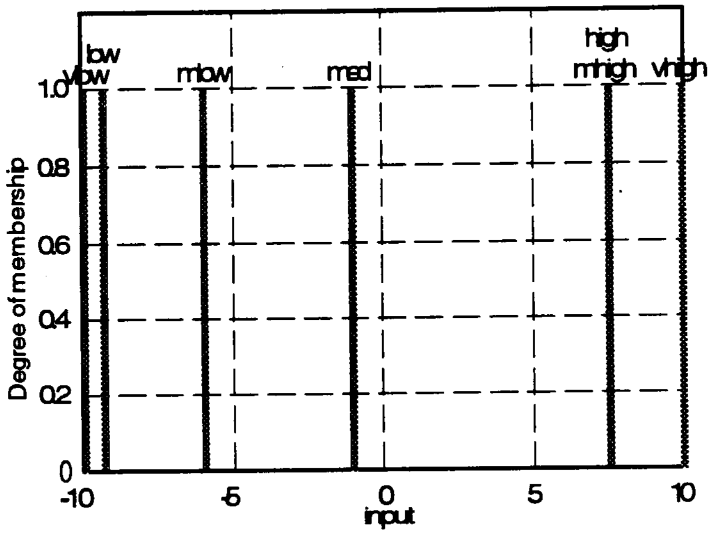


Figure D-2. Triangular MBF Second-order Definition

(a) input membership functions, (b) output membership functions



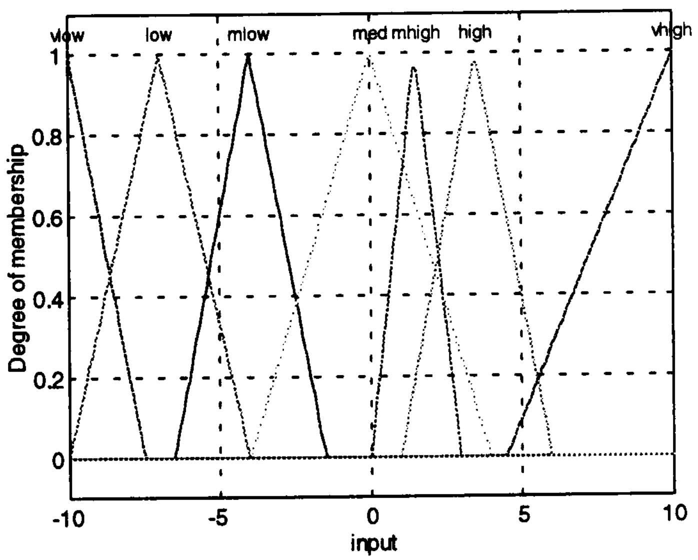
(a)



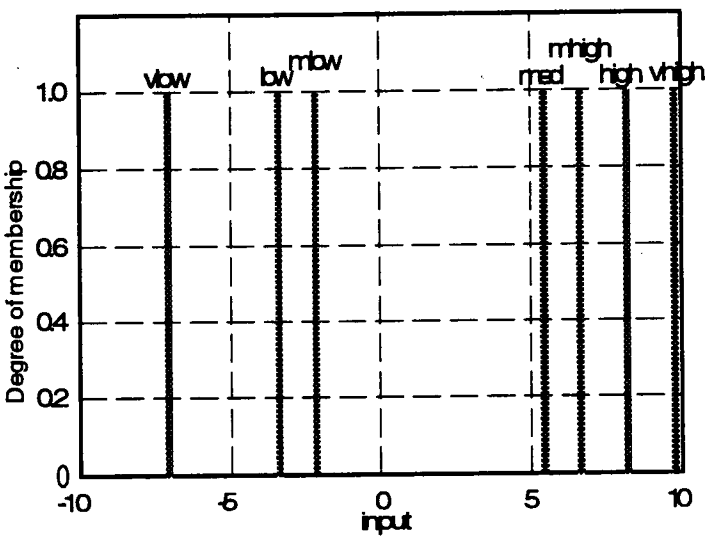
(b)

Figure D-3. Triangular MBF Third-order Definition

(a) input membership functions, (b) output membership functions



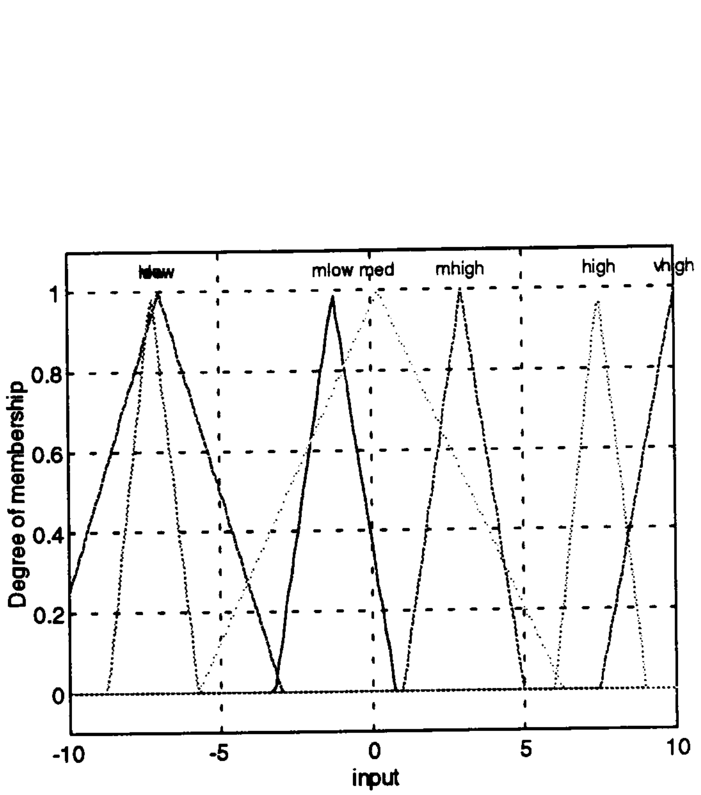
(a)



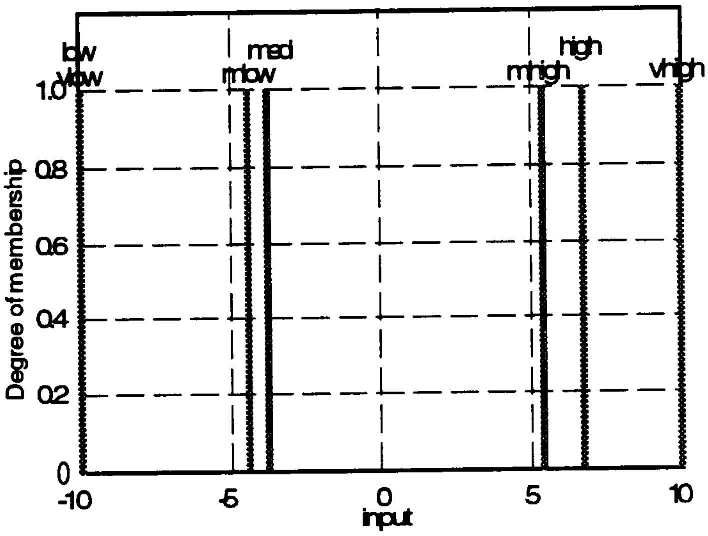
(b)

Figure D-4. Triangular MBF Piecewise-linear Definition

(a) input membership functions, (b) output membership functions



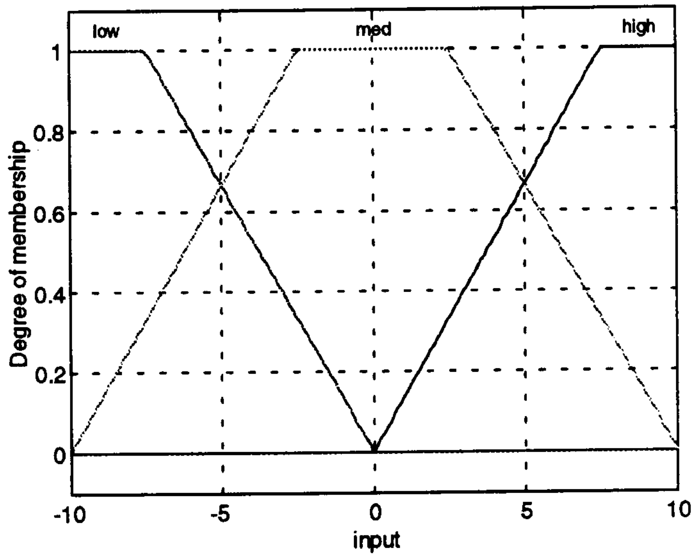
(a)



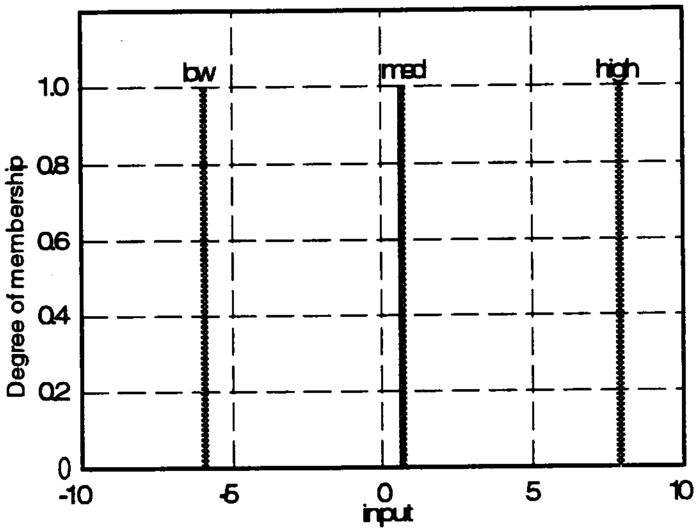
(b)

Figure D-5. Triangular MBF Sinusoidal Definition

(a) input membership functions, (b) output membership functions



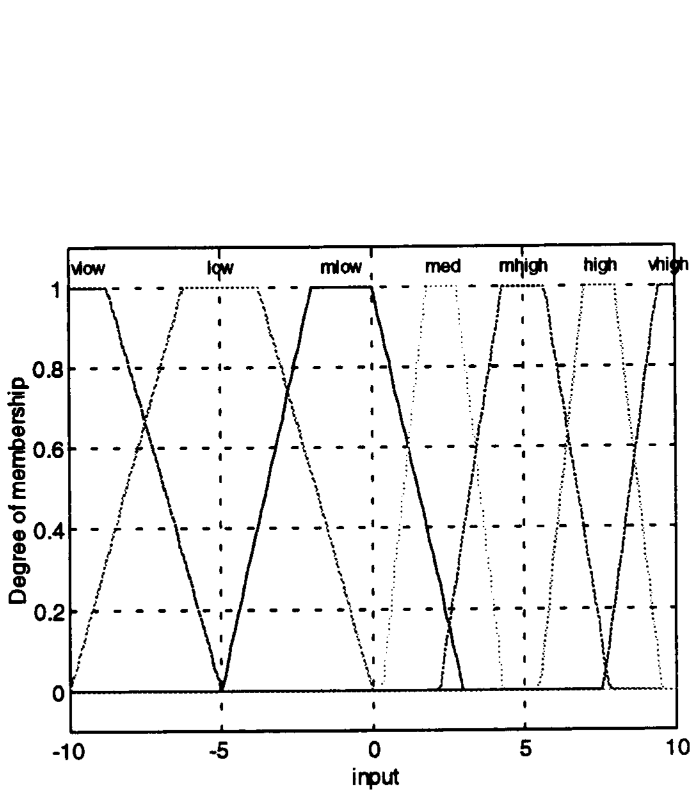
(a)



