

Fuzzy Logic Systems for Engineering: A Tutorial

JERRY M. MENDEL, FELLOW, IEEE

A fuzzy logic system (FLS) is unique in that it is able to simultaneously handle numerical data and linguistic knowledge. It is a nonlinear mapping of an input data (feature) vector into a scalar output, i.e., it maps numbers into numbers. Fuzzy set theory and fuzzy logic establish the specifics of the nonlinear mapping. This tutorial paper provides a guided tour through those aspects of fuzzy sets and fuzzy logic that are necessary to synthesize a FLS. It does this by starting with crisp set theory and dual logic and demonstrating how both can be extended to their fuzzy counterparts. Because engineering systems are, for the most part, causal, we impose causality as a constraint on the development of the FLS. Doing this lets us steer down a very special and widely used tributary of the FL literature, one that is valuable for engineering applications of FL, but may not be as valuable for nonengineering applications.

After synthesizing a FLS, we demonstrate that it can be expressed mathematically as a linear combination of fuzzy basis functions, and is a nonlinear universal function approximator, a property that it shares with feedforward neural networks. The fuzzy basis function expansion is very powerful because its basis functions can be derived from either numerical data or linguistic knowledge, both of which can be cast into the forms of IF-THEN rules. To date, a FLS is the only approximation method that is able to incorporate both types of knowledge in a unified mathematical manner.

I. INTRODUCTION

A. Problem Knowledge

For many problems two distinct forms of problem knowledge exist: 1) *objective knowledge*, which is used all the time in engineering problem formulations (e.g., mathematical models), and 2) *subjective knowledge*, which represents linguistic information that is usually impossible to quantify using traditional mathematics (e.g., rules, expert information, design requirements). Examples of objective knowledge are: equations of motion for a submarine, spacecraft, robot, etc.; convolutional model that describes a communication channel or a reflection seismology experiment; and, *a priori* statistics for random parameters. Examples of subjective information are: the following rule that might be valid for tracking a submarine or any other slowly moving large object—*If a target is being tracked at one time point, then it will not be too far away at the next time point*; and,

Manuscript received June 17, 1994; revised November 10, 1994. This work was supported by the National Science Foundation under NSF Grant MIP-9122018.

The author is with the Signal and Image Processing Institute, Department of Electrical Engineering Systems, University of Southern California, Los Angeles, CA 90089-2564 USA.

IEEE Log Number 9408047.

the following rule that might be valid for processing a specific section of reflection seismology data—*If the data does not contain too many significant events, then those events are very close to one another.*

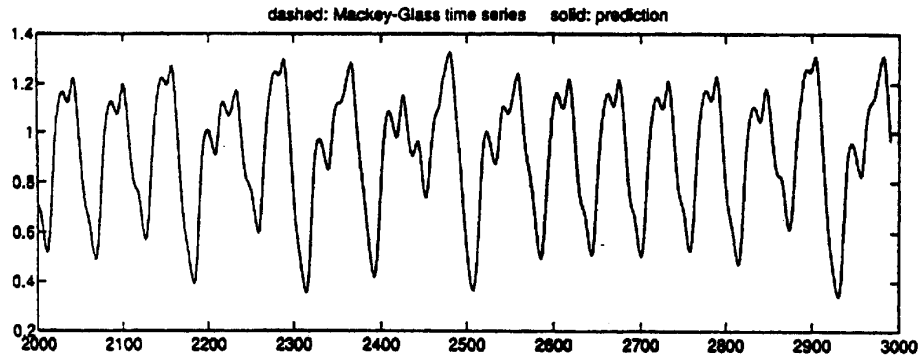
Subjective knowledge is usually ignored at the front end of engineering designs; but it is frequently used to evaluate such designs. I believe that both types of knowledge should and can be utilized to solve real problems. The two forms of knowledge can be coordinated in a logical way using *fuzzy logic (FL)*.

Two approaches for doing this have appeared in the literature: 1) *model-based approach* in which “objective” information is represented by mathematical models, and “subjective” information is represented by linguistic statements that are converted to rules, which are then quantified using FL [58]–[60]; and 2) *model-free approach* in which rules are extracted from numerical data and are then combined with linguistic information (collected from experts), both using FL. In this paper we focus on the model-free approach, because it can be applied to the same class of problems that feedforward neural networks (FFNN) can be; hence, we can compare the FL and FFNN approaches.

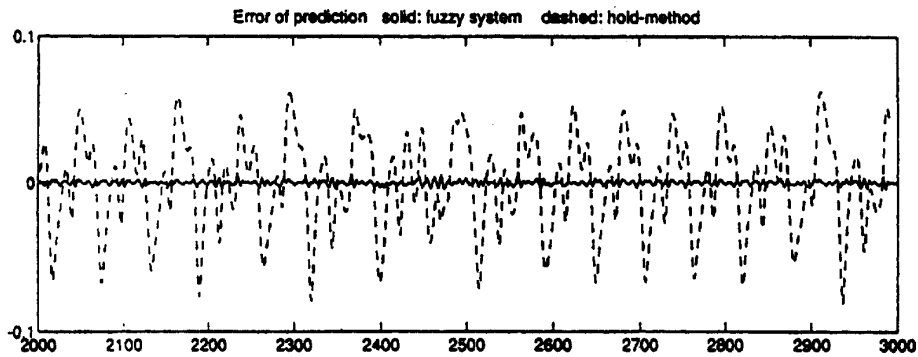
B. Purpose of this Paper

Most of the FL literature deals with mappings from fuzzy sets into fuzzy sets (we will define and illustrate what is meant by a *fuzzy set* later). In many applications of FL to engineering problems, we are interested in mappings from numbers into numbers, and not sets into sets. Consequently, our problem is more difficult than the usual FL problem. We have to add a front-end “fuzzifier” and a rear-end “defuzzifier” to the usual FL model. The result is a fuzzy logic system (FLS). *The purpose of this tutorial paper is to provide the reader with a guided tour through those parts of the FL literature that I believe are necessary in order to synthesize a FLS.*

There is a huge amount of literature on FL. A lot of it is theoretical, and is presently not utilized when FL is applied to engineering problems. This is analogous to the dichotomy between the pure and applied mathematics literatures and their use in engineering. Of course, we must always be on the lookout for important new theoretical ideas and results in FL theory that may lead to new and practical solutions to engineering problems.



(a)



(b)

Fig. 1. (a) Mackey-Glass chaotic time series and its single-step prediction. The latter was obtained using a FLS forecaster that predicts one step ahead using a window of 10 past measurements. The FLS forecaster was trained using the first 500 samples of the time series. (b) Baselineing the FLS forecaster against the most naive forecaster, the zero-order hold, which predicts the next value as the present value.

One danger in providing a guided tour is that it is only possible to reference a smattering of the huge FL literature (e.g., Maier and Sherif [48] provide 450 references), and this may offend some of the contributors to that literature. Even worse, it is not even possible to reference all of the works that contributors to the subfield of FLS's will feel should be referenced (especially if it is their work). To all of those contributors who feel offended by omission, the author sincerely apologizes.

C. What is a Fuzzy Logic System?

In general, a FLS is a nonlinear mapping of an input data (feature) vector into a scalar output (the vector output case decomposes into a collection of independent multi-input/single-output systems). The richness of FL is that there are enormous numbers of possibilities that lead to lots of different mappings. This richness does require a careful understanding of FL and the elements that comprise a FLS. One can, of course, challenge the validity of some of these possibilities. To me, this is analogous to the representation problem that we always face in engineering,

i.e., do we use a linear model, time-domain or frequency-domain model, lumped-parameter or distributed parameter model, state-space or input-output model, deterministic or random model, etc.? Once we agree on the representation, then we can proceed. The same is true about FL.

In this paper, we show how to interpret the nonlinear mapping of a FLS geometrically, as is commonly done in the FL controls literature, and also how to write a detailed formula for its input-output relationship. The latter lets us analyze a FLS, develop training algorithms for them, and write computer programs that incorporate the FLS into specific applications.

D. Potential of FLS's

One of the things we will cover in this paper is how time-series forecasting can be accomplished using a FLS. Our nonlinear FLS forecaster uses a small window of past measurements to forecast the next value of the time series. Fig. 1 depicts almost perfect one-step ahead forecasting of a chaotic Mackey-Glass time series [41], [47]. It is impossible to discern a difference between the actual

Table 1 Engineering Terms Whose Contextual Usage Is Usually Quite Fuzzy

| Term | Contextual Usage |
|-------------|---|
| Alias | none, a bit, high |
| Bandwidth | narrowband, broadband |
| Blur | somewhat, quite, very |
| Correlation | low, medium, high, perfect |
| Errors | large, medium, small, a lot of, not so great, very large, very small, almost zero |
| Frequency | high, low, ultra-high |
| Resolution | low, high |
| Sampling | low-rate, medium-rate, high-rate, very high-rate |
| Stability | stable (lightly damped, highly damped, over damped, critically damped), unstable |

and predicted time series in Fig. 1(a). Fig. 1(b) baselines the FLS forecaster against the most naive forecaster, the zero-order hold, which predicts the next value as the present value. If a new forecaster cannot do better than the zero-order hold forecaster, it is worthless. Hopefully, this example demonstrates the potential of a FLS.

E. Rationale for FL in Engineering

The first paper on FL, which is now considered to be the seminal paper on FL, was written by Lotfi Zadeh [84], who is also considered to be the founding father of the entire field of FL. In this paper, Zadeh states "... the fact remains that ... imprecisely defined 'classes' play an important role in human thinking, particularly in the domains of pattern recognition, communication of information and abstraction." In another important paper [85], we find the following, often quoted, *Principle of Incompatibility*, which many take as a rationale for the study of FL in engineering and other disciplines: "As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics"—or, "The closer one looks at a real-world problem, the fuzzier becomes its solution."

F. Fuzzy Concepts in Engineering

Some people object, on principle, to using FL instead of a more familiar model-based approach to design. To dispel the notion of crispness (i.e., dual-valued concepts, which either are true or are not true), I list a collection of terms (see Table 1) that are widely used in control, signal processing and communications. While we frequently strive for crisp values of these terms, we usually use them in

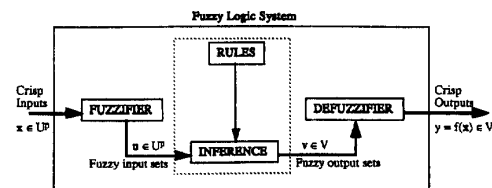


Fig. 2. Fuzzy logic system (FLS). The elements of the FLS are explained later.

fuzzy contexts, where they actually convey more useful information than would a crisp value.

Correlation is an interesting example, because it can be defined mathematically so that, for a given set of data, we can compute a crisp number for it. Let's assume that correlation has been normalized so that it can range between zero and unity, and that for a given set of data we compute the correlation value as 0.15. When explaining the amount of data correlation to someone else, it is usually more meaningful to explain it as "this data has low correlation." When we do this, we are actually fuzzifying the crisp value of 0.15 into the fuzzy set "low correlation."

Stability is another very interesting example. A system either is or is not stable; there is nothing fuzzy about this. However, if the system is stable, we frequently describe its degree of relative stability, using any of the terms listed in Table 1. These terms may be more meaningful than the following description: "The system has four complex poles and the effective damping ratio for the system is 0.3." We just describe the response of such a system as "lightly damped." Once again, we are fuzzifying the crisp value of 0.3 into the fuzzy set "lightly damped."

For interesting historical perspectives on FL, including its earlier origins (when it was called continuous-valued logic) see [50], and, for philosophical interpretations of FL, see [38].

G. Fuzzy Logic System: A High-Level Introduction

Fig. 2 depicts a FLS that is widely used in fuzzy logic controllers and signal processing applications. A FLS maps crisp inputs into crisp outputs. It contains four components: rules, fuzzifier, inference engine, and defuzzifier. Once the rules have been established, a FLS can be viewed as a mapping from inputs to outputs (the solid path in Fig. 2, from "Crisp Inputs" to "Crisp Outputs"), and this mapping can be expressed quantitatively as $y = f(\mathbf{x})$. One of the major goals of this paper is to obtain explicit formulas for the nonlinear mapping between \mathbf{x} and y .

Rules may be provided by experts (you may be such a person) or can be extracted from numerical data. In either case, *engineering rules* are expressed as a collection of IF-THEN statements, e.g., "IF u_1 is very warm and u_2 is quite low, THEN turn v somewhat to the right." This one rule reveals that we will need an understanding of: 1) linguistic variables versus numerical values of a variable (e.g., *very warm* versus 36°C); 2) quantifying linguistic variables (e.g.,

u_1 may have a finite number of linguistic terms associated with it, ranging from *extremely hot* to *extremely cold*), which is done using *fuzzy membership functions*; 3) logical connections for linguistic variables (e.g., “and,” “or,” etc.); and 4) implications, i.e., “IF A THEN B.” Additionally, we will need to understand how to combine more than one rule.

The *fuzzifier* maps crisp numbers into fuzzy sets. It is needed in order to activate rules which are in terms of linguistic variables, which have fuzzy sets associated with them.

The *inference engine* of the FLS maps fuzzy sets into fuzzy sets. It handles the way in which rules are combined. Just as we humans use many different types of inferential procedures to help us understand things or to make decisions, there are many different fuzzy logic inferential procedures. Only a very small number of them are actually being used in engineering applications of FL.