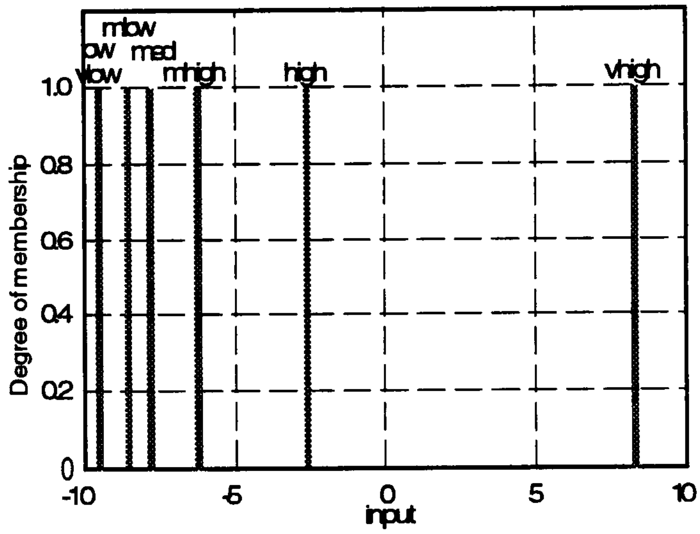
(b)

Figure D-6. Trapezoidal MBF First-order Definition

(a) input membership functions, (b) output membership functions



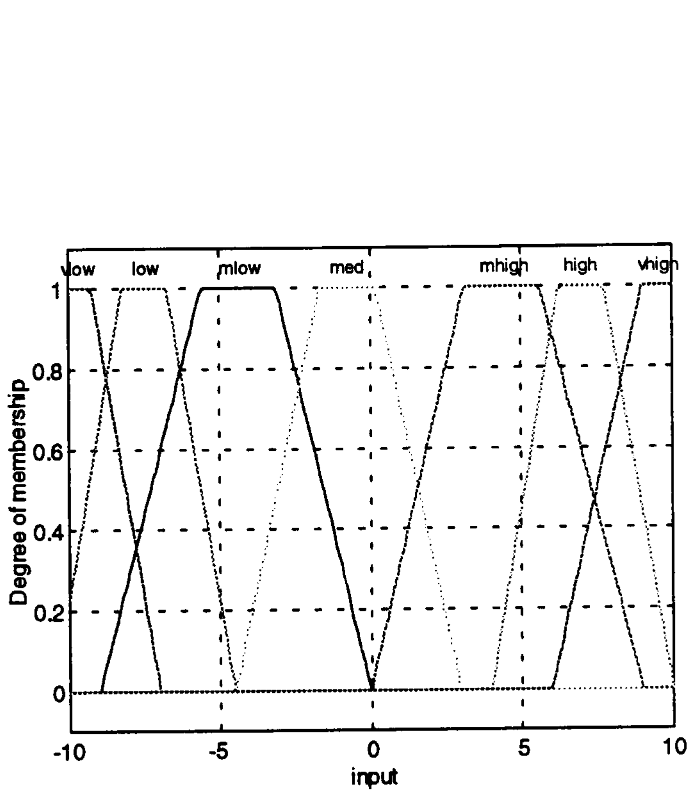
(a)



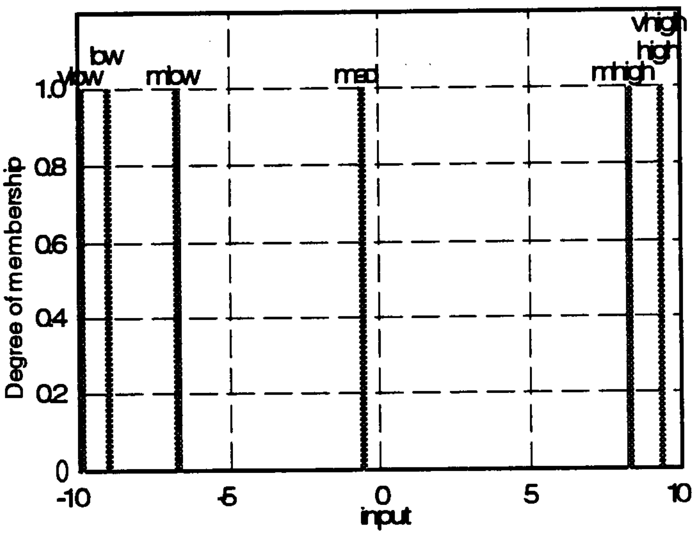
(b)

Figure D-7. Trapezoidal MBF Second-order Definition

(a) input membership functions, (b) output membership functions



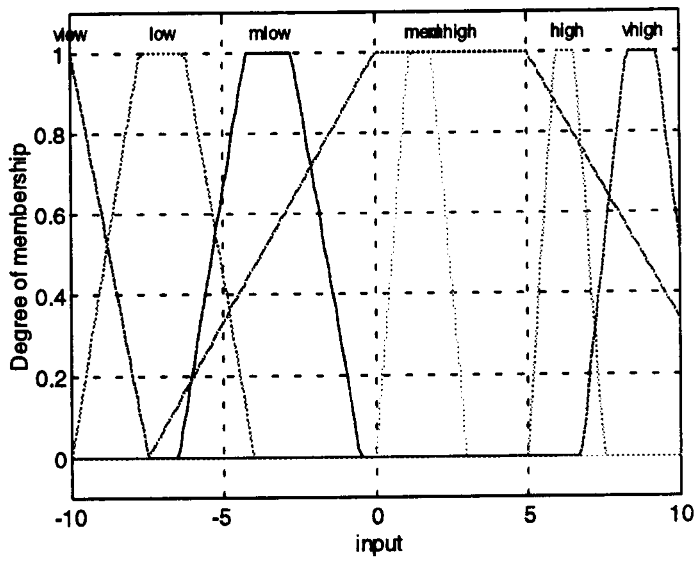
(a)



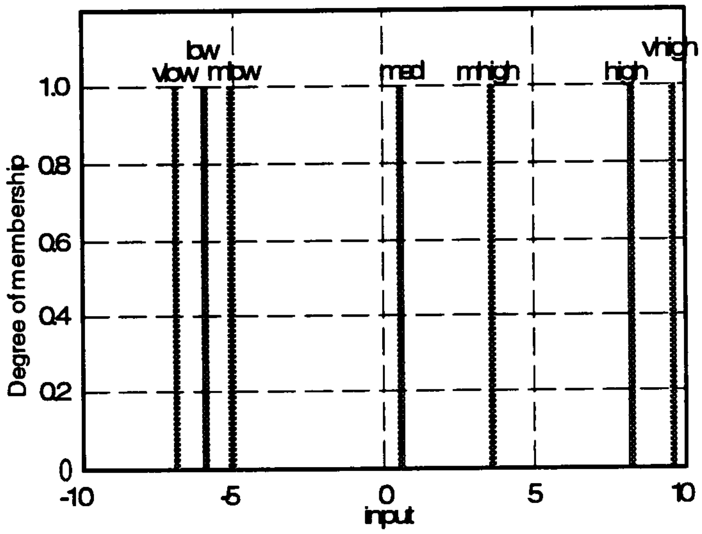
(b)

Figure D-8. Trapezoidal MBF Third-order Definition

(a) input membership functions, (b) output membership functions



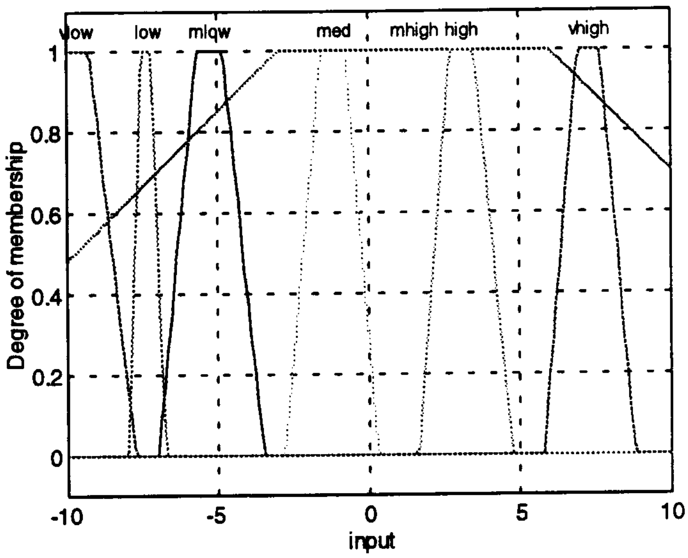
(a)



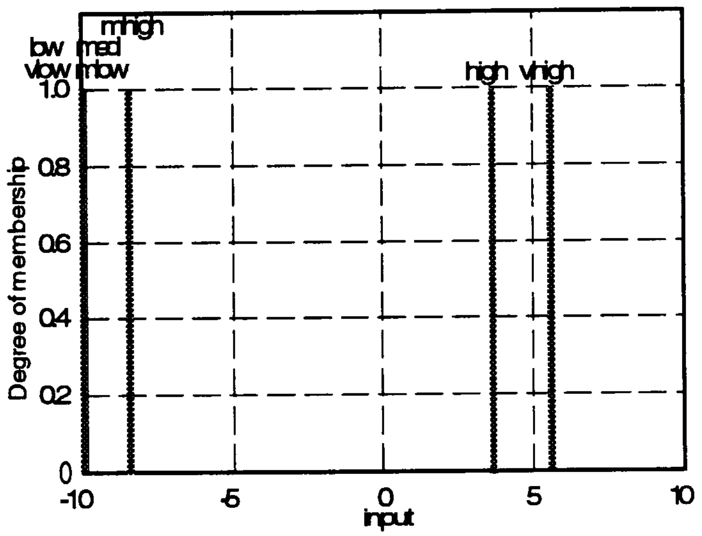
(b)

Figure D-9. Trapezoidal MBF Piecewise-linear Definition

(a) input membership functions, (b) output membership functions



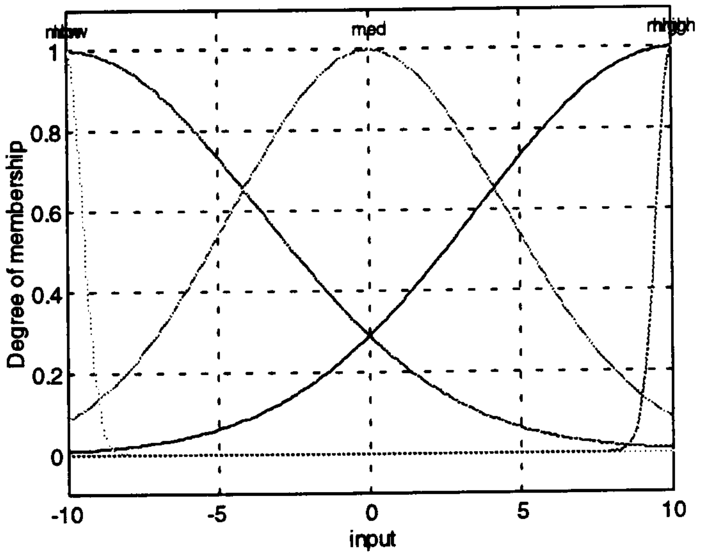
(a)



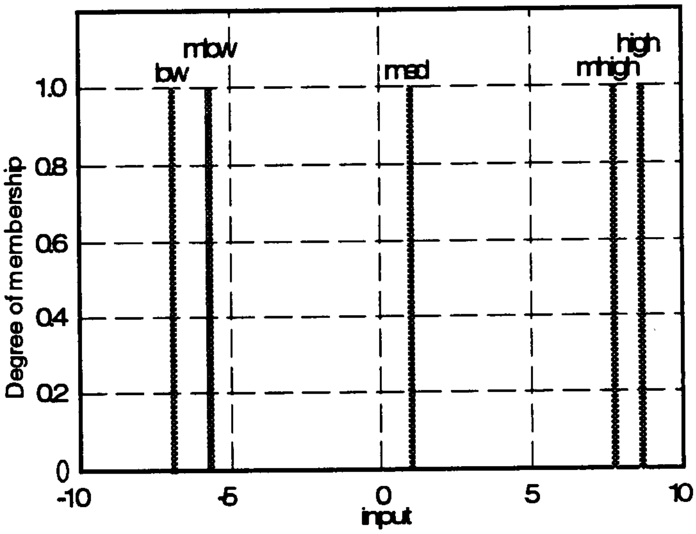
(b)

Figure D-10. Trapezoidal MBF Sinusoidal Definition

(a) input membership functions, (b) output membership functions



(a)



(b)

Figure D-11. Gaussian MBF First-order Definition

(a) input membership functions, (b) output membership functions

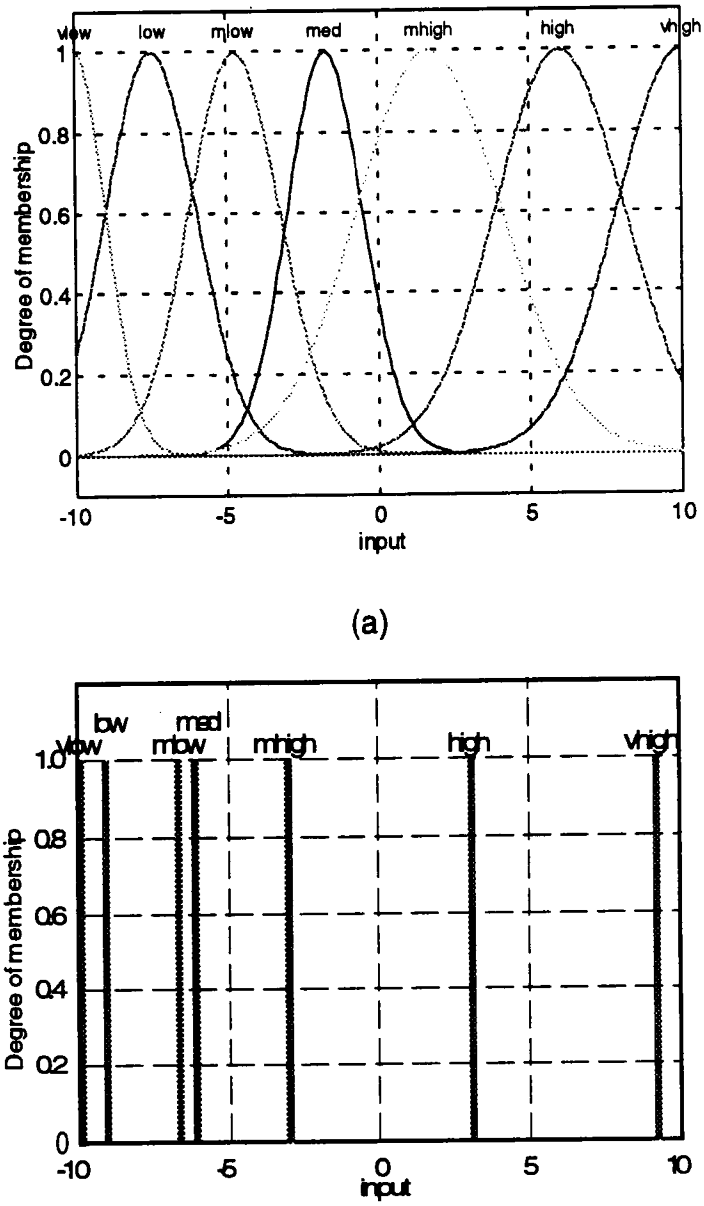
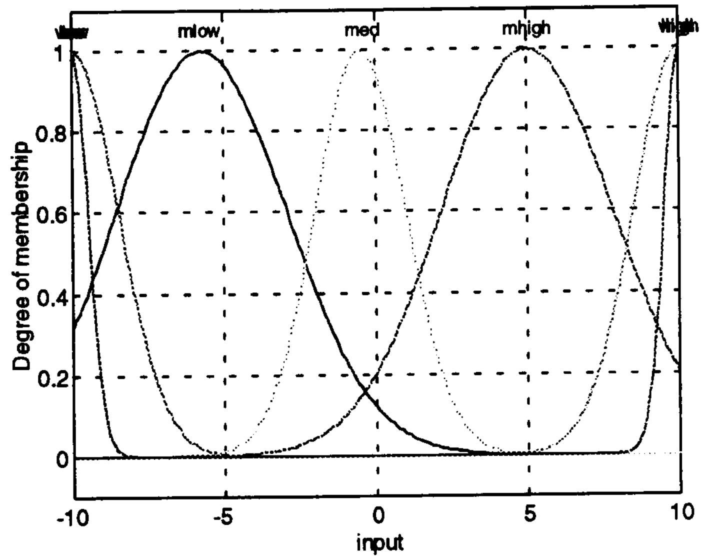
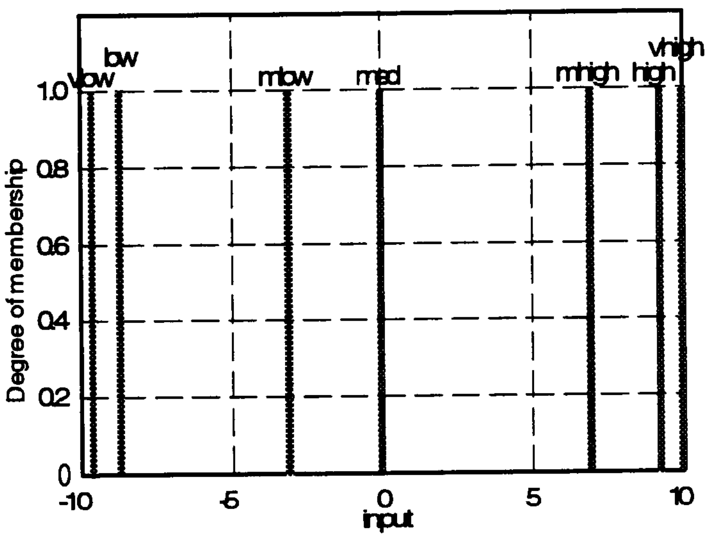


Figure D-12. Gaussian MBF Second-order Definition

(a) input membership functions, (b) output membership functions



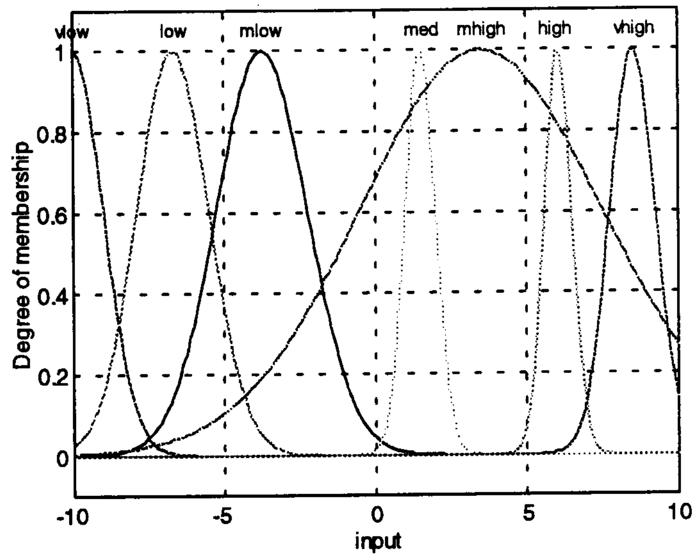
(a)



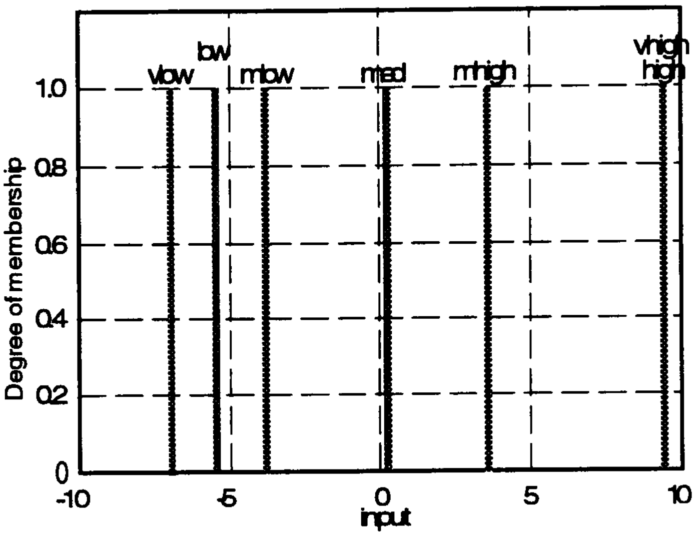
(b)

Figure D-13. Gaussian MBF Third-order Definition

(a) input membership functions, (b) output membership functions



(a)



(b)

Figure D-14. Gaussian MBF Piecewise-linear Definition

(a) input membership functions, (b) output membership functions

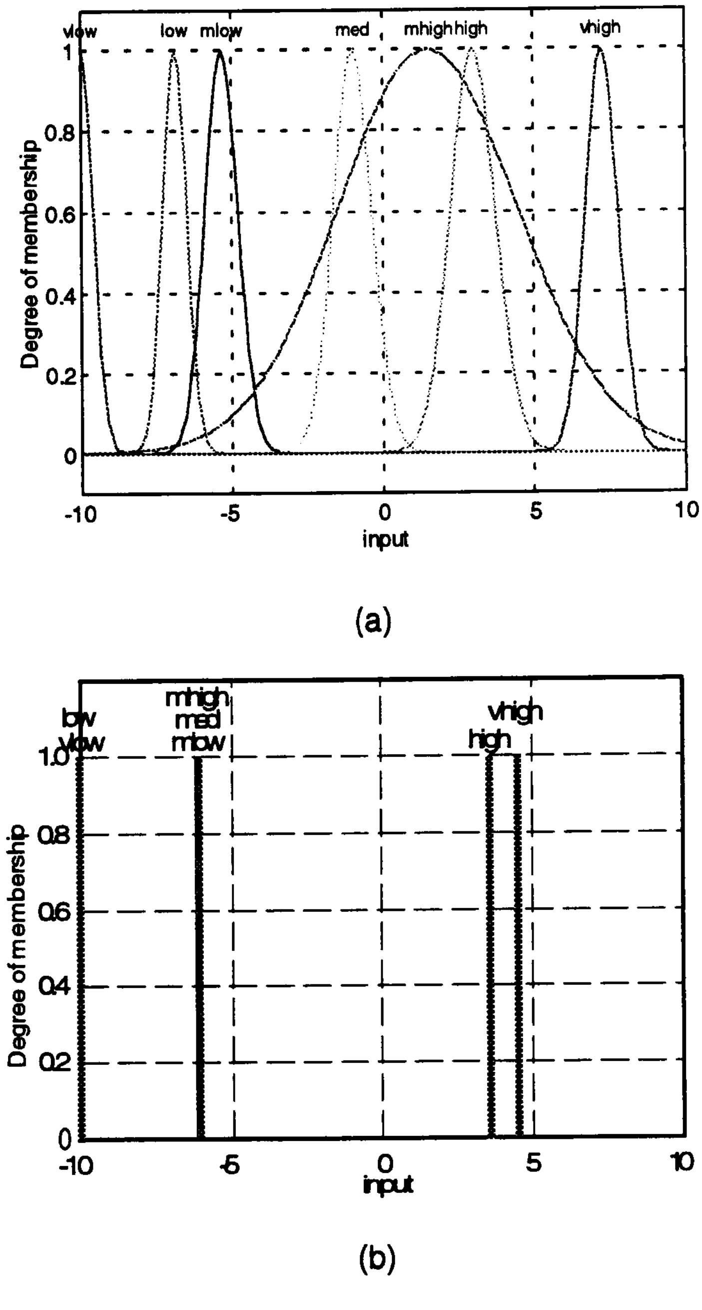
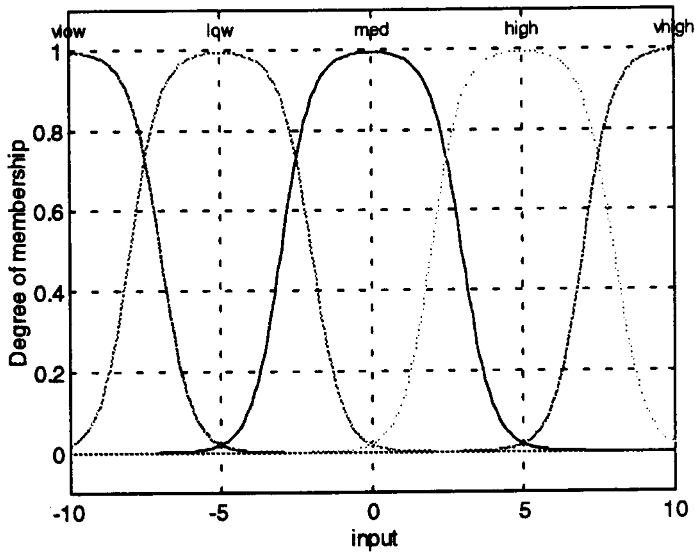
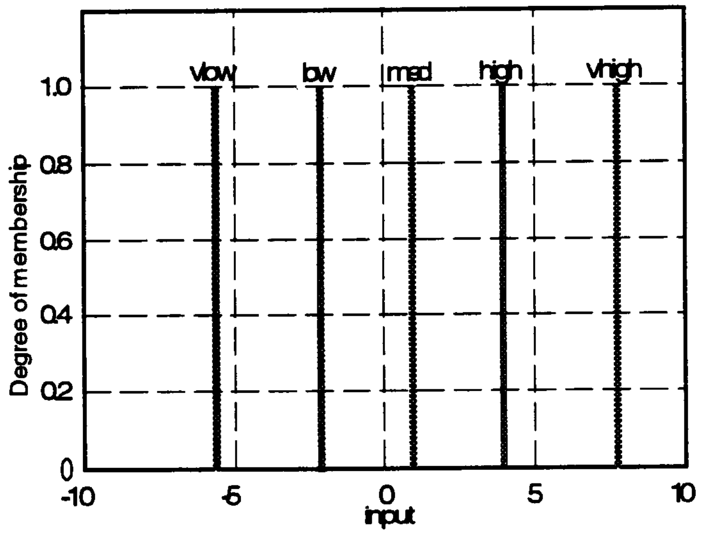