In many applications, crisp numbers must be obtained at the output of a FLS. The *defuzzifier* maps output sets into crisp numbers. In a controls application, for example, such a number corresponds to a control action. In a signal processing application, such a number could correspond to the prediction of next year’s sunspot activity, a financial forecast, or the location of a target.

H. Applications of Fuzzy Logic

A short list of applications of FL includes: *Controls Applications*—aircraft control (Rockwell Corp.), Sendai subway operation (Hitachi), cruise control (Nissan), automatic transmission (Nissan, Subaru), self-parking model car (Tokyo Tech. Univ.), and space shuttle docking (NASA); *Scheduling and Optimization*—elevator scheduling (Hitachi, Fujitech, Mitsubishi) and stock market analysis (Yamaichi Securities); and *Signal Analysis for Tuning and Interpretation*—TV picture adjustment (Sony), handwriting recognition (Sony Palm Top), video camera autofocus (Sanyo/Fisher, Canon) and video image stabilizer (Matshushita/Panasonic). For many additional applications, see [38], [44], [61], [69], and [82].

I. Fuzzy Logic and Probability

Some people maintain that there is no difference between FL and probability. When I am asked about this, often at the beginning of a lecture or course on FL, I ask the following question: “How many of you have had a formal course on probability?” Usually, all hands go up. Then I ask “How many of you have had a formal course on FL?” Usually, no hands, or only a very small number of hands go up. I then state that in order to explain the differences between FL and probability, we must first spend time formally understanding FL. Only then can we make intelligent comparisons between that which we understand (probability) and that which we will understand (FL). Consequently, we now turn to short primers on fuzzy sets and FL, after which we shall return to probability versus fuzzy logic.

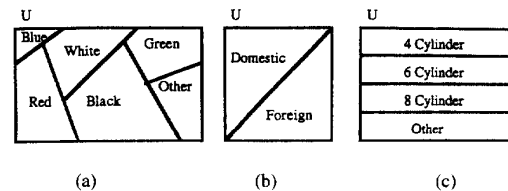


Fig. 3. Partitioning of the set of all automobiles in New York City into subsets by: (a) color, (b) domestic or foreign, and (c) number of cylinders.

II. SHORT PRIMER ON FUZZY SETS

A. Crisp Sets

Recall that a *crisp set* A in a universe of discourse U (which provides the set of allowable values for a variable) can be defined by listing all of its members or by identifying the elements $x \in A$. One way to do the latter is to specify a condition by which $x \in A$; thus A can be defined as $A = \{x \mid x \text{ meets some condition}\}$. Alternatively, we can introduce a zero-one *membership function* (also called a characteristic function, discrimination function, or indicator function) for A , denoted $\mu_A(x)$, such that $A \Rightarrow \mu_A(x) = 1$ if $x \in A$ and $\mu_A(x) = 0$ if $x \notin A$. Subset A is mathematically equivalent to its membership function $\mu_A(x)$ in the sense that knowing $\mu_A(x)$ is the same as knowing A itself.

Example 1: Consider the set of all automobiles in New York City; this is U . The elements of U are individual cars; but, there are many different types of subsets that can be established for U , including the three that are depicted in Fig. 3. Either a car has or does not have six cylinders. This is a very crisp requirement. Hence, if our car has four cylinders, its membership function value for the subset of four cylinder cars is unity, whereas its membership function value for the subset of six cylinder cars is zero.

B. Fuzzy Sets

A *fuzzy set* F defined on a universe of discourse U is characterized by a *membership function* $\mu_F(x)$ which takes on values in the interval $[0, 1]$. A fuzzy set is a generalization of an ordinary subset (i.e., a crisp subset) whose membership function only takes on two values, zero or unity. A membership function provides a *measure of the degree of similarity* of an element in U to the fuzzy subset.

Example 1 (Contd.): A car can be viewed as “domestic” or “foreign” from different perspectives. One perspective is that a car is domestic if it carries the name of a USA auto manufacturer; otherwise it is foreign. There is nothing fuzzy about this perspective; however, many people today feel that the distinction between a domestic and foreign automobile is not as crisp as it once was, because many of the components for what we consider to be domestic cars (e.g., Ford, GM, and Chrysler) are produced outside of the USA. Additionally, some “foreign” cars are manufactured here in the USA. Consequently, one could think of the membership functions for domestic and foreign

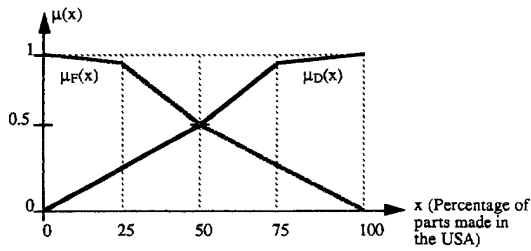


Fig. 4. Membership functions for domestic and foreign cars, based on the percentage of parts in the car made in the USA.

cars looking like $\mu_D(x)$ and $\mu_F(x)$ depicted in Fig. 4. Observe that a specific car (located along the horizontal axis by determining the percentage of its parts made in the USA) exists in both subsets simultaneously—domestic cars and foreign cars—but to different degrees of membership. For example, if our car has 75% of its parts made in the USA, then $\mu_D(75\%)=0.9$ and $\mu_F(75\%)=0.25$. Ultimately, we would describe our car as domestic. In fact, when we do this, we decide on the subset by choosing it to be associated with the maximum of $\mu_D(75\%)=0.9$ and $\mu_F(75\%)=0.25$. What do we call a car that has exactly 50% of its parts made in the USA and abroad (see [38] for discussions on maximum fuzziness and paradoxes which occur when membership value equals 0.5)?

Describing a car by its color is also not a crisp description, because each color has different shades associated with it.

The main point of this example is to demonstrate that in fuzzy logic an element can reside in more than one set to different degrees of similarity. This can not occur in crisp set theory. \square

A fuzzy set F in U may be represented as a set of ordered pairs of a generic element x and its grade of membership function: $F = \{(x, \mu_F(x)) \mid x \in U\}$. When U is continuous (e.g., the real numbers), F is commonly written as $F = \int_U \mu_F(x)/x$. In this equation, the integral sign does not denote integration; it denotes the collection of all points $x \in U$ with associated membership function $\mu_F(x)$. When U is discrete, F is commonly written as $F = \sum_U \mu_F(x)/x$. In this equation, the summation sign does not denote arithmetic addition; it denotes the collection of all points $x \in U$ with associated membership function $\mu_F(x)$; hence, it denotes the set theoretic operation of union. The slash in these expressions associates the elements in U with their membership grades, where $\mu_F(x) > 0$.

Example 2: [87] Let $F =$ integers close to 10; then $F = 0.1/7 + 0.5/8 + 0.8/9 + 1/10 + 0.8/11 + 0.5/12 + 0.1/13$. Three points to note from F are: 1) the integers not explicitly shown all have membership functions equal to zero; by convention, we do not list such elements; 2) the values for the membership functions were chosen by a specific individual; except for the unity membership value when $x = 10$, they can be modified based on our own personal interpretation of the phrase “close”; and 3) the

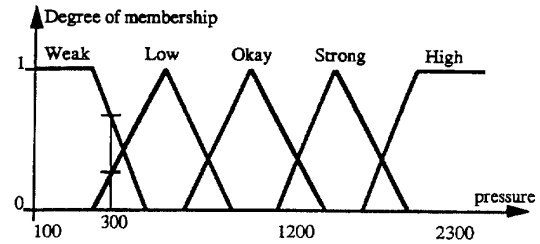


Fig. 5. Membership functions for $T(\text{pressure}) = \{\text{weak, low, okay, strong, high}\}$. The shapes of the membership functions as well as their degree of overlap are quite arbitrary. (© 1992 IEEE. This figure has been taken from E. Cox, “Fuzzy Fundamentals,” *IEEE Spectrum*, pp. 58–61, Oct. 1992.)

membership function is symmetric about $x = 10$, because there is no reason to believe that integers to the left of 10 are close to 10 in a different way than are integers to the right of 10; but, again, we are free to make other interpretations.

C. Linguistic Variables

Zadeh [86] states “In retreating from precision in the face of overpowering complexity, it is natural to explore the use of what might be called *linguistic* variables, that is, variables whose values are not numbers but words or sentences in a natural or artificial language. . . . The motivation of the use of words or sentences rather than numbers is that linguistic characterizations are, in general, less specific than numerical ones.”

Let u denote the name of a linguistic variable (e.g., temperature). Numerical values of a linguistic variable u are denoted x , where $x \in U$. Sometimes x and u are used interchangeably, especially when a linguistic variable is a letter, as is sometimes the case in engineering applications. A linguistic variable is usually decomposed into a set of terms, $T(u)$, which cover its universe of discourse.

Example 3: [13] Let *pressure* (u) be interpreted as a linguistic variable. It can be decomposed into the following set of terms: $T(\text{pressure}) = \{\text{weak, low, okay, strong, high}\}$, where each term in $T(\text{pressure})$ is characterized by a fuzzy set in the universe of discourse $U = [100 \text{ psi}, 2300 \text{ psi}]$. We might interpret *weak* as a pressure below 200 psi, *low* as a pressure close to 700 psi, *okay* as a pressure close to 1050 psi, *strong* as a pressure close to 1500 psi, and *high* as a pressure above 2200 psi. These terms can be characterized as fuzzy sets whose membership functions are shown in Fig. 5. Measured values of pressure (x) lie along the pressure axis. In this example, a vertical line from any measured value intersects at most two membership functions. So, for example, $x = 300$ resides in the fuzzy sets *weak pressure* and *low pressure*, to different degrees of similarity.

D. Membership Functions

In engineering applications of fuzzy logic, *membership functions*, $\mu_F(x)$, are, for the most part, associated with

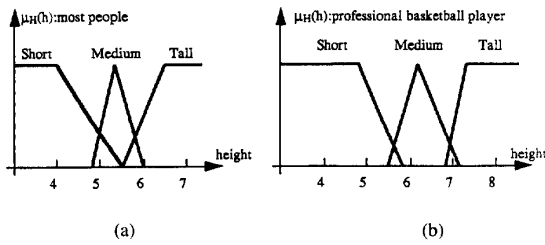


Fig. 6. Membership functions for $T(\text{height}) = \{\text{short men, medium men, tall men}\}$. (a) Most people's membership functions, and (b) professional basketball player's membership functions.

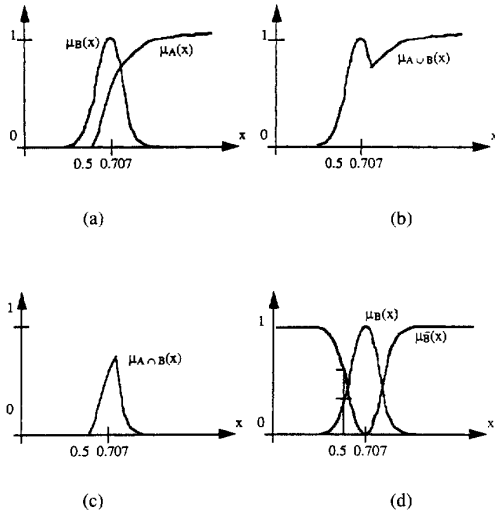


Fig. 7. Membership functions associated with $A =$ damping ratio u considerably larger than 0.5, and $B =$ damping ratio u approximately 0.707. (a) $\mu_A(x)$ and $\mu_B(x)$, (b) $\mu_{A \cup B}(x)$, (c) $\mu_{A \cap B}(x)$, and (d) $\mu_{\bar{A}}(x)$. (Adapted from Figs. 2 and 3 in [87]).

terms that appear in the antecedents or consequents of rules, or in phrases (e.g., *foreign cars*).

Example 4: Some examples of rules and associated membership functions (shown in brackets) are: 1) IF we are tracking a target at one instant of time, THEN the target will not be *too far away* at the next instant of time [$\mu_{\text{TOO-FAR-AWAY}}(x)$]; 2) IF the horizontal position is *medium positive* and the angular position is *small negative*, THEN the control angle is *large positive* [$\mu_{\text{MEDIUM-POSITIVE}}(x)$, $\mu_{\text{SMALL-NEGATIVE}}(\theta)$, $\mu_{\text{LARGE-POSITIVE}}(\phi)$]; and, 3) IF $y(t)$ is *close to 0.5*, THEN $f(y)$ is *close to zero* [$\mu_{\text{CLOSE-TO-0.5}}(y)$, $\mu_{\text{CLOSE-TO-ZERO}}(f(y))$]. \square

The most commonly used shapes for membership functions are triangular, trapezoidal, piecewise linear and Gaussian. Until very recently, membership functions were chosen by the user arbitrarily, based on the user's experience; hence, the membership functions for two users could be quite different depending upon their experiences, perspectives, cultures, etc. More recently, membership functions

have been designed using optimization procedures (e.g., [23], [27], [75], and [77]).