Figure D-15. Gaussian MBF Sinusoidal Definition

(a) input membership functions, (b) output membership functions



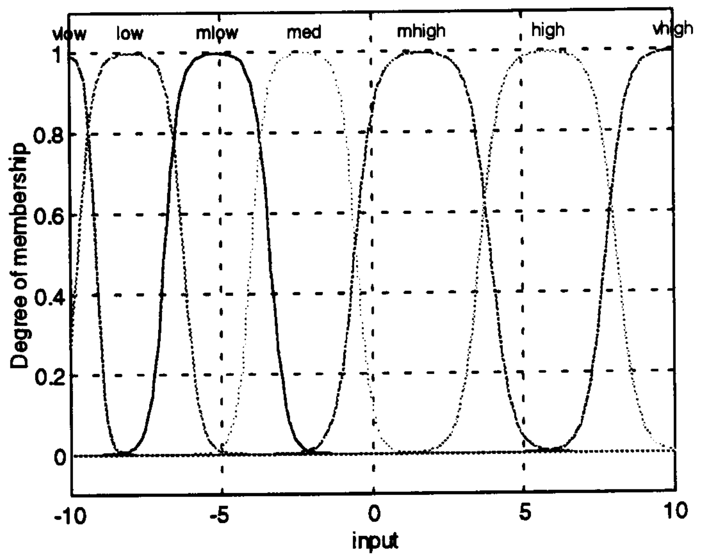
(a)



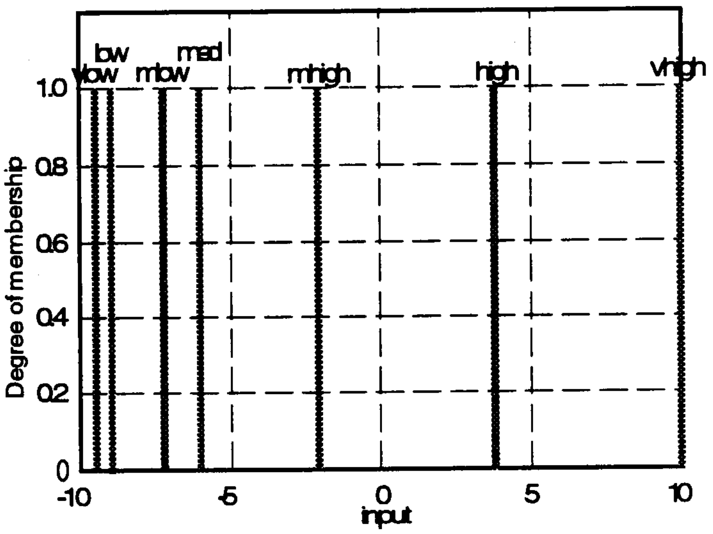
(b)

Figure D-16. Sigmoidal MBF First-order Definition

(a) input membership functions, (b) output membership functions



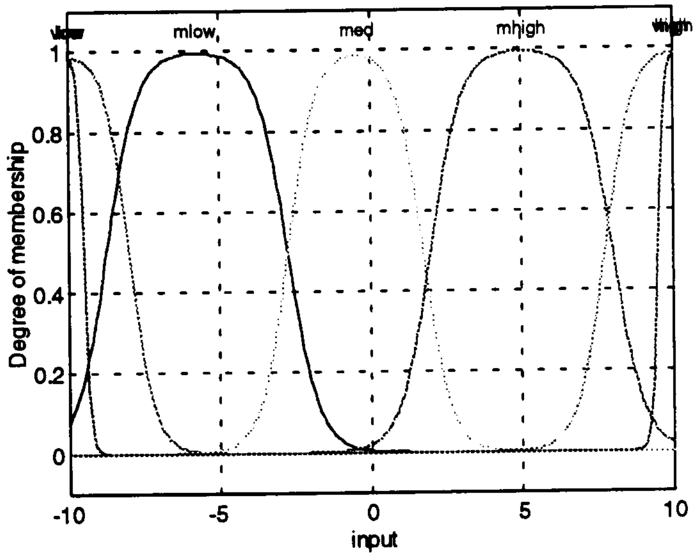
(a)



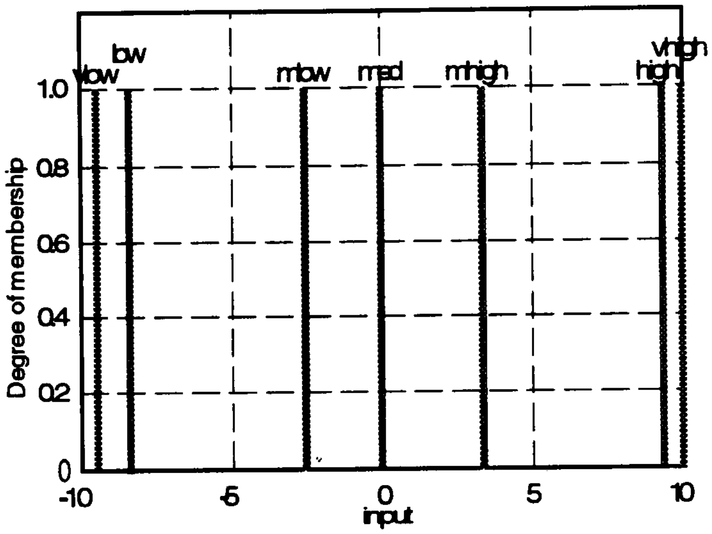
(b)

Figure D-17. Sigmoidal MBF Second-order Definition

(a) input membership functions, (b) output membership functions



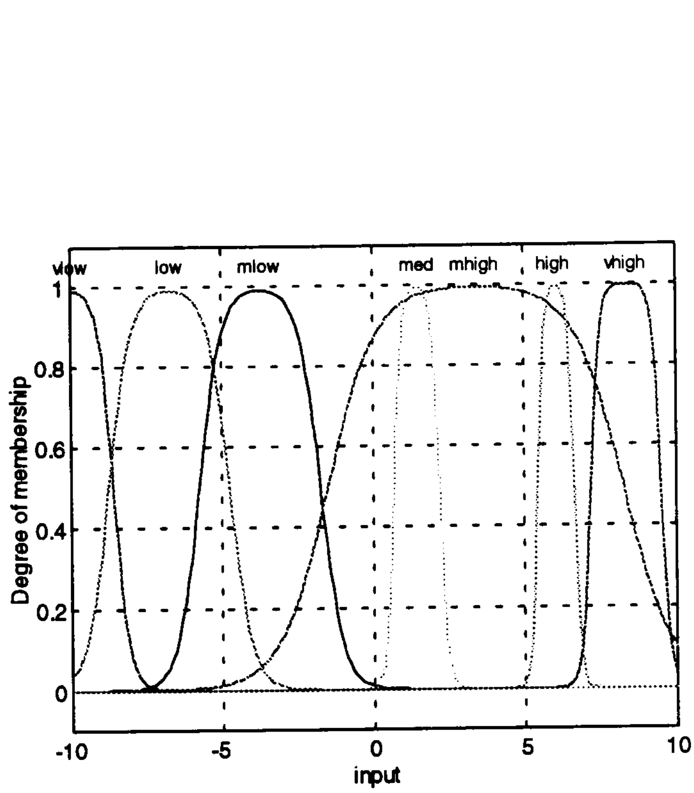
(a)



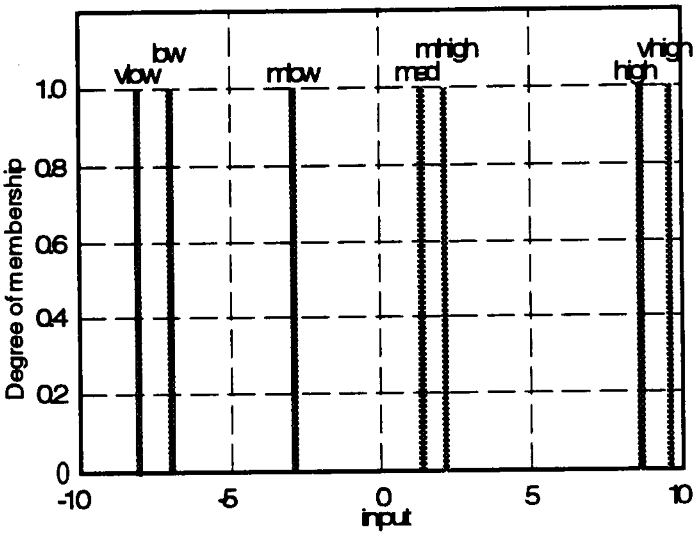
(b)

Figure D-18. Sigmoidal MBF Third-order Definition

(a) input membership functions, (b) output membership functions



(a)



(b)