Example 5: Let U be the set of all men. The term "height" can mean different things to different people. Fig. 7 depicts two sets of membership functions for the set of terms $\{\text{short men, medium men, tall men}\}$. Clearly, the terms *short men*, *medium men*, and *tall men* will have a very different meaning for a professional basketball player than they will for most other people. This illustrates the fact that membership functions can be quite context dependent. \square

The number of membership functions is up to us. Greater resolution is achieved by using more membership functions at the price of greater computational complexity. Membership functions don't have to overlap; but, one of the great strengths of FL is that membership functions can be made to overlap. This expresses the fact that "the glass can be partially full and partially empty at the same time." In this way we are able to distribute our decisions over more than one input class, which helps to make FL systems robust. Although membership functions do not have to be scaled between zero and unity, most people do this so that variables are normalized. We can always normalize a fuzzy set by dividing $\mu_F(x)$ by its largest value $\sup_x \mu_F(x)$.

E. Some Terminology

The *support* of a fuzzy set F is the crisp set of all points x in U such that $\mu_F(x) > 0$. For example, the support of the fuzzy set *short* in Fig. 6(a) is $x \in [0, 5.5]$. The element x in U at which $\mu_F(x) = 0.5$ is called the *crossover point*. A fuzzy set whose support is a single point in U with $\mu_F(x) = 1$, is called a *fuzzy singleton*.

F. Set Theoretic Operations

1) *Crisp Sets:* Now that we have defined fuzzy sets, *what can we do with them?* We could ask the same question about crisp sets, and we know that there are lots of things we can do with them; hence, we expect that we can do analogous things with fuzzy sets. To begin, let us briefly review the elementary crisp-set operations of union, intersection, and complement.

Let A and B be two subsets of U . The *union* of A and B , denoted $A \cup B$, contains all of the elements in either A or B , i.e., $\mu_{A \cup B}(x) = 1$ if $x \in A$ or $x \in B$, and $\mu_{A \cup B}(x) = 0$ if $x \notin A$ and $x \notin B$. The *intersection* of A and B , denoted $A \cap B$, contains all of the elements that are simultaneously in A and B , i.e., $\mu_{A \cap B}(x) = 1$ if $x \in A$ and $x \in B$, and $\mu_{A \cap B}(x) = 0$ if $x \notin A$ or $x \notin B$. Let \bar{A} denote the *complement* of A ; it contains all the elements that are not in A , i.e., $\mu_{\bar{A}}(x) = 1$ if $x \notin A$ and $\mu_{\bar{A}}(x) = 0$ if $x \in A$. From these facts, it is easy to show that:

$$A \cup B \Rightarrow \mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (1a)$$

$$A \cap B \Rightarrow \mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad (1b)$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x). \quad (1c)$$

For example, $x \in A$ or $x \in B$ means $(\mu_A(x) = 1, \mu_B(x) = 1)$, $(\mu_A(x) = 1, \mu_B(x) = 0)$, or $(\mu_A(x) =$

$0, \mu_B(x) = 1$), for which $\max[\mu_A(x), \mu_B(x)] = 1$; and, $x \notin A$ and $x \notin B$ means ($\mu_A(x) = 0, \mu_B(x) = 0$) for which $\max[\mu_A(x), \mu_B(x)] = 0$. Consequently, $\max[\mu_A(x), \mu_B(x)]$ does provide the correct membership function for union.

The formulas for $\mu_{A \cup B}(x)$, $\mu_{A \cap B}(x)$, and $\mu_{\bar{A}}(x)$ are very useful for proving other theoretical properties about sets. Note, also, that “max” and “min” are not the only ways to describe $\mu_{A \cup B}(x)$ and $\mu_{A \cap B}(x)$. While these formulas are not usually part of conventional set theory, they are essential to fuzzy set theory; however, as we have just demonstrated, they really do occur in conventional set theory. See [33] and [81] for other ways to characterize these operations.

The crisp union and intersection operations are *commutative* (e.g., $A \cup B = B \cup A$), *associative* (e.g., $A \cup (B \cup C) = (A \cup B) \cup C = A \cup (B \cup C)$) and *distributive* [e.g., $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$]. These properties can be proved either by Venn diagrams or by using the operations involving membership functions given in (1).

De Morgan’s laws for crisp sets are: $\overline{A \cup B} = \bar{A} \cap \bar{B}$ and $\overline{A \cap B} = \bar{A} \cup \bar{B}$. These laws, which are also very useful in proving things about more complicated operations on sets, can also be proved either by Venn diagrams or by using the operations involving membership functions given in (1).

The two fundamental (Aristotelian) laws of crisp set theory are: 1) *Law of Contradiction*: $A \cup \bar{A} = U$ (i.e., a set and its complement must comprise the universe of discourse), and 2) *Law of Excluded Middle*: $A \cap \bar{A} = \phi$ (i.e., an element can either be in its set or its complement; it cannot simultaneously be in both).

2) *Fuzzy Sets*: In FL, union, intersection and complement are defined in terms of their membership functions. Let fuzzy sets A and B be described by their membership functions $\mu_A(x)$ and $\mu_B(x)$. One definition of *fuzzy union* leads to the membership function

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (2a)$$

and one definition of *fuzzy intersection* leads to the membership function

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]. \quad (2b)$$

Additionally, the membership function for *fuzzy complement* is

$$\mu_{\bar{B}}(x) = 1 - \mu_B(x). \quad (2c)$$

Obviously, these three definitions were motivated by their crisp counterparts, in (1).

Although (2) and (1) look exactly alike, we must remember that: (1) sets A and B in (2) are fuzzy, whereas in (1) they are crisp; and, (2) fuzzy sets can *only* be characterized by their membership functions, whereas crisp sets can be characterized either by their membership functions, or a description of their elements, or a listing of their elements.

Example 6: Consider the fuzzy sets $A =$ damping ratio u considerably larger than 0.5, and $B =$ damping ratio u approximately 0.707. Note that damping ratio is a positive real number. Consequently, $A = \{(x, \mu_A(x)) \mid x \in U\}$, and $B = \{(x, \mu_B(x)) \mid x \in U\}$, where, for example, $\mu_A(x)$ and $\mu_B(x)$ are specified, as [87]: $\mu_A(x) = [0, x \leq 0.5]$ or $[1/(1 + (x - 0.5)^{-2}), x > 0.5]$, and $\mu_B(x) = 1/(1 + (x - 0.707)^4)$, $x > 0$. Fig. 7 depicts $\mu_A(x), \mu_B(x), \mu_{A \cup B}(x), \mu_{A \cap B}(x)$, and $\mu_{\bar{B}}(x)$. Observe, from Fig. 7(d), that the point $x = 0.5$ exists in both B and \bar{B} simultaneously, but to different degrees, because $\mu_B(0.5) \neq 0$ and $\mu_{\bar{B}}(0.5) \neq 0$. \square

This example demonstrates that for fuzzy sets the Laws of Contradiction and Excluded Middle are broken, i.e., for fuzzy sets: $A \cup \bar{A} \neq U$ and $A \cap \bar{A} \neq \phi$. We have also seen this in the automobile Example 1. In fact, one of the ways to describe the difference between crisp set theory and fuzzy set theory is to explain that these two laws do not hold in fuzzy set theory. Consequently, any other mathematics that relies on crisp set theory, such as (frequency-based) probability, must be different from fuzzy set theory.

The “max” and “min” operators are not the only ones that could have been chosen to model fuzzy union and fuzzy intersection. Zadeh, in his pioneering first paper [84], defined two operators each for fuzzy union and fuzzy intersection, namely: *fuzzy union*—maximum and algebraic sum [$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$]; *fuzzy intersection*—minimum and algebraic product [$\mu_{A \cap B}(x) = \mu_A(x)\mu_B(x)$]. Later, other operators, which have an axiomatic basis, were introduced— t -conorm operators for fuzzy union also known as an s -norm, and denoted \oplus), and t -norm operators for fuzzy intersection (denoted \star). See [80] for precise axiomatic-based definitions of both operators. Some other examples of t -conorms are: *bounded sum*: $x \oplus y = \min(1, x + y)$; and, *drastic sum*: $x \oplus y = x$ if $y = 0, y$ if $x = 0, 1$ if $x, y, > 0$. Some other examples of t -norms are: *bounded product*: $x \star y = \max[0, x + y - 1]$; and, *drastic product*: $x \star y = x$ if $y = 1, y$ if $x = 1, 0$ if $x, y < 1$. There is even an axiomatic definition for the complement of a fuzzy set (denoted c) [33, pp. 38–45]. In engineering applications, most people use the fuzzy complement whose membership function is given in (2c).

As pointed out by Zimmerman [87], pairs of t -norms and t -conorms satisfy the following generalization of DeMorgan’s laws [7]: $s[\mu_A(x), \mu_B(x)] = c\{t[c(\mu_A(x)), c(\mu_B(x))]\}$, and $t[\mu_A(x), \mu_B(x)] = c\{s[c(\mu_A(x)), c(\mu_B(x))]\}$, where $x \in U$. For example, $\max[\mu_A(x), \mu_B(x)] = 1 - \min[1 - \mu_A(x), 1 - \mu_B(x)]$ and $\min[\mu_A(x), \mu_B(x)] = 1 - \max[1 - \mu_A(x), 1 - \mu_B(x)]$.

Note, also, that there are other ways of combining fuzzy sets, e.g., the *fuzzy and*, *fuzzy or*, *compensatory and*, and *compensatory or*; [87], [81].

The different t -norms, t -conorms and complements, that are available to us in fuzzy set theory, provide us with a plethora of richness and also with some (tough) choices that will have to be made in our FLS. Zimmerman [87, pp. 42–43] provides eight criteria which might be helpful

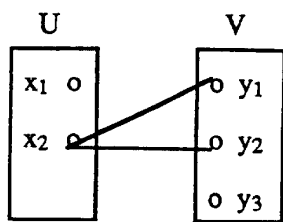


Fig. 8. Sagittal diagram for relation of stability between the set of all linear, second-order continuous-time systems and the set of the poles of such systems.

in selecting the connective's operator. Unfortunately, this author found most of these criteria to be so subjective that he could not use them in his engineering applications. *Most engineering applications of fuzzy sets use: 1) the min or algebraic product t-norm for fuzzy intersection; 2) the max t-conorm for fuzzy union; and, 3) $1 - \mu_A(x)$ for the membership function of the fuzzy complement.* Finally, we note that all of the operators that are available for fuzzy union, intersection and complement reduce to their dual-logic counterparts when the membership functions are restricted to the values 0 or 1.

G. Relations and Compositions on the Same Product Spaces

1) *Crisp Relations:* "A crisp relation represents the presence or absence of association, interaction, or interconnectiveness between the elements of two or more sets" [33, p. 65]. Here we limit our attention to relations between two sets U and V , i.e., to binary relations denoted $R(U, V)$. For example, let R represent the relation of stability between the set of all linear, second-order continuous-time systems and the set of the poles of such systems. Of all the possible pairings of linear second-order continuous-time systems, and poles, we know that only those pairs whose members are time-invariant and have poles lying either in the left-half of the complex s -plane or on the imaginary axis of that plane are stable.

We let $U \times V$ denote the Cartesian product of the two crisp sets U and V , i.e., $U \times V = \{(x, y) \mid x \in U \text{ and } y \in V\}$. $R(U, V)$ is a subset of $U \times V$, which should be very clear from the example of stability just given; however, let us elaborate on this example to make it even clearer. Let $U = \{x_1, x_2\} = \{\text{linear second-order time-varying continuous-time system, linear second-order time-invariant continuous-time system}\}$ and $V = \{y_1, y_2, y_3\} = \{\text{poles lie in the left-half } s\text{-plane, poles lie on the } j\omega\text{-axis, poles lie in the right-half } s\text{-plane}\}$. The cartesian product $U \times V$ can be visualized as a 2×3 array of ordered pairs, e.g., the (1-2) element is (linear second-order time-varying continuous-time system, poles lie on the $j\omega$ -axis). Clearly, our stability relation $R(U, V)$ is the following subset of $U \times V$: $R(U, V) = \{(\text{linear second-order time-invariant continuous-time system, poles lie in the left-half } s\text{-plane}), (\text{linear second-order time-invariant continuous-time system, poles lie on the } j\omega\text{-axis})\}$. Because a relation is itself a set,

all of the basic crisp set operations can be applied to it without modifications.

Crisp relation $R(U, V)$ can be defined by the following membership function:

$$\mu_R(x, y) = \begin{cases} 1 & \text{if and only if } (x, y) \in R(U, V) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

For binary relations defined over a cartesian product whose elements come from a discrete universe of discourse, it is convenient to collect the membership functions into a *relational matrix* whose elements are either zero or unity. The relational matrix for our stability relation is

$$\begin{matrix} & y_1 & y_2 & y_3 \\ x_1 & \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \\ x_2 & \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

An equivalent representation for a binary relation is a *sagittal diagram*, in which the sets U, V are each represented by a set of nodes in the diagram that are clearly distinguished from one another. Elements of $U \times V$ with nonzero membership grade in $R(U, V)$ are represented in the diagram by lines connecting the respective nodes. Although not explicitly shown, the lines have membership values equal to unity. The sagittal diagram for our stability relation is depicted in Fig. 8.

2) *Fuzzy Relations:* Fuzzy relations represent a *degree* of presence or absence of association, interaction, or interconnectiveness between the elements of two or more fuzzy sets. Some examples of binary fuzzy relations are: x is much larger than y , y is very close to x , z is much greener than y , system 1 is less damped than system 2, bandwidth of system A is larger than that of system B , and, Tone C is of higher local signal-to-noise ratio than tone D . Fuzzy relations play an important role in a FLS.