Figure D-19. Sigmoidal MBF Piecewise-linear Definition

(a) input membership functions, (b) output membership functions

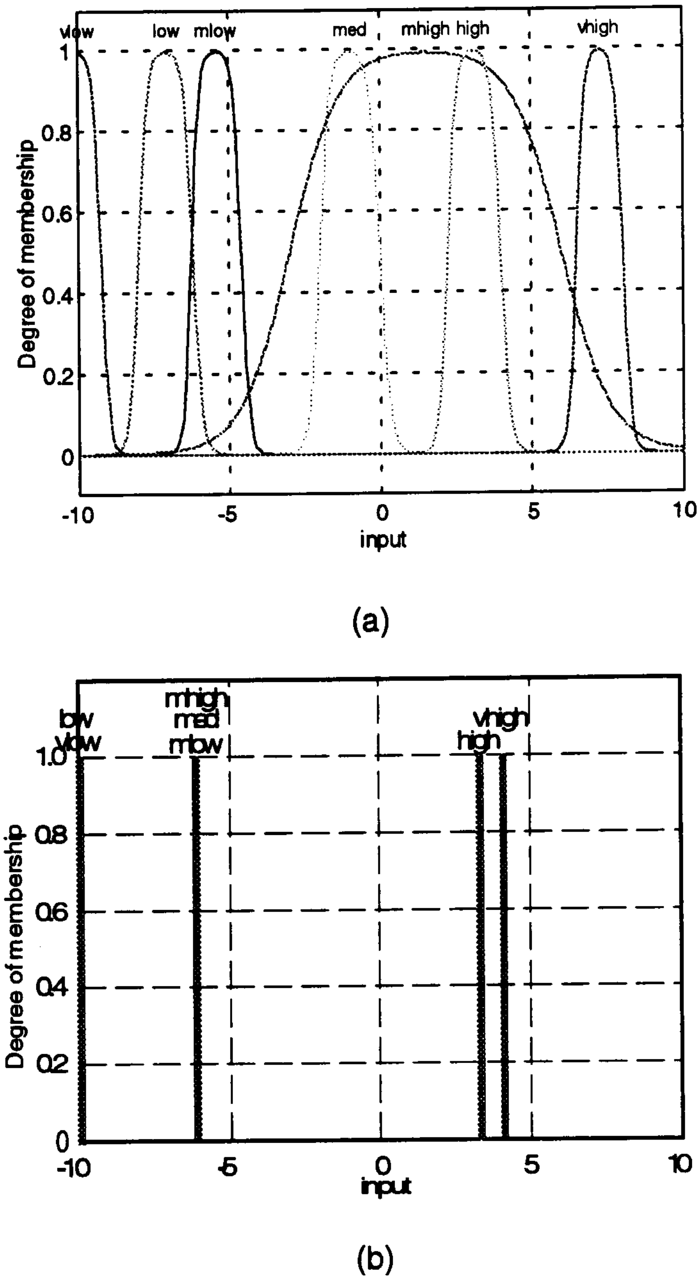


Figure D-20. Sigmoidal MBF Sinusoidal Definition

(a) input membership functions, (b) output membership functions

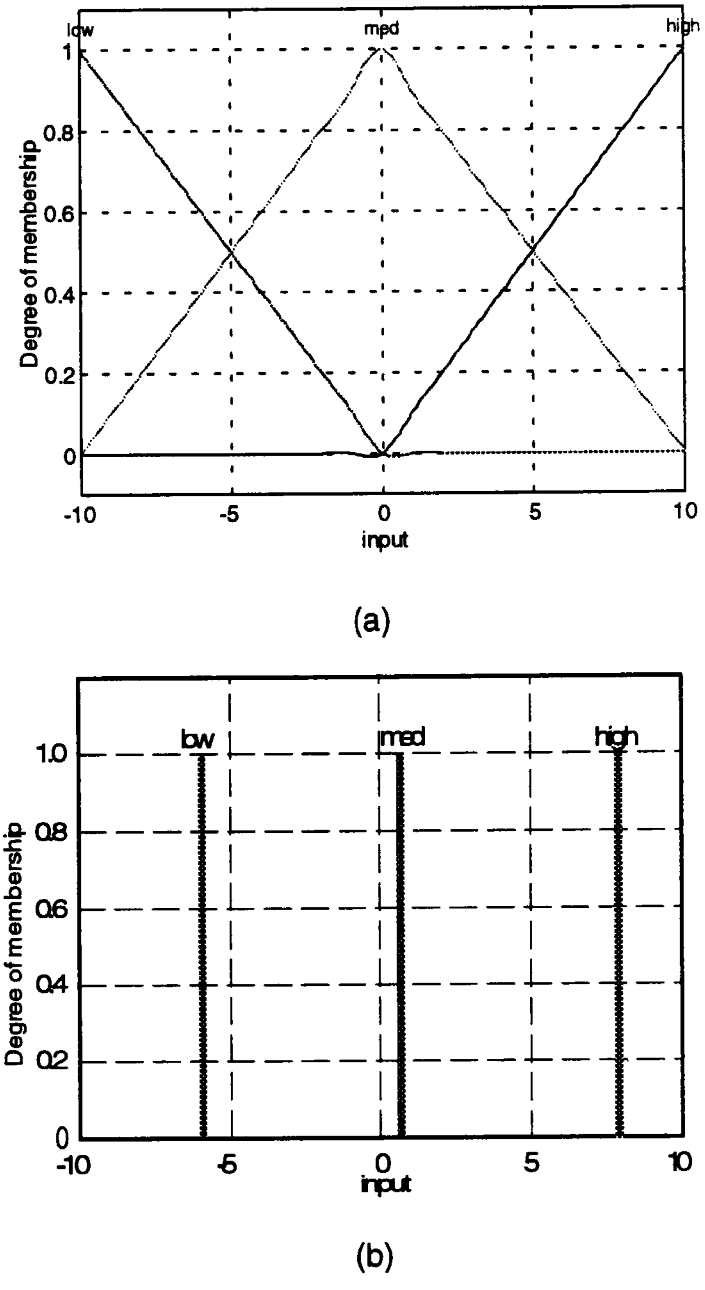
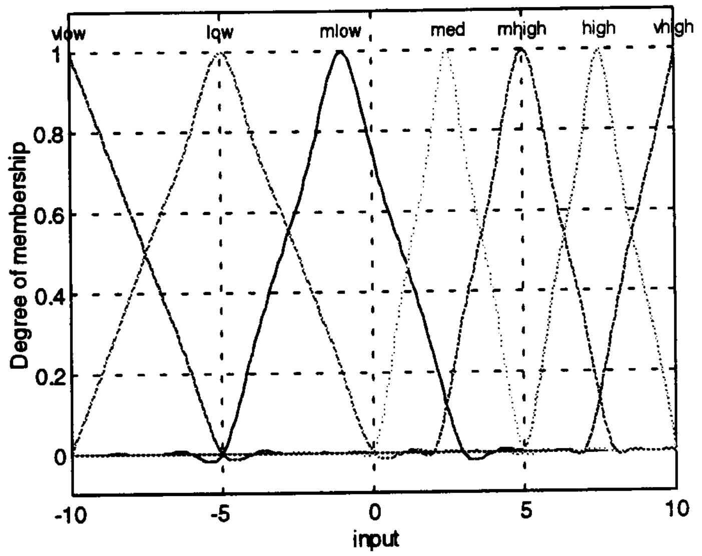
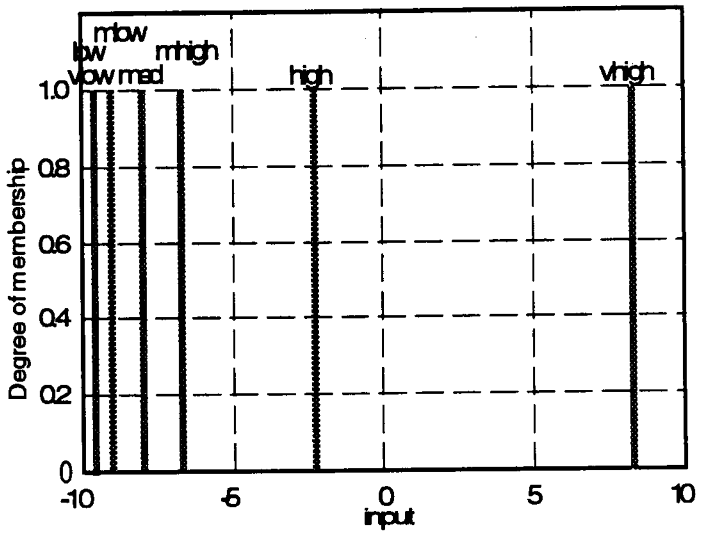


Figure D-21. Cubic Spline MBF First-order Definition

(a) input membership functions, (b) output membership functions



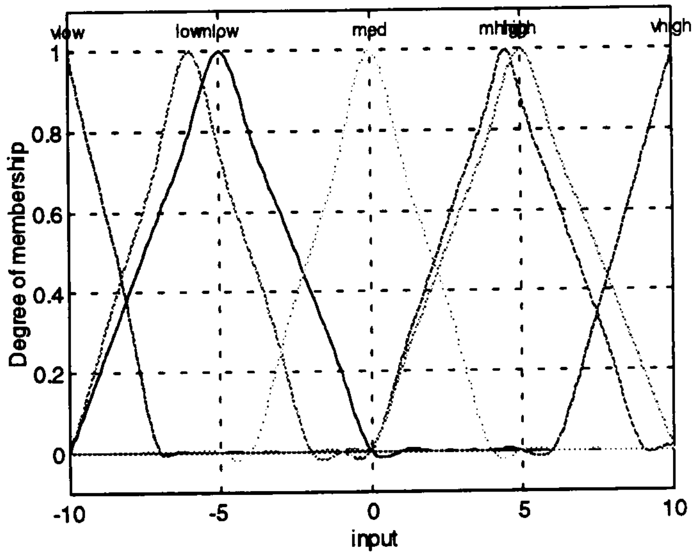
(a)



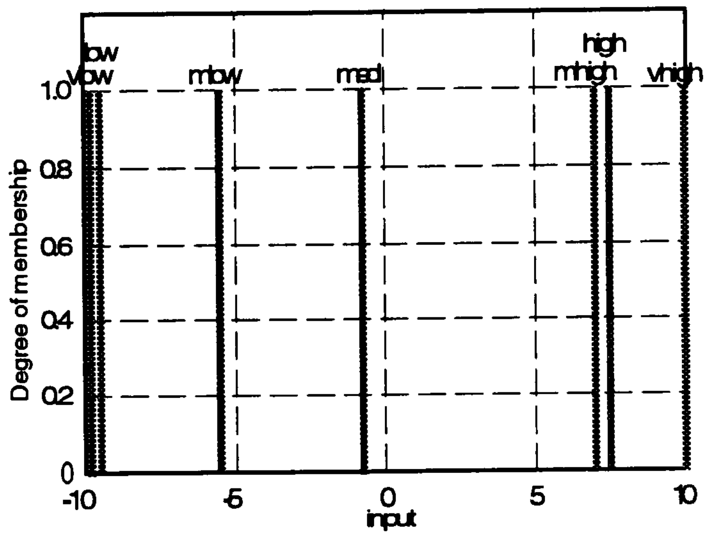
(b)

Figure D-22. Cubic Spline MBF Second-order Definition

(a) input membership functions, (b) output membership functions



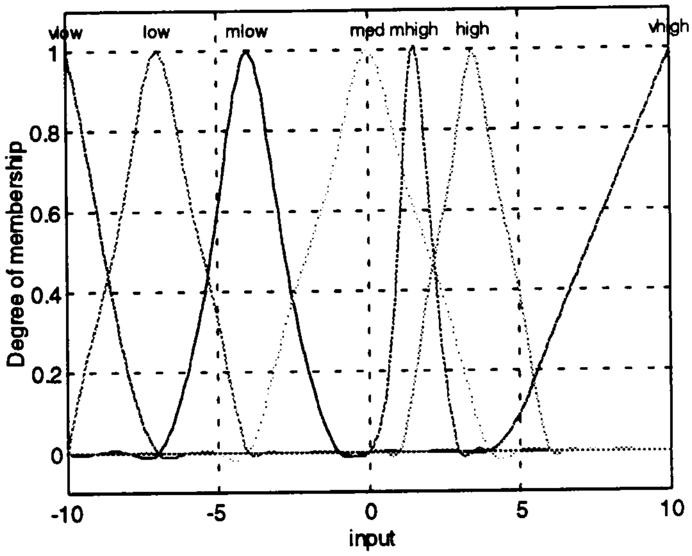
(a)



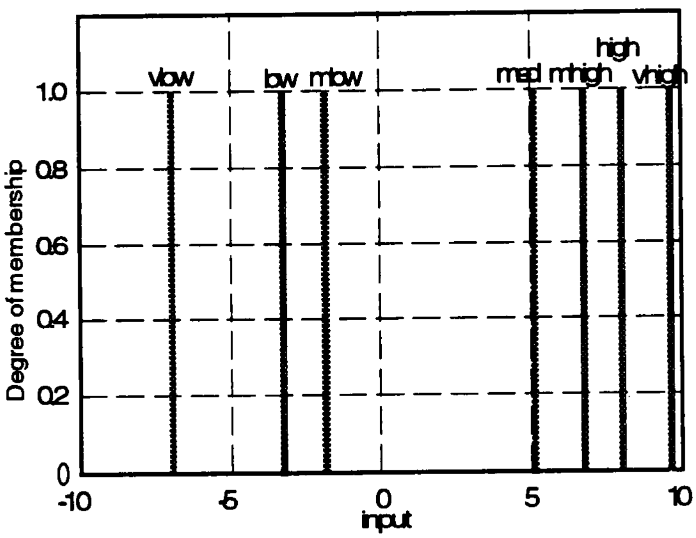
(b)