Let U and V be two universes of discourse. A fuzzy relation, $R(U, V)$ is a fuzzy set in the product space $U \times V$, i.e., it is a fuzzy subset of $U \times V$, and is characterized by membership function $\mu_R(x, y)$ where $x \in U$ and $y \in V$, i.e., $R(U, V) = \{(x, y), \mu_R(x, y) \mid (x, y) \in U \times V\}$. The difference between a fuzzy relation and a crisp relation is that for the former $\mu_R(x, y) \in [0, 1]$, whereas for the latter $\mu_R(x, y) = 0$ or 1. The generalization of a fuzzy relation to an n -dimensional cartesian product space is straightforward.

Example 7: Let U and V be the real numbers, and consider the fuzzy relation "target x is close to target y ." *Close* is sometimes referred to as a "root concept." Here is one membership function for this relation: $\mu_{\text{CLOSE}}(|x - y|) = \max[(5|x - y|)/5, 0]$. This relational membership function is depicted in Fig. 9. Note that the distance between the two targets $|x - y|$ is treated as the independent variable, which makes it possible to view the membership function on a two-dimensional plot. \square

Because fuzzy relations are fuzzy sets in product space, set theoretic and algebraic operations can be defined for them using our earlier operators for fuzzy union, intersec-

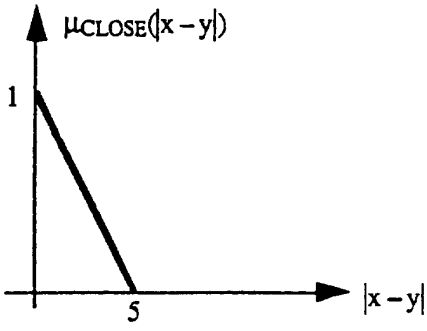


Fig. 9. Relational membership function $\mu_{\text{CLOSE}}(|x-y|)$.

tion and complement. Let $R(x, y)$ and $S(x, y)$ (shortened in the sequel to R and S) be two fuzzy relations in the same product space $U \times V$. The intersection and union of R and S , which are compositions of the two relations, are then defined as

$$\mu_{R \cap S}(x, y) = \mu_R(x, y) \star \mu_S(x, y) \quad (4a)$$

$$\mu_{R \cup S}(x, y) = \mu_R(x, y) \oplus \mu_S(x, y) \quad (4b)$$

where \star is any t -norm, and \oplus is any t -conorm.

Example 8: Consider the sentence “ x is much larger than y and y is very close to x ,” a sentence which we know is implausible. We wish to establish a fuzzy membership function for it. We begin by recognizing that this sentence is a composition between the two relations “ x is much larger than y ” and “ y is very close to x ,” and that both relations live on the same product space $U \times V$. We then create membership functions for the relations “ x is much larger than y ,” $\mu_{\text{ML}}(x, y)$, and “ y is very close to x ,” $\mu_{\text{VC}}(y, x)$. Finally, using these membership functions and an appropriate t -norm (e.g., \min), we can create the membership function for the sentence, as $\mu_{\text{ML} \cap \text{VC}}(x, y) = \min[\mu_{\text{ML}}(x, y), \mu_{\text{VC}}(y, x)]$.

As a concrete example of this procedure, let $U = \{x_1, x_2, x_3\}$ and $V = \{y_1, y_2, y_3, y_4\}$. The membership functions $\mu_{\text{ML}}(x, y)$ and $\mu_{\text{VC}}(y, x)$ are assumed to be given by the following relational matrices [87]:

$$\mu_{\text{ML}}(x, y) = \begin{matrix} & \begin{matrix} y_1 & y_2 & y_3 & y_4 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 0.8 & 1 & 0.1 & 0.7 \\ 0 & 0.8 & 0 & 0 \\ 0.9 & 1 & 0.7 & 0.8 \end{pmatrix} \end{matrix}$$

and

$$\mu_{\text{VC}}(y, x) = \begin{matrix} & \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{matrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{matrix} & \begin{pmatrix} 0.4 & 0.9 & 0.3 \\ 0 & 0.4 & 0 \\ 0.9 & 0.5 & 0.8 \\ 0.6 & 0.7 & 0.5 \end{pmatrix} \end{matrix} \quad (5a)$$

Then, for example, $\mu_{\text{ML} \cap \text{VC}}(x_3, y_4) = \min[\mu_{\text{ML}}(x_3, y_4), \mu_{\text{VC}}(y_4, x_3)] = \min(0.8, 0.5) = 0.5$. The complete mem-

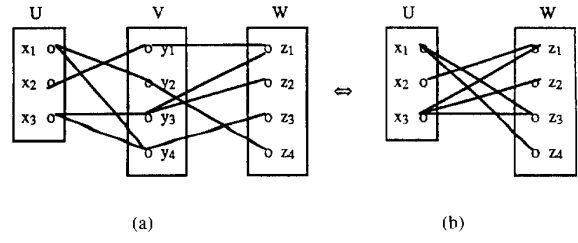


Fig. 10. Sagittal diagrams for Example 9; (a) original diagram for relations $R_1(U, V)$ and $R_2(V, W)$; (b) compositional diagram for $R_3(U, W)$.

bership function, $\mu_{\text{ML} \cap \text{VC}}(x, y)$, can similarly be shown to be given by

$$\mu_{\text{ML} \cap \text{VC}}(x, y) = \begin{matrix} & \begin{matrix} y_1 & y_2 & y_3 & y_4 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 0.4 & 0 & 0.1 & 0.6 \\ 0 & 0.4 & 0 & 0 \\ 0.3 & 0 & 0.7 & 0.5 \end{pmatrix} \end{matrix}. \quad (5b)$$

Observe from (5b) that, because most of its elements are less than 0.5, this sentence is treated with a high degree of disbelief, except, perhaps at (x_1, y_4) and (x_3, y_3) . \square

H. Relations and Compositions on Different Product Spaces

1) *Crisp Compositions:* Next, we consider the composition of crisp relations from different product spaces that share a common set, namely $P(U, V)$ and $Q(V, W)$. Klir and Folger [33, p. 75] state “The composition of these two relations is denoted by

$$R(U, W) = P(U, V) \circ Q(V, W) \quad (6)$$

and is defined as a subset $R(U, W)$ of $U \times W$ such that $(x, w) \in R$ if and only if there exists at least one $y \in V$ such that $(x, y) \in P$ and $(y, w) \in Q$.”

Example 9: Here we begin with the sagittal diagram depicted in Fig. 10, from which we conclude that the relational matrices $R_1(U, V)$, $R_2(V, W)$, and $R_3(U, W)$ are:

$$\begin{matrix} & \begin{matrix} y_1 & y_2 & y_3 & y_4 \end{matrix} \\ R_1(U, V) = \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} & \begin{matrix} z_1 & z_2 & z_3 & z_4 \end{matrix} \\ R_2(V, W) = \begin{matrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} & \begin{matrix} z_1 & z_2 & z_3 & z_4 \end{matrix} \\ R_3(U, W) = \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{matrix} \quad (7)$$

Because it is not efficient to keep describing our compositions in terms of sagittal diagrams, we need a formula that conveys the same information. \square

Definitions: The max-min composition of relations $P(U, V)$ and $Q(V, W)$, is defined by the membership function $\mu_{P \circ Q}(x, z)$, where

$$\mu_{P \circ Q}(x, z) = \{(x, z), \max_y [\min(\mu_P(x, y), \mu_Q(y, z))]\}. \quad (8)$$

The max-product composition of relations $P(U, V)$ and $Q(V, W)$, is defined by the membership function $\mu_{P \times Q}(x, z)$, where

$$\mu_{P \times Q}(x, z) = \{(x, z), \max_y [\mu_P(x, y)\mu_Q(y, z)]\}. \quad (9)$$

It is a fact that carrying out the operations of either the max-min or max-product compositions leads to the correct relational matrix $R(U, W)$.

Example 10: Let's verify (8) and (9) for the 1-2 element of $R_3(U, W)$ in (7). For this element, (8) becomes $\mu_{R_3}(x_1, z_2) = \{(x_1, z_2), \max_y [\min(\mu_{R_1}(x_1, y), \mu_{R_2}(y, z_2))]\} = \{(x_1, z_2), \max[\min(\mu_{R_1}(x_1, y_1), \mu_{R_2}(y_1, z_2)), \min(\mu_{R_1}(x_1, y_2), \mu_{R_2}(y_2, z_2)), \min(\mu_{R_1}(x_1, y_3), \mu_{R_2}(y_3, z_2)), \min(\mu_{R_1}(x_1, y_4), \mu_{R_2}(y_4, z_2))]\} = \{(x_1, z_2), \max[\min(0, 0), \min(1, 0), \min(0, 1), \min(1, 0)]\} = \{(x_1, z_2), \max[0, 0, 0, 0] = 0\}$, which agrees with (7). Similarly, (9) becomes $\mu_{R_3}(x_1, z_2) = \{(x_1, z_2), \max_y [\mu_{R_1}(x_1, y)\mu_{R_2}(y, z_2)]\} = \{(x_1, z_2), \max[\mu_{R_1}(x_1, y_1)\mu_{R_2}(y_1, z_2), \mu_{R_1}(x_1, y_2)\mu_{R_2}(y_2, z_2), \mu_{R_1}(x_1, y_3)\mu_{R_2}(y_3, z_2), \mu_{R_1}(x_1, y_4)\mu_{R_2}(y_4, z_2)]\} = \{(x_1, z_2), \max[(0 \times 0), (1 \times 0), (0 \times 1), (1 \times 0)]\} = \{(x_1, z_2), \max[0, 0, 0, 0] = 0\}$, which also agrees with (7). \square

While this example is not a proof of the validity of (8) and (9), it demonstrates that they both seem to be correct representations for $R(U, W)$.

The following shortcuts can be used to evaluate the max-min or max-product compositions.

Max-Min Composition: (1) Write out each element in the matrix product $Q(U, V) P(V, W)$; but, (2) treat each multiplication as a min operation; and, then, (3) treat each addition as a max operation.

Max-Product Composition: (1) Write out each element in the matrix product $Q(U, V) P(V, W)$; but, (2) treat each multiplication as an algebraic multiplication operation; and, then, (3) treat each addition as a max operation.

Example 10 (Continued): Let's use these two shortcuts to verify (8) and (9) for the 1-3 element of $R_3(U, W)$ in (7). Now (8) leads to $R_3(x_1, z_3) = 0 \times 0 + 1 \times 0 + 0 \times 0 + 1 \times 1 = \min(0, 0) + \min(1, 0) + \min(0, 0) + \min(1, 1) = \max(0, 0, 0, 1) = 1$. Similarly, (9) leads to $R_3(x_1, z_3) = 0 \times 0 + 1 \times 0 + 0 \times 0 + 1 \times 1 = \max(0, 0, 0, 1) = 1$. \square

The max-min and max-product compositions are not the only ones that correctly represent $R(U, W)$; however, they seem to be the most widely used ones.

2) *Fuzzy Compositions:* Next, we consider the composition of fuzzy relations from different product spaces that share a common set, namely $R(U, V)$ and $S(V, W)$, e.g., x is smaller than y and y is close to z . The composition of fuzzy relations from different product spaces that share a common set is defined analogously to the crisp composition, except that in the fuzzy case the sets are fuzzy sets. Associated with fuzzy relation R is its membership function $\mu_R(x, y)$, where $\mu_R(x, y) \in [0, 1]$ and associated with

fuzzy relation S is its membership function $\mu_S(y, z)$, where $\mu_S(y, z) \in [0, 1]$. When R and S are from discrete universes of discourse, then the fuzzy composition of R and S , denoted $R \circ S$, can be described either by a sagittal diagram, in which each branch is labeled by its membership function value, or a fuzzy relational matrix, in which each element is a positive real number between and including zero and unity. A mathematical formula for $\mu_{R \circ S}(x, z)$, that is motivated by (8) and (9), is the following sup-star composition of R and S :

$$\mu_{R \circ S}(x, z) = \sup_{y \in V} [\mu_R(x, y) * \mu_S(y, z)]. \quad (10)$$

When U, V , and W are discrete universes of discourse, then the sup operation is the maximum. Motivation for using the "star" operation, which, of course is short for a t -norm, comes from the crisp max-min and max-product compositions, because both the min and product are t -norms. Although it is permissible to use other t -norms, the most commonly used sup-star compositions are the sup-min and sup-product. The shortcuts for computing the sup-min and sup-product, given above, apply also to fuzzy compositions over discrete universes of discourse.

Example 11: Returning to Example 8, let us consider the sentence " x is much larger than y and y is very close to z ." We begin by creating membership functions for the relations " x is much larger than y ," $\mu_{ML}(x, y)$, and " y is very close to z ," $\mu_{VC}(y, z)$. As in Example 8, let $U = \{x_1, x_2, x_3\}$ and $V = \{y_1, y_2, y_3, y_4\}$; additionally, let $W = \{z_1, z_2, z_3\}$. Membership functions $\mu_{ML}(x, y)$ and $\mu_{VC}(y, z)$ are given by

$$\mu_{ML}(x, y) = \begin{matrix} & y_1 & y_2 & y_3 & y_4 \\ x_1 & 0.8 & 1 & 0.1 & 0.7 \\ x_2 & 0 & 0.8 & 0 & 0 \\ x_3 & 0.9 & 1 & 0.7 & 0.8 \end{matrix} \quad \text{and} \quad \mu_{VC}(y, z) = \begin{matrix} & z_1 & z_2 & z_3 \\ y_1 & 0.4 & 0.9 & 0.3 \\ y_2 & 0 & 0.4 & 0 \\ y_3 & 0.9 & 0.5 & 0.8 \\ y_4 & 0.6 & 0.7 & 0.5 \end{matrix}. \quad (11)$$

Using the max-min composition, we find that

$$\mu_{ML \circ VC}(x, z) = \begin{matrix} & z_1 & z_2 & z_3 \\ x_1 & 0.6 & 0.8 & 0.5 \\ x_2 & 0 & 0.4 & 0 \\ x_3 & 0.7 & 0.9 & 0.7 \end{matrix}. \quad (12)$$

whereas using the max-product composition, we find that

$$\mu_{ML \times VC}(x, z) = \begin{matrix} & z_1 & z_2 & z_3 \\ x_1 & 0.42 & 0.72 & 0.35 \\ x_2 & 0 & 0.32 & 0 \\ x_3 & 0.63 & 0.81 & 0.56 \end{matrix}. \quad (13)$$

Observe that, unlike the case of crisp compositions, for which exactly the same results are obtained using either

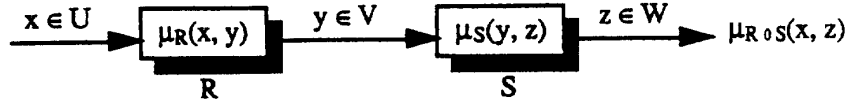
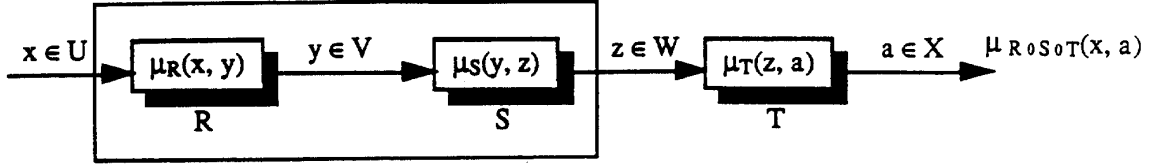
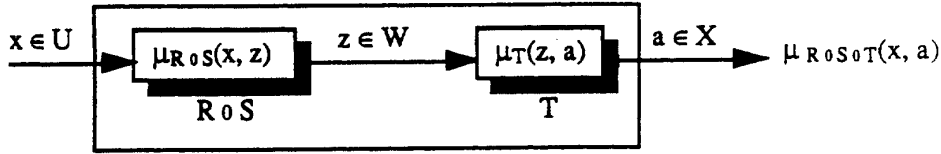


Fig. 11. Block diagram interpretation for the sup-star composition.



(a)



(b)

Fig. 12. Multiple fuzzy relations in (a) can be handled two at a time, as in (b).

the max-min or max-product compositions, the same results are not obtained in the case of fuzzy compositions. This is a major difference between fuzzy composition and crisp composition.

Fig. 11 provides a block diagram interpretation for the sup-star composition. It is equally valid for crisp and fuzzy compositions. It suggests a simple way to compose more complicated fuzzy relations.

Example 12: Fig. 12 depicts the interconnection of three fuzzy relations and how they can be composed. First relations R and S are composed using the sup-star composition; then, that result is composed with relation T , again using the sup-star composition. Of course, we could have first composed relations S and T , after which we could compose R with that result. \square

Suppose fuzzy relation R is just a fuzzy set, so that $\mu_R(x, y)$ just becomes $\mu_R(x)$, e.g., “ y is medium large and z is smaller than y .” Then $V = U$, and Fig. 11 reduces to Fig. 13, which gives us the sense of how a fuzzy set can activate a fuzzy relation. This special case will, as we shall demonstrate in the subsection below, entitled “Fuzzy Implication,” be extremely important to us, especially in our later development of our FLS.

What happens to the sup-star composition in this case? Because $V = U$, $\sup_{y \in V} [\mu_R(x, y) * \mu_S(y, z)] = \sup_{x \in U} [\mu_R(x) * \mu_S(x, z)]$ which is only a function of output variable z ; hence, we can simplify the notation $\mu_{R o S}(x, z)$ to $\mu_{R o S}(z)$, so that when R is just a fuzzy set,

$$\mu_{R o S}(z) = \sup_{x \in U} [\mu_R(x) * \mu_S(x, z)]. \quad (14)$$

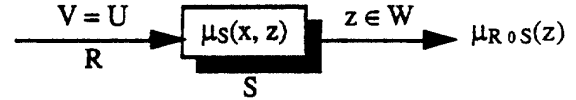


Fig. 13. Block diagram interpretation for the sup-star composition when the first relation is just a fuzzy set.

For discrete universes of discourse, we can evaluate the max-min or max-product compositions in (14) using the shortcuts described above; however, we must first create a row matrix for $\mu_R(x)$, i.e., if $x \in U = \{x_1, x_2, \dots, x_n\}$, let $R(U) = (\mu_R(x_1), \mu_R(x_2), \dots, \mu_R(x_n))$. Then, we have for:

Max-Min Composition: 1) Write out each element in the matrix product $R(U) S(U, W)$; but, 2) treat each multiplication as a min operation; and then, 3) treat each addition as a max operation.

Max-Product Composition: 1) Write out each element in the matrix product $R(U) S(U, W)$; but, 2) treat each multiplication as an algebraic multiplication operation; and, then, 3) treat each addition as a max operation.

I. Hedges

A linguistic hedge or modifier is an operation that modifies the meaning of a term, or more generally, of a fuzzy set. For example, if *weak pressure* is a fuzzy set, then *very weak pressure*, *more-or-less weak pressure*, *extremely weak pressure*, and *not-so weak pressure* are examples of hedges

which are applied to this fuzzy set. *Hedges can be viewed as operators that act upon a fuzzy set's membership function to modify it.* Here we give a small sample of these operators; many more can be found in [14].

- 1) *Concentration*: $\mu_{\text{con}(U)}(x) \triangleq [\mu_U(x)]^2$. If, for example, *weak pressure* has membership function $\mu_{\text{WP}}(p)$, then *very weak pressure* is a fuzzy set with membership function $[\mu_{\text{WP}}(p)]^2$, and *very very weak pressure* is a fuzzy set with membership function $[\mu_{\text{WP}}(p)]^4$. Because our membership functions have been assumed to be normalized, it is clear that the operation of *concentration* leads to a membership function that lies within the membership function of the original fuzzy set (thus, the term *concentration*); both have the same support, and the same membership values where the value of the original membership function equals unity or zero.
- 2) *Dilation*: $\mu_{\text{dil}(U)}(x) \triangleq [\mu_U(x)]^{1/2}$. If, for example, *weak pressure* has membership function $\mu_{\text{WP}}(p)$, then *more or less weak pressure* is a fuzzy set with membership function $[\mu_{\text{WP}}(p)]^{1/2}$. The operation of *dilation* leads to a membership function that lies outside of the membership function of the original fuzzy set (thus, the term *dilation*); both have the same support, and the same membership values where the value of the original membership function equals unity or zero.
- 3) *Artificial Hedges*: Two hedges that are quite useful are the *plus* and *minus* hedges, whose membership functions are $\mu_{\text{plus}(U)}(x) \triangleq [\mu_U(x)]^{1.25}$ and $\mu_{\text{minus}(U)}(x) \triangleq [\mu_U(x)]^{0.75}$. These artificial hedges provide milder degrees of concentration and dilation than those associated with the concentration and dilation hedges.

We have used the \triangleq sign in these hedge membership functions to convey the fact that the exponents used in the hedge membership functions are quite arbitrary; they can be changed depending upon our interpretation of the hedges.

Example 13: [85] We frequently use the phrase *highly unlikely*. Here we show how to obtain a membership function for it. Let U denote a universe of discourse associated with an appropriate quantity related to our notion of *likely*. We will clarify U below. Let $\mu_{\text{LIKELY}}(x)$ be the fuzzy membership function for the root concept of *likely*. Then,

$$\mu_{\text{HIGHLY UNLIKELY}}(x) = [1 - \mu_{\text{LIKELY}}(x)]^{4 \times 0.75}. \quad (15)$$

To obtain (15), we have interpreted the hedge *highly* as *minus very very* (which, of course, is subjective) and have used the fact that *unlikely* is the complement of *likely*.

From estimation theory (e.g., [19]), we know that *likelihood is proportional to probability*. This fact helps us to establish the universe of discourse, U , as values of probability (the constant of proportionality between probability and likelihood is irrelevant), i.e., $p \in U$ where $p \in [0, 1]$. As a concrete example, we assume the following discrete universe of discourse: $U = 0 + 0.1 + 0.2 + 0.3 + 0.4 + 0.5 + 0.6 + 0.7 + 0.8 + 0.9 + 1$, where the $+$

sign denotes union rather than arithmetic sum. In order to evaluate (15), we need to specify $\mu_{\text{LIKELY}}(p)$. Based on our own perception of the fuzzy set *likely*, we make the following choice for $\mu_{\text{LIKELY}}(p)$ (your choice may be different):

$$\mu_{\text{LIKELY}}(p) = 1/1 + 1/0.9 + 1/0.8 + 0.8/0.7 + 0.6/0.6 + 0.5/0.5 + 0.3/0.4 + 0.2/0.3. \quad (16)$$

Recall that the terms not shown have zero membership function values. Evaluating (15), we find that

$$\mu_{\text{HIGHLY UNLIKELY}}(p) \approx 1/0 + 1/0.1 + 1/0.2 + 0.5/0.3 + 0.3/0.4. \quad (17)$$

Observe, from (16) and (17), that the membership function $\mu_{\text{HIGHLY UNLIKELY}}(p)$ seems to make sense, i.e., it agrees with our own notion that something that is highly unlikely has a very very small chance (i.e., probability) of occurring. Consequently, large values for $\mu_{\text{HIGHLY UNLIKELY}}(p)$ should and indeed do occur for small values of probability, p .

III. SHORT PRIMER ON FUZZY LOGIC

From Fig. 2, we see that one of the major components of a FLS is *Rules*. Our rules will be expressed as logical implications, i.e., in the forms of IF-THEN statements, e.g., IF u is A , THEN v is B , where $u \in U$ and $v \in V$. A rule represents a special type of relation between A and B ; its membership function is denoted $\mu_{A \rightarrow B}(x, y)$. What is a proper and appropriate choice for this membership function? Nothing that we have presented so far helps us to answer this question, because an implication resides within a branch of mathematics known as logic, and so far we have been discussing set theory. Fortunately, as stated in [33] "It is well established that propositional logic is isomorphic to set theory under the appropriate correspondence between components of these two mathematical systems. Furthermore, both of these systems are isomorphic to a Boolean algebra, which is a mathematical system defined by abstract (interpretation-free) entities and their axiomatic properties. The isomorphisms between Boolean algebra, set theory, and propositional logic guarantee that every theorem in any one of these theories has a counterpart in each of the other two theories. These isomorphisms allow us, in effect, to cover all these theories by developing only one of them." Consequently, we will not spend a lot of time reviewing crisp logic; but, we must spend some time on it, especially on the concept of implication, in order to reach the comparable concept in fuzzy logic.

A. Crisp Logic (Much of This Section is Paraphrased from [1])

Rules are a form of propositions. A *proposition* is an ordinary statement involving terms which have been defined, e.g., "The damping ratio is low." Consequently, we could have the following rule: "IF the damping ratio is low, THEN the system's impulse response oscillates a long

Table 2 Truth Table for Five Operations That Are Frequently Applied to Propositions

| p | q | $p \wedge q$ | $p \vee q$ | $p \rightarrow q$ | $p \leftrightarrow q$ | $\sim p$ |
|-----|-----|--------------|------------|-------------------|-----------------------|----------|
| T | T | T | T | T | T | F |
| T | F | F | T | F | F | F |
| F | T | F | T | T | F | T |
| F | F | F | F | T | T | T |

time before it dies out.” In traditional propositional logic, a proposition must be meaningful to call it “true” or “false,” whether or not we know which of these terms properly applies.

Logical reasoning is the process of combining given propositions into other propositions, and then doing this over and over again. Propositions can be combined in many ways, all of which are derived from three fundamental operations: *conjunction* (denoted $p \wedge q$), where we assert the simultaneous truth of two separate propositions p and q (e.g., damping ratio is low and bandwidth is large); *disjunction* (denoted $p \vee q$) where we assert the truth of either or both of two separate propositions (e.g., I will design an analog filter or I will design a digital filter); and, *implication* (denoted $p \rightarrow q$) which usually takes the form of an IF-THEN rule, an example of which has been given in the previous paragraph. The IF part of an implication is called the *antecedent*, whereas the THEN part is called the *consequent*.

In addition to generating propositions using conjunction, disjunction or implication, a new proposition can be obtained from a given one by prefixing the clause “it is false that”. This is the operation of *negation* (denoted $\sim p$). Additionally, $p \leftrightarrow q$ is the *equivalence* relation; it means that p and q are both true or false.