Figure D-23. Cubic Spline MBF Third-order Definition

(a) input membership functions, (b) output membership functions



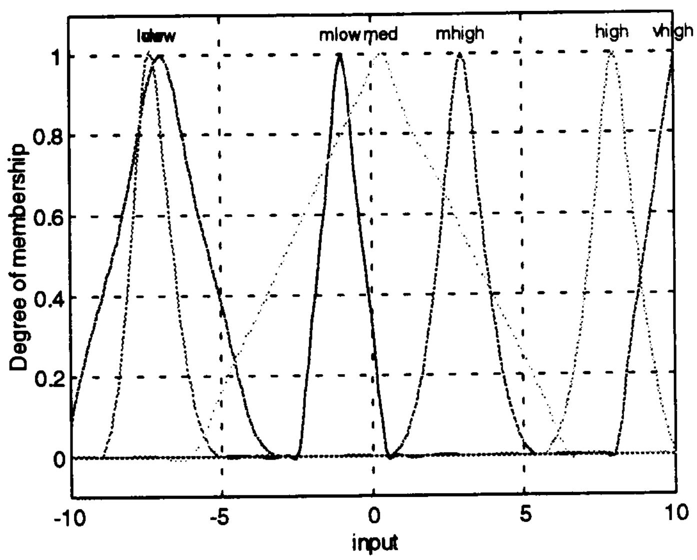
(a)



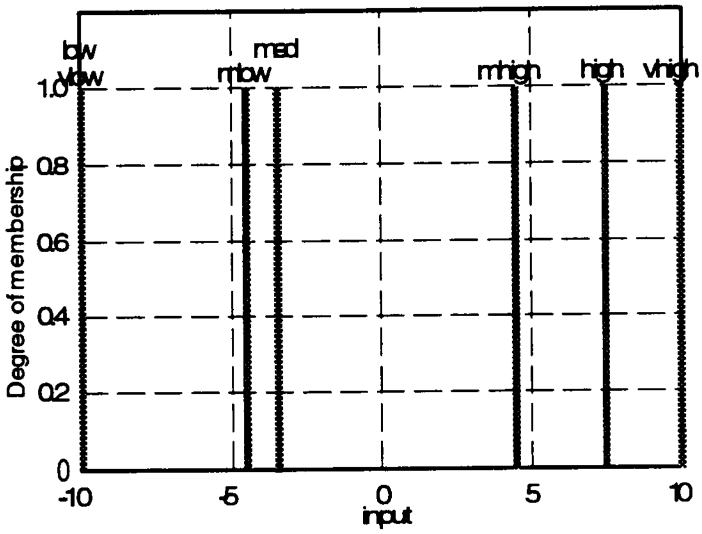
(b)

Figure D-24. Cubic Spline MBF Piecewise-linear Definition

(a) input membership functions, (b) output membership functions



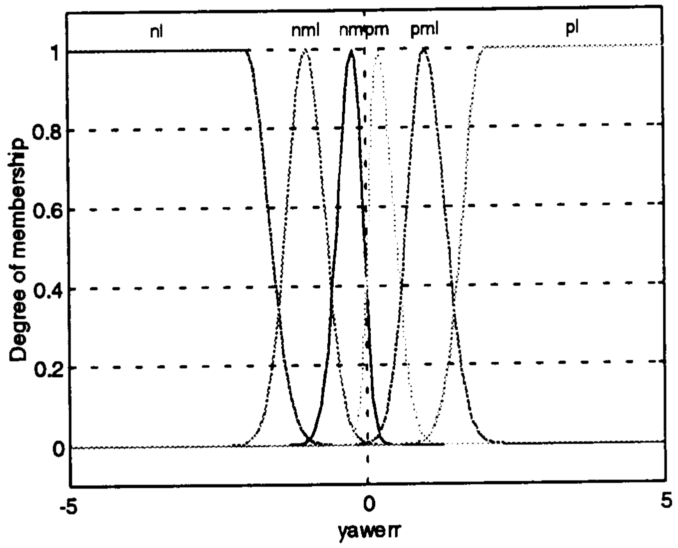
(a)



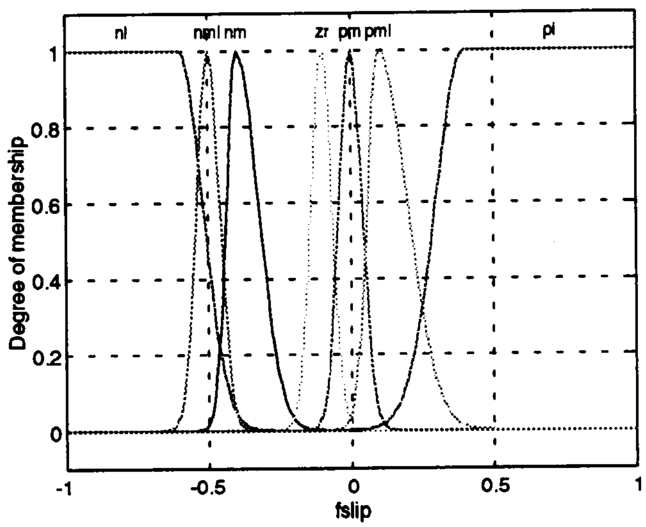
(b)

Figure D-25. Cubic Spline MBF Sinusoidal Definition

(a) input membership functions, (b) output membership functions



(a)



(b)

Figure D-26. Vehicular Model Input Membership Functions

(a) yaw error membership functions, (b) rslip membership functions

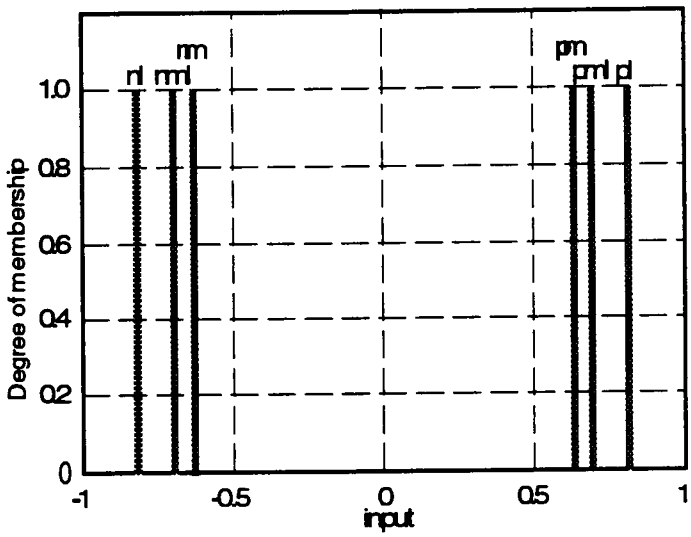


Figure D-27. Vehicular Model Output Membership Functions

APPENDIX E

FUZZY INFERENCE SYSTEM DEFINITIONS

APPENDIX E

Triangular MBF First-Order Fuzzy System

1. Name TRI3S01
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [3]
5. NumOutputMFs [3]
6. NumRules 3
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-20 -10 0 0]
17. [-10 -1.11e-016 10 0]
18. [0 10 20 0]
19. OutMFParams [0 -5.7]
20. [0 1]
21. [0 7.7]

1. If (input is low) then (output is high) (1)
2. If (input is med) then (output is med) (1)
3. If (input is high) then (output is low) (1)

Triangular MBF Second-order Fuzzy System

1. Name TRI7S02C
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-15 -10 -5 0]
17. [-10 -5 0 0]
18. [-5 -1 3 0]
19. [2 5 8 0]
20. [7 10 13 0]
21. [0 2.5 5 0]
22. [5 7.5 10 0]
23. OutMFParams [0 -9.5]
24. [0 -9.5]
25. [0 -8.75]
26. [0 -8]
27. [0 -6.6]
28. [0 -2.4]
29. [0 8.2]

1. If (input is vlow) then (output is vhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is med) (1)
4. If (input is med) then (output is vlow) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is mlow) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Triangular MBF Third-order Fuzzy System

1. Name TRI7S03
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-13 -10 -7 0]
17. [-10 -6 -2 0]
18. [-10 -5 0 0]
19. [-4 0 4 0]
20. [0 4.5 9 0]
21. [0 5 10 0]
22. [6 10 14 0]
23. OutMFParams [0 -10]
24. [0 -9.35]
25. [0 -5.5]
26. [0 -1]
27. [0 7.5]
28. [0 7.5]
29. [0 10]

1. If (input is vlow) then (output is mlow) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is vhigh) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is vlow) (1)
6. If (input is high) then (output is low) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Triangular MBF Piecewise-linear Fuzzy System

1. Name TRI7S04C
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-12.5 -10 -7.5 0]
17. [-10 -7 -4 0]
18. [-6.5 -4 -1.5 0]
19. [-4 0 4 0]
20. [0 1.5 3 0]
21. [1 3.5 6 0]
22. [4.5 10 15.5 0]
23. OutMFParams [0 -7]
24. [0 -3.25]
25. [0 -2]
26. [0 5.5]
27. [0 6.55]
28. [0 8]
29. [0 9.75]

1. If (input is vlow) then (output is vlow) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is mlow) (1)
4. If (input is med) then (output is mhigh) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is vhigh) (1)
7. If (input is vhigh) then (output is med) (1)

Triangular MBF Sinusoidal Fuzzy System

1. Name TRI7S05C
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-11 -7 -3 0]
17. [-8.75 -7.25 -5.75 0]
18. [-3.25 -1.25 0.75 0]
19. [-5.75 0.25 6.25 0]
20. [1 3 5 0]
21. [6 7.5 9 0]
22. [7.5 10 12.5 0]
23. OutMFParams [0 -10]
24. [0 -10]
25. [0 -4.5]
26. [0 -4]
27. [0 5.2]
28. [0 6.25]
29. [0 10]

1. If (input is vlow) then (output is med) (1)
2. If (input is low) then (output is vhigh) (1)
3. If (input is mlow) then (output is vlow) (1)
4. If (input is med) then (output is high) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is mlow) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Trapezoidal MBF First-order Fuzzy System

1.	Name	TRP3S01
2.	Type	sugeno
3.	Inputs/Outputs	[1 1]
4.	NumInputMFs	[3]
5.	NumOutputMFs	[3]
6.	NumRules	3
7.	AndMethod	prod
8.	OrMethod	probor
9.	ImpMethod	min
10.	AggMethod	max
11.	DefuzzMethod	wtaver
12.	InLabels	input
13.	OutLabels	output
14.	InRange	[-10 10]
15.	OutRange	[-10 10]
16.	InMFParams	[-19 -10.25 -7.5 0]
17.		[-10 -2.5 2.5 10]
18.		[0 7.5 10.25 19]
19.	OutMFParams	[0 -5.7]
20.		[0 1]
21.		[0 7.7]

1. If (input is low) then (output is high) (1)
2. If (input is med) then (output is med) (1)
3. If (input is high) then (output is low) (1)

Trapezoidal MBF Second-order Fuzzy System

1. Name TRP7S02A
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-15 -11.25 -8.75 -5]
17. [-10 -6.25 -3.75 0]
18. [-5 -2 0 3]
19. [0.3 1.8 2.8 4.3]
20. [2.2 4.3 5.7 7.8]
21. [5.5 7 8 9.5]
22. [7.6 9.4 10.6 12.4]
23. OutMFParams [0 -9.5]
24. [0 -9.5]
25. [0 -8.5]
26. [0 -8.25]
27. [0 -6.8]
28. [0 -2.3]
29. [0 8.2]

1. If (input is vlow) then (output is vhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is med) (1)
4. If (input is med) then (output is vlow) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is mlow) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Trapezoidal MBF Third-order Fuzzy System

1. Name TRP7S03D
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-13 -10.75 -9.25 -7]
17. [-10.5 -8.25 -6.75 -4.5]
18. [-9 -5.625 -3.125 0]
19. [-4.5 -1.688 0.1875 3]
20. [0 3.125 5.625 9]
21. [4 6.25 7.75 10]
22. [6 9 11 14]
23. OutMFParams [0 -10]
24. [0 -9]
25. [0 -6.5]
26. [0 -0.5]
27. [0 8.25]
28. [0 9.25]
29. [0 9.25]