In traditional propositional logic we combine unrelated propositions into an implication, and *we do not assume any cause or effect relation to exist*. We will see later that this last statement causes serious problems when we try to apply traditional propositional logic to engineering applications, where cause and effect rule (i.e., a system does not respond until an input is applied to it; a system’s impulse response is zero for all $t < 0$).

In traditional propositional logic an implication is said to be *true* if one of the following holds: 1) (antecedent is true, consequent is true), 2) (antecedent is false, consequent is false), and 3) (antecedent is false, consequent is true); the implication is called *false* when 4) (antecedent is true, consequent is false). Situation 1) is the familiar one of common experience. Situation 2) is also reasonable, for if we start from a false assumption we expect to reach a false conclusion, however, intuition is not always reliable. We may reason correctly from a false antecedent to a true consequent (e.g., IF $1 = 2$ is false, but, adding $2 = 1$ to this false statement, lets us correctly conclude that $3 = 3$); hence, a false antecedent can lead to a consequent which is either true or false, and thus both situations 2) and 3) are

allowed in traditional propositional logic. Finally, situation 4) is in accord with our intuition, for an implication is clearly false if a true antecedent leads to a false consequent.

A logical structure is constructed by applying the above four operations to propositions. The objective of a logical structure is to determine the truth or falsehood of all propositions which can be stated in the terminology of this structure.

A *truth table* is very convenient for showing relationships between several propositions. The fundamental truth tables for conjunction, disjunction, implication, equivalence and negation are collected together in Table 2, in which symbol T means that the corresponding proposition is true, and symbol F that it is false.

The fundamental axioms of traditional propositional logic are: 1) every proposition is either true or false, but not both true or false; 2) the expressions given by defined terms are propositions; and, 3) the truth table (in Table 2) for conjunction, disjunction, implication, equivalence, and negation. Using truth tables, we can derive many interpretations of the preceding operations and can also prove relationships about them.

A *tautology* is a proposition formed by combining other propositions (p, q, r, \dots) which is true regardless of the truth or falsehood of p, q, r, \dots . The most important tautology for our work is: $(p \rightarrow q) \leftrightarrow \sim [p \wedge (\sim q)]$. A proof of this tautology, using truth tables, is given in Table 3. Observe that the entries in the two columns $p \rightarrow q$ and $\sim [p \wedge (\sim q)]$ are identical; this proves the tautology. This tautology can also be expressed as $(p \rightarrow q) \leftrightarrow (\sim p) \vee q$, the truth of which is also demonstrated in Table 3. The importance of these tautologies is that they let us express the membership function for $p \rightarrow q$ in terms of membership functions of either propositions p and $\sim q$ or $\sim p$ and q , which was the main objective for this section.

Some of the most important mathematical equivalences between logic and set theory are

| Logic | Set Theory |
|----------|-----------------|
| \wedge | \cap |
| \vee | \cup |
| \sim | $(\bar{\quad})$ |

Additionally, as mentioned above, there is a correspondence between elementary logic and Boolean Algebra (0, 1). Any statement that is true in one system becomes a true statement in the other, simply by carrying through the following changes in notation:

| Logic | Boolean Algebra (0, 1) |
|-------------------|------------------------|
| T | 1 |
| F | 0 |
| \wedge | \times |
| \vee | $+$ |
| \sim | $'$ |
| \leftrightarrow | $=$ |
| p, q, r, \dots | a, b, c, \dots |

In this list, $'$ stands for complement, and a, b, c, \dots are arbitrary elements of the set (0, 1).

Table 3 Proofs of $(p \rightarrow q) \leftrightarrow \sim[p \wedge (\sim q)]$ and $(p \rightarrow q) \leftrightarrow (\sim p) \vee q$

| p | q | $(p \rightarrow q)$ | $\sim q$ | $p \wedge (\sim q)$ | $\sim [p \wedge (\sim q)]$ | $\sim p$ | $(\sim p) \vee q$ |
|-----|-----|---------------------|----------|---------------------|----------------------------|----------|-------------------|
| T | T | T | F | F | T | F | T |
| T | F | F | T | T | F | F | F |
| F | T | T | F | F | T | T | T |
| F | F | T | T | F | T | T | T |

Table 4 Validations of (18) and (19)

| $\mu_p(x)$ | $\mu_q(y)$ | $1 - \mu_p(x)$ | $1 - \mu_q(y)$ | $\max[1 - \mu_p(x), \mu_q(y)]$ | $1 - \min[\mu_p(x), 1 - \mu_q(y)]$ |
|------------|------------|----------------|----------------|--------------------------------|------------------------------------|
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |

Using the facts that $(p \rightarrow q) \leftrightarrow \sim[p \wedge (\sim q)]$ and $(p \rightarrow q) \leftrightarrow (\sim p) \vee q$, and the equivalences between logic and set theory, we can now obtain two (nonunique) membership functions for $\mu_{p \rightarrow q}(x, y)$. The first of these tautologies lets us show that

$$\begin{aligned} \mu_{p \rightarrow q}(x, y) &= 1 - \mu_{p \cap \bar{q}}(x, y) \\ &= 1 - \min[\mu_p(x), 1 - \mu_q(y)] \end{aligned} \quad (18)$$

and, the second of these tautologies lets us show that

$$\begin{aligned} \mu_{p \rightarrow q}(x, y) &= \mu_{\bar{p} \cup q}(x, y) \\ &= \max[1 - \mu_p(x), \mu_q(y)]. \end{aligned} \quad (19)$$

In order to validate the truth of these two membership functions, construct a Boolean truth table, such as the one in Table 4. Observe that the entries in the last two columns agree with the entries in Table 2 for $p \rightarrow q$, where we are interchanging logical T and F with Boolean 1 and 0, respectively.

The implication membership functions in (18) and (19) are by no means the only ones that give agreement with $p \rightarrow q$. Two others are (see [3] and [79] for many more):

$$\mu_{p \rightarrow q}(x, y) = 1 - \mu_p(x)(1 - \mu_q(y)) \quad (20)$$

$$\mu_{p \rightarrow q}(x, y) = \min[1, 1 - \mu_p(x) + \mu_q(y)]. \quad (21)$$

The membership function in (20) is similar to the one in (18), except that a product operation is used for conjunction instead of the minimum operation.

In traditional propositional logic there are two very important inference rules, *Modus Ponens* and *Modus Tollens*.

Modus Ponens—*Premise 1*: “ x is A ”; *Premise 2*: “IF x is A THEN y is B ”; *Consequence*: “ y is B .” *Modus Ponens* is associated with the implication “ A implies B ” [$A \rightarrow B$]. In terms of propositions p and q , *Modus Ponens* is expressed as $(p \wedge (p \rightarrow q)) \rightarrow q$.

Modus Tollens:—*Premise 1*: “ y is not B ”; *Premise 2*: “IF x is A THEN y is B ”; *Consequence*: “ x is not A .” In terms of propositions p and q , *Modus Tollens* is expressed as $(\bar{q} \wedge (p \rightarrow q)) \rightarrow \bar{p}$.

Whereas *Modus Ponens* plays a central role in engineering applications of logic, due in large part to cause and effect, *Modus Tollens* does not seem to have yet played much of a role.

B. Fuzzy Logic

Fuzzy logic begins by borrowing notions from crisp logic, just as fuzzy set theory borrows from crisp set theory; however, as we shall demonstrate below, doing this is inadequate for engineering applications of fuzzy logic, because, in engineering, *cause and effect is the cornerstone of modeling, whereas in traditional propositional logic it is not*. Ultimately, this will cause us to define “engineering” implication operators. Before doing this, let us develop an understanding of why the traditional approach fails us in engineering.

As in our extension of crisp set theory to fuzzy set theory, our extension of crisp logic to fuzzy logic is made by replacing the bivalent membership functions of crisp logic with fuzzy membership functions. That is all there is to it; hence, the IF-THEN statement “IF u is A , THEN v is B ,” where $u \in U$ and $v \in V$, has a membership function $\mu_{A \rightarrow B}(x, y)$ where $\mu_{A \rightarrow B}(x, y) \in [0, 1]$. Note that $\mu_{A \rightarrow B}(x, y)$ measures the degree of truth of the implication relation between x and y . Examples of such membership functions, are:

$$\mu_{A \rightarrow B}(x, y) = 1 - \min[\mu_A(x), 1 - \mu_B(y)] \quad (22)$$

$$\mu_{A \rightarrow B}(x, y) = \max[1 - \mu_A(x), \mu_B(y)] \quad (23)$$

and

$$\mu_{A \rightarrow B}(x, y) = 1 - \mu_A(x)(1 - \mu_B(y)) \quad (24)$$

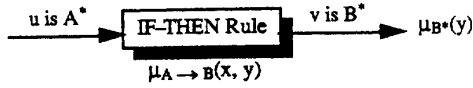


Fig. 14. System's interpretation for Generalized Modus Ponens. Remember that u and v are linguistic variables, and that their numerical values are x and y , respectively.

which, of course, are fuzzy versions of (18)–(20), respectively.

In fuzzy logic, Modus Ponens is extended to *Generalized Modus Ponens*—*Premise 1*: “ u is A^* ”; *Premise 2*: “IF u is A THEN v is B ”; *Consequence*: “ v is B^* .” Compare Modus Ponens and Generalized Modus Ponens to see their subtle differences, namely, in the latter, fuzzy set A^* is not necessarily the same as rule antecedent fuzzy set A , and fuzzy set B^* is not necessarily the same as rule consequent B .

Example 14: Consider the rule “IF a man is short, THEN he will not make a very good professional basketball player.” Here fuzzy set A is *short man*, and fuzzy set B is *not a very good professional basketball player*. We are now given Premise 1, as “This man is under five feet tall.” Here A^* is the fuzzy set *man under five feet tall*. Clearly, $A^* \neq A$; but, A^* is similar to A . We now draw the following consequence: “He will make a poor professional basketball player.” Here B^* is the fuzzy set *poor professional basketball player*, and $B^* \neq B$, although B^* is indeed similar to B . \square

We see that in crisp logic a rule will be fired only if the first premise is exactly the same as the antecedent of the rule, and, the result of such rule-firing is the rule's actual consequent. In fuzzy logic, on the other hand, a rule is fired so long as there is a nonzero degree of similarity between the first premise and the antecedent of the rule, and, the result of such rule-firing is a consequent that has a nonzero degree of similarity to the rule's consequent.

A system's interpretation for Generalized Modus Ponens is given in Fig. 14. This diagram coincides with Fig. 13, from which we conclude that Generalized Modus Ponens is a fuzzy composition where the first fuzzy relation is merely the fuzzy set, A^* . Consequently, $\mu_{B^*}(y)$ is obtained from the sup-star composition by comparing Figs. 14 and 13 and making the appropriate symbolic transformations in (14) (note that a max-min or max-product formula can also be stated for Modus Ponens), i.e.,

$$\mu_{B^*}(y) = \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \rightarrow B}(x, y)]. \quad (25)$$

In order to help us understand the meaning of (25), we shall consider some examples. In all these examples we assume that $\mu_{A^*}(x) = 1$ for $x = x'$ and $\mu_{A^*}(x) = 0$ for all other $x \in U$ with $x \neq x'$ (later, we will call this a *singleton fuzzifier*, and will learn why it is so popular).

Example 15: To begin, let us examine the result of using (22), for $\mu_{A \rightarrow B}(x, y)$, in (25), in which we use the

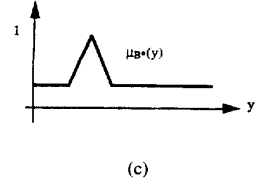
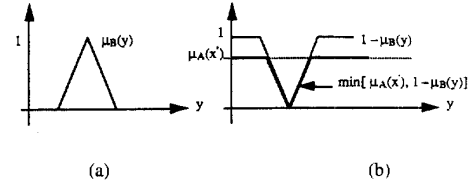


Fig. 15. Construction of $\mu_{B^*}(y)$ in (26). (a) Consequent membership function $\mu_B(y)$; (b) construction of $\min[\mu_{A^*}(x'), 1 - \mu_B(y)]$; (c) $\mu_{B^*}(y)$.

minimum as the t -norm for the star composition. We find that

$$\begin{aligned} \mu_{B^*}(y) &= \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \rightarrow B}(x, y)] \\ &= \mu_{A^*}(x') * \mu_{A \rightarrow B}(x', y) \\ &= 1 * \mu_{A \rightarrow B}(x', y) = \min[1, \mu_{A \rightarrow B}(x', y)] \\ &= \mu_{A \rightarrow B}(x', y) = 1 - \min[\mu_{A^*}(x'), 1 - \mu_B(y)]. \end{aligned} \quad (26)$$

Observe that for this $\mu_{A^*}(x)$ the supremum operation is unnecessary, because $\mu_{A^*}(x)$ is nonzero at only one point, x' . A graphical interpretation of this result is given in Fig. 15. The result, shown in (c), is disturbing for an engineering application. It tells us that, given a specific input $x = x'$, the result of firing a specific rule, whose consequent is associated with a specific fuzzy set of finite support [the base of the triangle in (a)], is a fuzzy set whose support is infinite. The same result is obtained if we use the product as the t -norm for the star composition, i.e., $1 * \mu_{A \rightarrow B}(x', y) = \mu_{A \rightarrow B}(x', y)$ when $*$ is minimum or product.

Note, also, from our derivation of (26) that for all $x \neq x'$, $\mu_{B^*}(y) = 1$ [i.e., $\mu_{A^*}(x \neq x') * \mu_{A \rightarrow B}(x \neq x', y) = 0 * \mu_{A \rightarrow B}(x \neq x', y) = \min[0, \mu_{A \rightarrow B}(x \neq x', y)] = 0$, so that $\mu_{B^*}(y) = 1 - \min[0, \mu_{A \rightarrow B}(x \neq x', y)] = 1$]. This means that this rule will be fired for all $x \neq x'$ with maximum possible output membership function value, unity. Clearly, this does not make much sense from an engineering perspective, where cause (e.g., system input) should lead to effect (e.g., system output), and noncause should not lead to anything. \square

Example 16: Perhaps the problem we experienced in Example 15 is a result of a poor choice for $\mu_{A \rightarrow B}(x, y)$. Let us, therefore, examine the result of using $\mu_{A \rightarrow B}(x, y)$ obtained from (21), i.e., $\mu_{A \rightarrow B}(x, y) = \min[1, 1 - \mu_A(x) + \mu_B(y)]$ which, by the way, is the implication membership function given by Zadeh in his important 1973 paper [85].

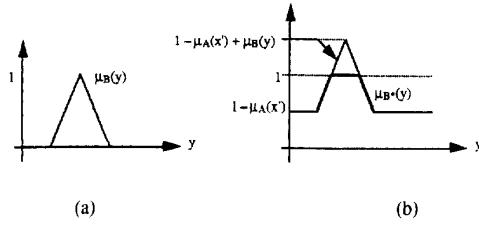


Fig. 16. Construction of $\mu_{B^*}(y)$ in (27). (a) Consequent membership function $\mu_B(y)$; (b) construction of $\mu_{B^*}(y)$.

Table 5 Demonstration That Minimum and Product Inferences Do Not Agree With $\mu_{p \rightarrow q}(x, y)$

| $\mu_p(x)$ | $\mu_q(y)$ | $\min[\mu_p(x), \mu_q(y)]$ | $\mu_p(x)\mu_q(y)$ |
|------------|------------|----------------------------|--------------------|
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 |

Substituting this expression for $\mu_{A \rightarrow B}(x, y)$ into (25), we find that:

$$\begin{aligned} \mu_{B^*}(y) &= \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \rightarrow B}(x, y)] \\ &= \mu_{A^*}(x') * \mu_{A \rightarrow B}(x', y) \\ &= 1 * \mu_{A \rightarrow B}(x', y) = \mu_{A \rightarrow B}(x', y) \\ &= \min[1, 1 - \mu_A(x') + \mu_B(y)] \end{aligned} \quad (27)$$

regardless of whether we use minimum or product for $*$. A graphical interpretation of this result is given in Fig. 16.

Once again, we have obtained a result, in Fig. 16(b) that violates engineering common sense. We leave it to the reader to demonstrate that all of the other choices we have provided for $\mu_{A \rightarrow B}(x, y)$ have the same problem. Even those which we have not listed here (e.g., see [3] and [79]) have the same problem. \square

Mamdani [49] seems to have been the first one to recognize the problem we have just demonstrated, although he does not explain it the way we have. He chose to work with the following *minimum implication*:

$$\mu_{A \rightarrow B}(x, y) \triangleq \min[\mu_A(x), \mu_B(y)]. \quad (28)$$

His reasons for choosing this definition do not seem to be based on cause and effect, but, instead on simplicity of computation. Later, Larsen [42] proposed the following *product implication*:

$$\mu_{A \rightarrow B}(x, y) \triangleq \mu_A(x)\mu_B(y). \quad (29)$$

Again, the reason for this choice was simplicity of computation rather than cause and effect. *Today, minimum and product inferences are the most widely used inferences in the engineering applications of fuzzy logic*; but, what do they have to do with traditional propositional logic?

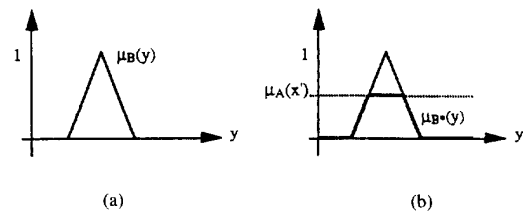


Fig. 17. Construction of $\mu_{B^*}(y)$ in (31). (a) Consequent membership function $\mu_B(y)$; (b) construction of $\mu_{B^*}(y)$.

Table 5 demonstrates that neither minimum inference nor product inference agree with the accepted propositional logic definition of implication, given in Table 2 and evaluated in Table 4; hence, minimum and product inferences have nothing to do with traditional propositional logic. Interestingly enough, Table 5 reveals that *minimum and product inferences preserve cause and effect*, i.e., $\mu_{p \rightarrow q}(x, y)$ is fired only when the antecedent and the consequent are both true. We, therefore, propose that minimum and product implications be referred to collectively as *engineering implications*.

Example 17: The purpose of this example is to demonstrate that both the minimum and product implications lead to output fuzzy sets that do not violate common engineering sense. As in Examples 15 and 16, we assume that $\mu_{A^*}(x) = 1$ for $x = x'$ and $\mu_{A^*}(x) = 0$ for all other $x \in U$ with $x \neq x'$. Then (25) becomes

$$\begin{aligned} \mu_{B^*}(y) &= \sup_{x \in A^*} [\mu_{A^*}(x) * \mu_{A \rightarrow B}(x, y)] \\ &= \mu_{A^*}(x') * \mu_{A \rightarrow B}(x', y) \\ &= 1 * \mu_{A \rightarrow B}(x', y) = \mu_{A \rightarrow B}(x', y) \end{aligned} \quad (30)$$

regardless of whether we use minimum or product for $*$.

Let us consider *minimum implication* first; then, (30) becomes

$$\mu_{B^*}(y) = \min[\mu_A(x'), \mu_B(y)]. \quad (31)$$

A graphical interpretation of this result is given in Fig. 17. Observe from (b) that given a specific antecedent $x = x'$, the result of firing a specific rule is a fuzzy set whose support is finite and is associated just with the consequent of that rule. Additionally, from our derivation of (31), observe that for all $x \neq x'$, $\mu_{B^*}(y) = 0$ [i.e., $\mu_{A^*}(x \neq x') * \mu_{A \rightarrow B}(x \neq x', y) = 0 * \mu_{A \rightarrow B}(x \neq x', y) = 0$, regardless of whether we use minimum or product for $*$]. I believe that these properties are desirable for engineering applications of fuzzy logic.

Next, we consider the *product implication* for which (30) becomes:

$$\mu_{B^*}(y) = \mu_A(x')\mu_B(y). \quad (32)$$

A graphical interpretation of this result is given in Fig. 18. We draw the same conclusions from this figure as we did for minimum implication; hence, our overall conclusions

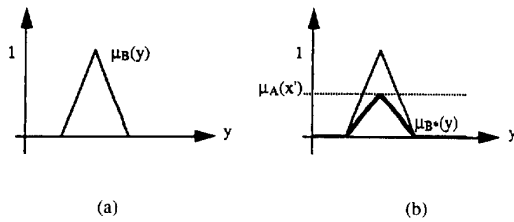


Fig. 18. Construction of $\mu_{B^*}(y)$ in (32). (a) Consequent membership function $\mu_B(y)$; (b) construction of $\mu_{B^*}(y)$.

are that *minimum and product inferences are, indeed, useful engineering implications*, and, that $\mu_{B^*}(y)$ can be expressed as $\mu_{B^*}(y) = \mu_A(x') \star \mu_B(y)$, where \star is either the minimum or product. \square

This completes our primers on fuzzy set theory and fuzzy logic. Some other topics, which appear frequently in the FL literature, and are sometimes used in engineering applications of FL, include: cardinality, extension principle, quantifiers $\exists x$ and $\forall x$, and α -cut of a fuzzy set; see [17], [33], [81], and [87].

IV. FUZZINESS AND OTHER MODELS

A lot has been written about fuzzy logic and its relation to probability (e.g., [9], [35], [43], [46] and *IEEE Transactions on Fuzzy Systems*, Mar. 1994, Special Issue). Many fuzzy logic theorists maintain that FL is quite different than probability, for a wide variety of reasons, including the facts that: the laws of excluded middle and contradiction are broken in FL, but are not broken in probability, and, that conditional probability must be defined in probability theory, but can be derived from first principles using FL [35], [36]. Others maintain that FL subsumes probability. Subjective (as distinguished from frequency based) probabilists on the other hand, maintain that anything one can do with FL can also be done with subjective probability, and that the latter is to be preferred because it has an axiomatic basis, whereas FL does not. They bemoan the fact that engineers, who are the largest users of FLS's, are not adequately trained in subjective probability.

The fact of the matter is that there is some truth to both sides of *fuzziness versus probability*. While it is of great intellectual interest to establish the proper connections between FL and probability, this author does not believe that doing so will change the ways in which we solve problems, because both probability and FL should be in the arsenal of tools used by engineers. FL will not solve all problems, nor will probability. Wang [73] argues it isn't that important whether you call something a "membership function" or a "probabilistic function." What is important is how to use the function to represent human knowledge.

That FL is a tool of enrichment and not replacement is clearly explained by Bezdek and Pal [5], who ask the question: "Where do fuzzy models fit in with other models?" They then give the following answer: "Fuzzy models belong wherever they can provide collateral or

competitively better information about a physical process. . . . We note that each of the following disciplines provides some information about the dynamics of motion: Newtonian mechanics, relativistic mechanics, statistical mechanics, quantum mechanics, and auto mechanics. These models provide us with different, useful, auxiliary, and sometimes contradictory information about various facets of dynamics. Each contributes something about the physical world, so it is with various classes of models. . . . From a different point of view, because every hard set is fuzzy but not conversely, the mathematical embedding of conventional set theory into fuzzy sets is as natural as the idea of embedding the real numbers into the complex plane. In both cases we can expect the larger "space" to contain answers to (real) questions that cannot be found in the smaller one. Thus the idea of fuzziness is one of enrichment not of replacement."

Addressing the fuzziness versus probability issue, Bezdek and Pal also ask "Isn't fuzziness just a clever disguise for probability?" Their answer is ". . . an emphatic no. There is a strong philosophical argument against regarding fuzziness as the surrogate for (frequency based) probability. The *spirit* of this argument is contained in (the following) example. Let L = set of all liquids, and let fuzzy subset $\mathcal{L} = \{ \text{all (potable) liquids} \}$. Suppose you had been in the desert for a week without a drink and you came upon two bottles marked C and A ." Bottle C is labeled $\mu_{\mathcal{L}}(C) = 0.91$ and bottle A is labeled $\text{Pr}[A \in \mathcal{L}] = 0.91$. "Confronted with this pair of bottles, and given that you must drink from the one you choose, which would you choose to drink from? Most readers when presented with this experiment immediately see that while C could contain, say, swamp water, it would not . . . contain liquids such as hydrochloric acid. That is *membership* of 0.91 means that the contents of C are fairly similar to perfectly potable liquids (e.g., pure water). On the other hand, the probability that A is potable = 0.91 means that over a long run of experiments, the contents of A are expected to be potable in about 91% of the trials; in the other 9% the contents will be deadly—about a 1 chance in 10. Thus most subjects will opt for a chance to drink swamp water. . . . There is another facet to this example, and it concerns the idea of *observation*. Continuing then, suppose we examine the contents of C and A and discover them to be" Dixie beer and hydrochloric acid, respectively. "Note that, *after observation*, the membership value of C is unchanged while the probability value for A drops from 0.91 to 0.0. This example shows that these two models possess philosophically different kinds of information: fuzzy memberships, which represent similarities of objects to imprecisely defined properties; and probabilities, which convey information about relative frequencies."

We are now ready to return to the FLS in Fig. 1.

V. FUZZY LOGIC SYSTEMS

We will now discuss the four elements of our Fig. 2 FLS, so that we will be able to write a mathematical formula that relates the output of the FLS to its inputs.

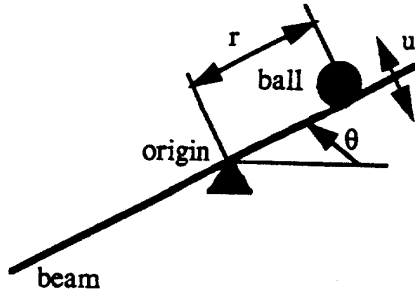


Fig. 19. Ball on beam.

A. Rules

A fuzzy rule base consists of a collection of IF-THEN rules, which can be expressed as:

$$R^{(l)}: \text{IF } u_1 \text{ is } F_1^l \text{ and } u_2 \text{ is } F_2^l \text{ and} \\ \dots u_p \text{ is } F_p^l, \text{ THEN } v \text{ is } G^l \quad (33)$$

where $l = 1, 2, \dots, M$, F_i^l and G^l are fuzzy sets in $U_i \subset R$ and $V \subset R$, respectively (R denotes the set of real numbers), $\mathbf{u} = \text{col}(u_1, \dots, u_p) \in U_1 \times \dots \times U_p$, and $v \in V$. \mathbf{u} and v are linguistic variables. Their numerical values are $\mathbf{x} \in U$ and $y \in V$, respectively. The main difference between this rule and the ones already presented is *multiple antecedents*. The following examples demonstrate how rules can be constructed for some engineering problems.