1. If (input is vlow) then (output is mlow) (1)
2. If (input is low) then (output is mhigh) (1)
3. If (input is mlow) then (output is high) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is vlow) (1)
7. If (input is vhigh) then (output is vhigh) (1)

Trapezoidal MBF Piecewise-linear Fuzzy System

1. Name TRP7S04
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-20 -15 -10 -7.5]
17. [-10 -7.75 -6.25 -4]
18. [-6.5 -4.25 -2.75 -0.5]
19. [0 1.125 1.875 3]
20. [-7.5 0 5 12.5]
21. [5 5.938 6.562 7.5]
22. [6.75 8.25 9.25 10.75]
23. OutMFParams [0 -6.7]
24. [0 -6.25]
25. [0 -5.167]
26. [0 1]
27. [0 3.5]
28. [0 8]
29. [0 9.5]

1. If (input is vlow) then (output is vlow) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is low) (1)
4. If (input is med) then (output is mlow) (1)
5. If (input is mhigh) then (output is vhigh) (1)
6. If (input is high) then (output is mhigh) (1)
7. If (input is vhigh) then (output is med) (1)

Trapezoidal MBF Sinusoidal Fuzzy System

1. Name TRP7S05
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-20 -15 -9.25 -7.75]
17. [-8 -7.531 -7.219 -6.75]
18. [-7 -5.688 -4.812 -3.5]
19. [-2.8 -1.563 -0.737 0.3]
20. [-16.5 -3 6 19.5]
21. [1.65 2.719 3.53 4.75]
22. [5.8 6.925 7.675 8.8]
23. OutMFParams [0 -10]
24. [0 -10]
25. [0 -10]
26. [0 -10]
27. [0 -9]
28. [0 4]
29. [0 5.25]

1. If (input is vlow) then (output is mhigh) (1)
2. If (input is low) then (output is vhigh) (1)
3. If (input is mlow) then (output is med) (1)
4. If (input is med) then (output is low) (1)
5. If (input is mhigh) then (output is high) (1)
6. If (input is high) then (output is vlow) (1)
7. If (input is vhigh) then (output is mlow) (1)

Gaussian MBF First-order Fuzzy System

1.	Name	GAU5S01A
2.	Type	sugeno
3.	Inputs/Outputs	[1 1]
4.	NumInputMFs	[5]
5.	NumOutputMFs	[5]
6.	NumRules	5
7.	AndMethod	prod
8.	OrMethod	probor
9.	ImpMethod	min
10.	AggMethod	max
11.	DefuzzMethod	wtaver
12.	InLabels	input
13.	OutLabels	output
14.	InRange	[-10 10]
15.	OutRange	[-10 10]
16.	InMFParams	[6.35 -10 0 0]
17.		[4.5 0 0 0]
18.		[6.35 10 0 0]
19.		[0.5 -10 0 0]
20.		[0.5 10 0 0]
21.	OutMFParams	[0 -6.3]
22.		[0 1]
23.		[0 8.5]
24.		[0 -5.7]
25.		[0 7.7]

1. If (input is mlow) then (output is high) (1)
2. If (input is low) then (output is mhigh) (1)
3. If (input is med) then (output is med) (1)
4. If (input is mhigh) then (output is low) (1)
5. If (input is high) then (output is mlow) (1)

Gaussian MBF Second-order Fuzzy System

1. Name GAU7S02B
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [1.5 -7.5 0 0]
17. [1.5 -4.75 0 0]
18. [1.25 -1.75 0 0]
19. [2.4 1.7 0 0]
20. [2.123 6 0 0]
21. [1 -10 0 0]
22. [2.123 10 0 0]
23. OutMFParams [0 -9]
24. [0 -6.2]
25. [0 -6]
26. [0 -3]
27. [0 3.3]
28. [0 -10]
29. [0 9.3]

1. If (input is vlow) then (output is vhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is mhigh) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is vlow) (1)
7. If (input is vhigh) then (output is mlow) (1)

Gaussian MBF Third-order Fuzzy System

1. Name GAU7S03D
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [0.5 -10 0 0]
17. [1.5 -10 0 0]
18. [2.8 -5.75 0 0]
19. [1.5 -0.5 0 0]
20. [2.8 5 0 0]
21. [1.5 10 0 0]
22. [0.5 10 0 0]
23. OutMFParams [0 -9.5]
24. [0 -8]
25. [0 -3.5]
26. [0 0]
27. [0 7]
28. [0 9.5]
29. [0 10]

1. If (input is vlow) then (output is low) (1)
2. If (input is low) then (output is mlow) (1)
3. If (input is mlow) then (output is high) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is vlow) (1)
6. If (input is high) then (output is mhigh) (1)
7. If (input is vhigh) then (output is vhigh) (1)

Gaussian MBF Piecewise-linear Fuzzy System

1. Name GAU7S04A
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [1 -10 0 0]
17. [1.2 -6.7 0 0]
18. [1.5 -3.75 0 0]
19. [0.5 1.5 0 0]
20. [4 3.5 0 0]
21. [0.45 6.05 0 0]
22. [0.75 8.5 0 0]
23. OutMFParams [0 -6.75]
24. [0 -5.5]
25. [0 -4]
26. [0 0.5]
27. [0 3.25]
28. [0 9.5]
29. [0 9.5]

1. If (input is vlow) then (output is low) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is mlow) (1)
4. If (input is med) then (output is vlow) (1)
5. If (input is mhigh) then (output is vhigh) (1)
6. If (input is high) then (output is mhigh) (1)
7. If (input is vhigh) then (output is med) (1)

Gaussian MBF Sinusoidal Fuzzy System

1. Name GAU7S05
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [0.4 -10 0 0]
17. [0.4 -6.9 0 0]
18. [0.55 -5.35 0 0]
19. [0.65 -1 0 0]
20. [3 1.5 0 0]
21. [0.75 3 0 0]
22. [0.54 7.25 0 0]
23. OutMFParams [0 -10]
24. [0 -10]
25. [0 -6]
26. [0 -6]
27. [0 -6]
28. [0 3.55]
29. [0 4.75]

1. If (input is vlow) then (output is mhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is med) (1)
4. If (input is med) then (output is low) (1)
5. If (input is mhigh) then (output is vhigh) (1)
6. If (input is high) then (output is vlow) (1)
7. If (input is vhigh) then (output is mlow) (1)

Sigmoidal MBF First-order Fuzzy System

1. Name SIG5S01A
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [5]
5. NumOutputMFs [5]
6. NumRules 5
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [2 -13 2 -7]
17. [2 -8 2 -2]
18. [2 -3 2 3]
19. [2 2 2 8]
20. [2 7 2 13]
21. OutMFParams [0 -5.7]
22. [0 -2.35]
23. [0 1]
24. [0 4.35]
25. [0 7.7]

1. If (input is vlow) then (output is vhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is med) then (output is med) (1)
4. If (input is high) then (output is low) (1)
5. If (input is vhigh) then (output is vlow) (1)

Sigmoidal MBF Second-order Fuzzy System

1. Name SIG7S02C
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [6.5 -10.85 6.5 -9.15]
17. [4 -9.75 4 -6.25]
18. [4 -6.877 4 -3.375]
19. [4 -4 4 -0.5]
20. [3 -0.55 3 3.95]
21. [3 3.6 3 8.1]
22. [3 7.75 3 12.25]
23. OutMFParams [0 -9.5]
24. [0 -9]
25. [0 -7]
26. [0 -6]
27. [0 -2]
28. [0 4.05]
29. [0 10]

1. If (input is vlow) then (output is vhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is mhigh) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is vlow) (1)
7. If (input is vhigh) then (output is mlow) (1)

Sigmoidal MBF Third-order Fuzzy System

1. Name SIG7S03
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [10 -10.5 10 -9.5]
17. [2.5 -12 2.5 -8]
18. [2 -8.75 2 -2.75]
19. [2.25 -2.75 2.25 1.75]
20. [2 2 2 8]
21. [2.25 7.75 2.25 12.25]
22. [10 9.5 10 10.5]
23. OutMFParams [0 -9.5]
24. [0 -8]
25. [0 -2.5]
26. [0 0]
27. [0 3.75]
28. [0 9.5]
29. [0 10]

1. If (input is vlow) then (output is low) (1)
2. If (input is low) then (output is mlow) (1)
3. If (input is mlow) then (output is high) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is vlow) (1)
6. If (input is high) then (output is mhigh) (1)
7. If (input is vhigh) then (output is vhigh) (1)

Sigmoidal MBF Piecewise-linear Fuzzy System

1. Name SIG7S04A
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [4 -11.4 4 -8.6]
17. [2.75 -8.75 2.75 -4.75]
18. [2.75 -5.75 2.75 -1.75]
19. [8 0.7 8 2.2]
20. [1.25 -1.5 1.25 8.25]
21. [10 5.45 10 6.65]
22. [7 7.2 7 9.5]
23. OutMFParams [0 -7.75]
24. [0 -7]
25. [0 -2.75]
26. [0 1.55]
27. [0 2.25]
28. [0 8.5]
29. [0 9.5]

1. If (input is vlow) then (output is low) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is mlow) (1)
4. If (input is med) then (output is vlow) (1)
5. If (input is mhigh) then (output is vhigh) (1)
6. If (input is high) then (output is mhigh) (1)
7. If (input is vhigh) then (output is med) (1)

Sigmoidal MBF Sinusoidal Fuzzy System

1. Name SIG7S05
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [8 -10.75 8 -9.25]
17. [7 -8 7 -6.2]
18. [8 -6.25 8 -4.6]
19. [6 -2 6 0]
20. [1.25 -3 1.25 6]
21. [7 2.25 7 4]
22. [8 6.5 8 8.015]
23. OutMFParams [0 -10]
24. [0 -10]
25. [0 -6]
26. [0 -6]
27. [0 -6]
28. [0 3.55]
29. [0 4]

1. If (input is vlow) then (output is mlow) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is med) (1)
4. If (input is med) then (output is low) (1)
5. If (input is mhigh) then (output is vhigh) (1)
6. If (input is high) then (output is vlow) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Cubic Spline MBF First-order Fuzzy System

1. Name SPL3S01
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [3]
5. NumOutputMFs [3]
6. NumRules 3
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-10 0 10 0]
17. [-10 0 10 0]
18. [-10 0 10 0]
19. OutMFParams [0 -5.7]
20. [0 1]
21. [0 7.7]

1. If (input is low) then (output is high) (1)
2. If (input is med) then (output is med) (1)
3. If (input is high) then (output is low) (1)

Cubic Spline MBF Second-order Fuzzy System

1. Name SPL7S02
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-10 0 10 0]
17. [-10 0 10 0]
18. [-10 0 10 0]
19. [-10 0 10 0]
20. [-10 0 10 0]
21. [-10 0 10 0]
22. [-10 0 10 0]
23. OutMFParams [0 -9.5]
24. [0 -9.5]
25. [0 -8.75]
26. [0 -8]
27. [0 -6.6]
28. [0 -2.4]
29. [0 8.2]

1. If (input is vlow) then (output is vhigh) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is med) (1)
4. If (input is med) then (output is vlow) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is mlow) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Cubic Spline MBF Third-order Fuzzy System

1. Name SPL7S03A
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-10 0 10 0]
17. [-10 0 10 0]
18. [-10 0 10 0]
19. [-10 0 10 0]
20. [-10 0 10 0]
21. [-10 0 10 0]
22. [-10 0 10 0]
23. OutMFParams [0 -9.775]
24. [0 -9.35]
25. [0 -5.5]
26. [0 -1]
27. [0 7.33]
28. [0 7.5]
29. [0 10]

1. If (input is vlow) then (output is mlow) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is vhigh) (1)
4. If (input is med) then (output is med) (1)
5. If (input is mhigh) then (output is vlow) (1)
6. If (input is high) then (output is low) (1)
7. If (input is vhigh) then (output is mhigh) (1)

Cubic Spline MBF Piecewise-linear Fuzzy System

1. Name SPL7S04
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-10 0 10 0]
17. [-10 0 10 0]
18. [-10 0 10 0]
19. [-10 0 10 0]
20. [-10 0 10 0]
21. [-10 0 10 0]
22. [-10 0 10 0]
23. OutMFParams [0 -7]
24. [0 -3.25]
25. [0 -2]
26. [0 5.25]
27. [0 6.55]
28. [0 8.25]
29. [0 9.75]

1. If (input is vlow) then (output is vlow) (1)
2. If (input is low) then (output is high) (1)
3. If (input is mlow) then (output is mlow) (1)
4. If (input is med) then (output is mhigh) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is vhigh) (1)
7. If (input is vhigh) then (output is med) (1)

Cubic Spline MBF Sinusoidal Fuzzy System

1. Name SPL7S05A
2. Type sugeno
3. Inputs/Outputs [1 1]
4. NumInputMFs [7]
5. NumOutputMFs [7]
6. NumRules 7
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels input
13. OutLabels output
14. InRange [-10 10]
15. OutRange [-10 10]
16. InMFParams [-10 0 10 0]
17. [-10 0 10 0]
18. [-10 0 10 0]
19. [-10 0 10 0]
20. [-10 0 10 0]
21. [-10 0 10 0]
22. [-10 0 10 0]
23. OutMFParams [0 -10]
24. [0 -10]
25. [0 -4.65]
26. [0 -3.5]
27. [0 4.75]
28. [0 6.2]
29. [0 10]

1. If (input is vlow) then (output is mlow) (1)
2. If (input is low) then (output is vhigh) (1)
3. If (input is mlow) then (output is vlow) (1)
4. If (input is med) then (output is mhigh) (1)
5. If (input is mhigh) then (output is low) (1)
6. If (input is high) then (output is med) (1)
7. If (input is vhigh) then (output is high) (1)

Vehicular Model Front Fuzzy Controller

1. Name FUZFRNT5
2. Type sugeno
3. Inputs/Outputs [2 1]
4. NumInputMFs [6 7]
5. NumOutputMFs [7]
6. NumRules 42
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels yawerr
13. fslip
14. OutLabels fsteer
15. InRange [-5 5]
16. [-1 1]
17. OutRange [-1 1]
18. InMFParams [1 -5 0.35 -2]
19. [0.33 -1 0.3 -1]
20. [0.27 -0.2 0.15 -0.2]
21. [0.15 0.2 0.27 0.2]
22. [0.3 1 0.33 1]
23. [0.35 2 1 5]
24. [1 -1 0.08 -0.6]
25. [0.04 -0.5 0.04 -0.5]
26. [0.035 -0.4 0.075 -0.4]
27. [0.04 -0.1 0.035 -0.1]
28. [0.04 0 0.04 0]
29. [0.04 0.1 0.1 0.1]
30. [0.1 0.4 1 1]
31. OutMFParams [0 0 -0.78]
32. [0 0 -0.7]
33. [0 0 -0.62]
34. [0 0 0]
35. [0 0 0.62]
36. [0 0 0.7]
37. [0 0 0.78]

1. If (yawerr is pl) and (fslip is nl) then (fsteer is nl) (1)
2. If (yawerr is pl) and (fslip is nml) then (fsteer is nml) (1)
3. If (yawerr is pl) and (fslip is nm) then (fsteer is nm) (1)

4. If (yawerr is pl) and (fslip is zr) then (fsteer is pm) (1)
5. If (yawerr is pl) and (fslip is pm) then (fsteer is pm) (1)
6. If (yawerr is pl) and (fslip is pml) then (fsteer is pm) (1)
7. If (yawerr is pl) and (fslip is pl) then (fsteer is pml) (1)
8. If (yawerr is pml) and (fslip is nl) then (fsteer is nml) (1)
9. If (yawerr is pml) and (fslip is nml) then (fsteer is nml) (1)
10. If (yawerr is pml) and (fslip is nm) then (fsteer is nm) (1)
11. If (yawerr is pml) and (fslip is zr) then (fsteer is pm) (1)
12. If (yawerr is pml) and (fslip is pm) then (fsteer is pm) (1)
13. If (yawerr is pml) and (fslip is pml) then (fsteer is pm) (1)
14. If (yawerr is pml) and (fslip is pl) then (fsteer is pm) (1)
15. If (yawerr is pm) and (fslip is nl) then (fsteer is nm) (1)
16. If (yawerr is pm) and (fslip is nml) then (fsteer is nm) (1)
17. If (yawerr is pm) and (fslip is nm) then (fsteer is nm) (1)
18. If (yawerr is pm) and (fslip is zr) then (fsteer is pm) (1)
19. If (yawerr is pm) and (fslip is pm) then (fsteer is pm) (1)
20. If (yawerr is pm) and (fslip is pml) then (fsteer is pm) (1)
21. If (yawerr is pm) and (fslip is pl) then (fsteer is pm) (1)
22. If (yawerr is nm) and (fslip is nl) then (fsteer is pm) (1)
23. If (yawerr is nm) and (fslip is nml) then (fsteer is pm) (1)
24. If (yawerr is nm) and (fslip is nm) then (fsteer is pm) (1)
25. If (yawerr is nm) and (fslip is zr) then (fsteer is nm) (1)
26. If (yawerr is nm) and (fslip is pm) then (fsteer is nm) (1)
27. If (yawerr is nm) and (fslip is pml) then (fsteer is nm) (1)
28. If (yawerr is nm) and (fslip is pl) then (fsteer is nm) (1)
29. If (yawerr is nml) and (fslip is nl) then (fsteer is pl) (1)
30. If (yawerr is nml) and (fslip is nml) then (fsteer is pml) (1)
31. If (yawerr is nml) and (fslip is nm) then (fsteer is pm) (1)
32. If (yawerr is nml) and (fslip is zr) then (fsteer is nm) (1)
33. If (yawerr is nml) and (fslip is pm) then (fsteer is nm) (1)
34. If (yawerr is nml) and (fslip is pml) then (fsteer is nm) (1)
35. If (yawerr is nml) and (fslip is pl) then (fsteer is nml) (1)
36. If (yawerr is nl) and (fslip is nl) then (fsteer is pl) (1)
37. If (yawerr is nl) and (fslip is nml) then (fsteer is pml) (1)
38. If (yawerr is nl) and (fslip is nm) then (fsteer is pm) (1)
39. If (yawerr is nl) and (fslip is zr) then (fsteer is nm) (1)
40. If (yawerr is nl) and (fslip is pm) then (fsteer is nm) (1)
41. If (yawerr is nl) and (fslip is pml) then (fsteer is nm) (1)
42. If (yawerr is nl) and (fslip is pl) then (fsteer is nml) (1)

Vehicular Model Rear Fuzzy Controller

1. Name FUZREAR5
2. Type sugeno
3. Inputs/Outputs [2 1]
4. NumInputMFs [6 7]
5. NumOutputMFs [7]
6. NumRules 42
7. AndMethod prod
8. OrMethod probor
9. ImpMethod min
10. AggMethod max
11. DefuzzMethod wtaver
12. InLabels yawerr
13. rslip
14. OutLabels rsteer
15. InRange [-5 5]
16. [-1 1]
17. OutRange [-1 1]
18. InMFParams [1 -5 0.35 -2]
19. [0.33 -1 0.3 -1]
20. [0.27 -0.2 0.15 -0.2]
21. [0.15 0.2 0.27 0.2]
22. [0.3 1 0.33 1]
23. [0.35 2 1 5]
24. [1 -1 0.08 -0.6]
25. [0.04 -0.5 0.04 -0.5]
26. [0.035 -0.4 0.075 -0.4]
27. [0.04 -0.1 0.035 -0.1]
28. [0.04 0 0.04 0]
29. [0.04 0.1 0.1 0.1]
30. [0.1 0.4 1 1]
31. OutMFParams [0 0 -0.78]
32. [0 0 -0.7]
33. [0 0 -0.62]
34. [0 0 0]
35. [0 0 0.62]
36. [0 0 0.7]
37. [0 0 0.78]

1. If (yawerr is pl) and (rslip is nl) then (rsteer is pl) (1)
2. If (yawerr is pl) and (rslip is nml) then (rsteer is pml) (1)
3. If (yawerr is pl) and (rslip is nm) then (rsteer is pm) (1)

4. If (yawerr is pl) and (rslip is zr) then (rsteer is nm) (1)
5. If (yawerr is pl) and (rslip is pm) then (rsteer is nm) (1)
6. If (yawerr is pl) and (rslip is pml) then (rsteer is nm) (1)
7. If (yawerr is pl) and (rslip is pl) then (rsteer is nml) (1)
8. If (yawerr is pml) and (rslip is nl) then (rsteer is pml) (1)
9. If (yawerr is pml) and (rslip is nml) then (rsteer is pml) (1)
10. If (yawerr is pml) and (rslip is nm) then (rsteer is pm) (1)
11. If (yawerr is pml) and (rslip is zr) then (rsteer is nm) (1)
12. If (yawerr is pml) and (rslip is pm) then (rsteer is nm) (1)
13. If (yawerr is pml) and (rslip is pml) then (rsteer is nm) (1)
14. If (yawerr is pml) and (rslip is pl) then (rsteer is nm) (1)
15. If (yawerr is pm) and (rslip is nl) then (rsteer is pm) (1)
16. If (yawerr is pm) and (rslip is nml) then (rsteer is pm) (1)
17. If (yawerr is pm) and (rslip is nm) then (rsteer is pm) (1)
18. If (yawerr is pm) and (rslip is zr) then (rsteer is nm) (1)
19. If (yawerr is pm) and (rslip is pm) then (rsteer is nm) (1)
20. If (yawerr is pm) and (rslip is pml) then (rsteer is nm) (1)
21. If (yawerr is pm) and (rslip is pl) then (rsteer is nm) (1)
22. If (yawerr is nm) and (rslip is nl) then (rsteer is nm) (1)
23. If (yawerr is nm) and (rslip is nml) then (rsteer is nm) (1)
24. If (yawerr is nm) and (rslip is nm) then (rsteer is nm) (1)
25. If (yawerr is nm) and (rslip is zr) then (rsteer is pm) (1)
26. If (yawerr is nm) and (rslip is pm) then (rsteer is pm) (1)
27. If (yawerr is nm) and (rslip is pml) then (rsteer is pm) (1)
28. If (yawerr is nm) and (rslip is pl) then (rsteer is pm) (1)
29. If (yawerr is nml) and (rslip is nl) then (rsteer is nl) (1)
30. If (yawerr is nml) and (rslip is nml) then (rsteer is nml) (1)
31. If (yawerr is nml) and (rslip is nm) then (rsteer is nm) (1)
32. If (yawerr is nml) and (rslip is zr) then (rsteer is pm) (1)
33. If (yawerr is nml) and (rslip is pm) then (rsteer is pm) (1)
34. If (yawerr is nml) and (rslip is pml) then (rsteer is pm) (1)
35. If (yawerr is nml) and (rslip is pl) then (rsteer is pml) (1)
36. If (yawerr is nl) and (rslip is nl) then (rsteer is nl) (1)
37. If (yawerr is nl) and (rslip is nml) then (rsteer is nml) (1)
38. If (yawerr is nl) and (rslip is nm) then (rsteer is nm) (1)
39. If (yawerr is nl) and (rslip is zr) then (rsteer is pm) (1)
40. If (yawerr is nl) and (rslip is pm) then (rsteer is pm) (1)
41. If (yawerr is nl) and (rslip is pml) then (rsteer is pm) (1)
42. If (yawerr is nl) and (rslip is pl) then (rsteer is pml) (1)

VITA

Kevin J. Doran received his Bachelor of Science degree in electrical engineering from Arizona State University in 1989. In 1991, he entered the University of Tennessee as a part-time graduate student and in 1996, received his Master of Science degree in electrical engineering. Currently, he is a senior engineer with Raytheon Electronic Systems where he designs programmable logic devices and works in the intelligent transportation field.