Example 18: Fig. 19 depicts a ball on a beam [76]. The beam is made to rotate in a vertical plane by applying a torque at the center of rotation and the ball is free to roll along the beam. We require the ball to remain in contact with the beam. The control $u(t)$ is the acceleration of θ . The problem is to design a controller that drives the ball into the origin so that the ball remains at the origin. This design must be accomplished regardless of where the ball starts on the beam, and regardless of the position of the beam.

This system is nonlinear and is described by four state variables, $r(t)$, $dr(t)/dt$, $\theta(t)$, and $d\theta(t)/dt$. Hauser *et al.* [21] have designed a control law that accomplishes the stated goal using an input-output linearization technique. Here we provide four high-level common sense rules that are associated with the control of the ball. They are four-input and one-output rules. By themselves, these high-level rules are unable to control the ball; but, taken together with a small amount of training data, generated by Hauser *et al.*'s controller, they are able to control the ball.

Consider the configuration shown in Fig. 19. If the ball stays at the depicted position (this corresponds to the IF part of $R^{(1)}$ below), then we should move the beam downwards (but not a lot) to reduce θ , which is equivalent to saying " u is *negative*," because the control equals the acceleration of θ . Similar reasoning can be made when the ball starts out to the right of the fulcrum and the beam is in the fourth quadrant, or to the left of the fulcrum and the beam is

in either the second or third quadrants. The resulting four high-level rules are:

- $R^{(1)}$: IF (radial position) r is *positive* and (radial velocity) dr/dt is *near zero* and (angular position) θ is *positive* and (angular velocity) $d\theta/dt$ is *near zero*, THEN (control) u is *negative*
- $R^{(2)}$: IF r is *negative* and dr/dt is *near zero* and θ is *negative* and $d\theta/dt$ is *near zero*, THEN u is *positive*
- $R^{(3)}$: IF r is *positive* and dr/dt is *near zero* and θ is *negative* and $d\theta/dt$ is *near zero*, THEN u is *positive big*
- $R^{(4)}$: IF r is *negative* and dr/dt is *near zero* and θ is *positive* and $d\theta/dt$ is *near zero*, THEN u is *negative big*

where $r, dr/dt, \theta$, and $d\theta/dt$ play a dual role of linguistic and numerical variables. Although the use of symbols for linguistic variables may seem somewhat sloppy notation, it really does not cause confusion.

I want to reemphasize the point that the rules just stated are in no way meant to be a control design; they act in a supervisory capacity.

Example 19: A well studied problem, both in neural networks and fuzzy logic control, is the *truck backing up problem*. Backing a truck into a loading dock is a difficult exercise for all but the most skilled truck drivers. It is a severely nonlinear control problem for which no traditional control system design method may exist. Nguyen and Widrow [55] developed a neural network controller for this problem. It was trained using numerical data, but did not use linguistic rules from experts. Kong and Kosko [34] developed a fuzzy control strategy for the same problem, one that initially only used linguistic rules, and later used numerical data and linguistic rules. Wang and Mendel [74], [75] developed a numerical-fuzzy approach that used both numerical data and linguistic rules.

Fig. 20 depicts the truck position in relation to the loading dock. The truck's position is determined by the three state variables ϕ , x and y . Control to the truck is the angle θ . Only backing up is considered, and enough clearance is assumed between the truck and the loading dock so that y does not have to be considered as an active state variable. The task is to design a control system whose inputs are $\phi \in [-90^\circ, 270^\circ]$ and $x \in [0, 20]$ and whose control is $\theta \in [-40^\circ, 40^\circ]$, such that the final states will be $(x_f, \phi_f) = (10, 90^\circ)$.

Let us assume that we have a collection of representative trajectories and control angles, and this information is used in a way that is described in [75] or [34] to obtain a set of 27 IF-THEN rules that are summarized in the relational matrix that is depicted in Fig. 21. This matrix is also referred to as a *fuzzy associative memory* [36]. Membership functions which are associated with ϕ , x and θ , are depicted in Fig. 22. Examples of some of the rules are (ϕ, x, θ) denote linguistic or numerical variables): $R^{(1, 2)}$: IF ϕ is $S3$ and x is $S1$, THEN θ is $S3$; $R^{(3, 5)}$: IF ϕ is $S1$ and x is $B2$, THEN θ is $S2$; $R^{(4, 3)}$: IF ϕ is CE and x is CE , THEN θ is CE ; $R^{(7, 5)}$: IF ϕ is $B3$ and x is $B2$, THEN θ is $B2$.

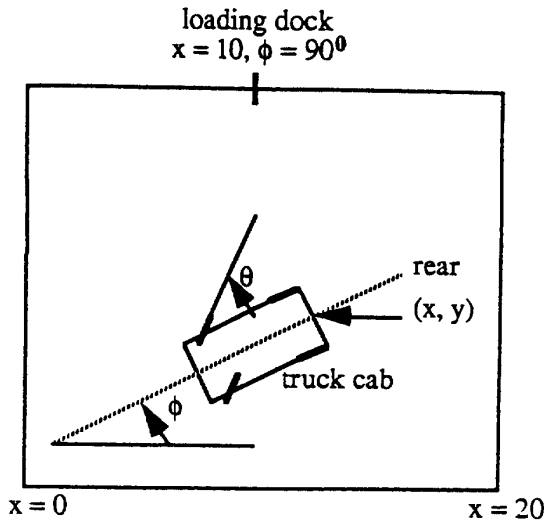
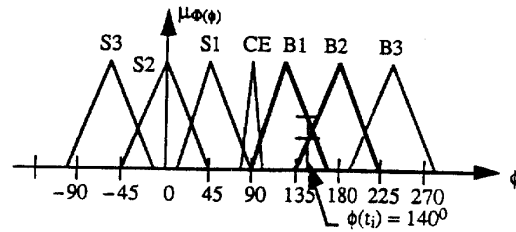


Fig. 20. Truck (cab) in relation to the loading dock.

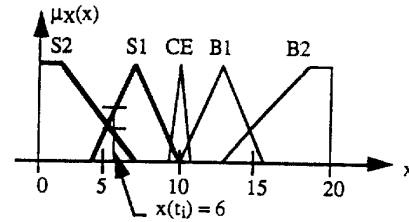
| | | | | | |
|-----------|----|----|----|----|----|
| | S2 | S3 | | | |
| S3 | S2 | S3 | | | |
| S2 | S2 | S3 | S3 | S3 | |
| S1 | B1 | S1 | S2 | S3 | S2 |
| ϕ C3 | B2 | B2 | CE | S2 | S2 |
| B1 | B2 | B3 | B2 | B1 | S1 |
| B2 | | B3 | B3 | B3 | B2 |
| B3 | | | | B3 | B2 |
| | S2 | S1 | CE | B1 | B2 |
| | x | | | | |

Fig. 21. Relational matrix with the rules of the truck backing up controller. The entries in the matrix are the fuzzy sets for control angle θ , which are a function of the two states, angular position, ϕ , and horizontal position, x . Blank entries have no consequent associated with them. See Fig. 22 for the membership functions which are associated with ϕ , x , and θ . Three activated rules at time t , are enclosed in the heavier square (see Example 22).

Example 20: In the problem of identifying a nonlinear dynamical system, where we have access to both the system's input, $x(k)$, and output, $y(k)$, suppose that we also have some rough high-level knowledge about the structure of the nonlinearity. To begin, we find out that the nonlinearity $f(\cdot)$ depends only on $y(k)$ and $y(k-1)$. Additionally, we are told that $f(\cdot)$ is close to zero when either $y(k)$ is close to zero or -4 , or when $y(k-1)$ is close to zero. This qualitative information can be expressed as the following three rules: $R^{(1)}$: IF $y(k)$ is close to zero, THEN $f(y(k), y(k-1))$ is close to zero; $R^{(2)}$: IF $y(k)$ is close to -4 , THEN $f(y(k), y(k-1))$ is close to zero; $R^{(3)}$: IF $y(k-1)$ is close to zero, THEN $f(y(k), y(k-1))$ is



(a)



(b)

Fig. 22. Fuzzy membership functions for the truck backing up problem. These membership functions were chosen by designers. In these figures, S, CE, and B are short for *small*, *center*, and *big*, respectively. Fuzzy sets that are activated by the measured states at the time t , are shown in (a) and (b) (see Example 22). These activated fuzzy sets are shown darker than the unactivated sets.

close to zero. The use of such rules in the identification of $f(\cdot)$ has been shown by Wang and Mendel [77] to accelerate the convergence of $\hat{f}(\cdot)$ to $f(\cdot)$, and to improve the approximation of $f(\cdot)$ by $\hat{f}(\cdot)$.

Example 21: Let $x(k), k = 1, 2, \dots$, be a time series, such as daily temperatures of Juneau, AK, hourly measurements of the Dow-Jones stock index, or the chaotic Mackey-Glass time series depicted in Fig. 1. The problem of time-series prediction (i.e., forecasting) is: Given a window of n past measurements of $x(k)$, namely $x(k-n+1), x(k-n+2), \dots, x(k)$, determine a future value of $x, x(k+l)$, where n and l are fixed positive integers. In this example, let us suppose that $l = 1$, so that we are interested in the single-stage predictor of x .

Suppose that we are given a collection of D data points, $x(1), x(2), \dots, x(D)$, so that there are at most $D-n$ training pairs, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(D-n)}$, where $\mathbf{x}^{(j)} = \text{col}[n \times 1 \text{ input: desired output}]$ and: $\mathbf{x}^{(1)} = \text{col}[x(1), x(2), \dots, x(n) : x(n+1)]$, $\mathbf{x}^{(2)} = \text{col}[x(2), x(3), \dots, x(n+1) : x(n+2)]$, \dots , $\mathbf{x}^{(D-n)} = \text{col}[x(D-n), \dots, x(D-1) : x(D)]$.

There are at least two ways to extract rules from numerical data: 1) let the data establish the fuzzy sets that appear in the antecedents and consequents of the rules, or, 2) prespecify fuzzy sets for the antecedents and consequents and then associate the data with these fuzzy sets. We will briefly describe both approaches. Because a predicted value of x will depend on n past values of x , there will be

n antecedents in each rule. Let these n antecedents be denoted u_1, u_2, \dots , and u_n . The interesting feature of this example is that, although each rule has n antecedents, these antecedents are all associated with the *same* variable, $x(k)$, and so is the consequent.

For purposes of single-stage prediction, here are $D - n$ rules that we can cull from the D training pairs, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(D-n)}$:

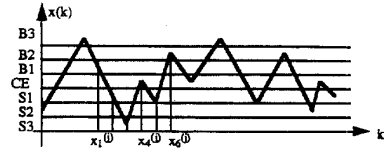
$R^{(1)}$: IF u_1 is F_1^1 and u_2 is F_2^1 and \dots and u_n is F_n^1 , THEN v is G^1 . In this rule, F_1^1 is a fuzzy set whose membership function is centered at $x(1)$, F_2^1 is a fuzzy set whose membership function is centered at $x(2)$, \dots , F_n^1 is a fuzzy set whose membership function is centered at $x(n)$, and G^1 is a fuzzy set whose membership function is centered at $x(n + 1)$.

$R^{(2)}$: IF u_1 is F_1^2 and u_2 is F_2^2 and \dots and u_n is F_n^2 , THEN v is G^2 . In this rule, F_1^2 is a fuzzy set whose membership function is centered at $x(2)$, F_2^2 is a fuzzy set whose membership function is centered at $x(3)$, \dots , F_n^2 is a fuzzy set whose membership function is centered at $x(n + 1)$, and G^2 is a fuzzy set whose membership function is centered at $x(n + 2)$.

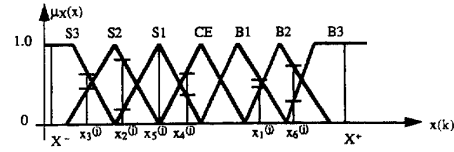
$R^{(D-n)}$: IF u_1 is F_1^{D-n} and u_2 is F_2^{D-n} and \dots and u_n is F_n^{D-n} , THEN v is G^{D-n} . In this rule, F_1^{D-n} is a fuzzy set whose membership function is centered at $x(D - n)$, F_2^{D-n} is a fuzzy set whose membership function is centered at $x(D - n + 1)$, \dots , F_n^{D-n} is a fuzzy set whose membership function is centered at $x(D - 1)$, and G^{D-n} is a fuzzy set whose membership function is centered at $x(D)$.

In this first approach to obtaining rules from numerical data, we see that the antecedent and consequent membership functions adapt to the locations of the data that are used to create the rules.

In the second approach [71], [73], [75] we begin by establishing fuzzy sets for all the antecedents and the consequent. This is done by first establishing domain intervals for all input and output variables. For the example of time-series prediction, these domain intervals are all the same, because u_1, u_2, \dots, u_n and v are all sampled values of the time series, $x(k)$. In other situations (e.g., the ball and beam Example 18), each antecedent and consequent will have different domain intervals. Let us assume that, by examining the time series, we establish that $x(k) \in [X^-, X^+] = U$. Next, we divide each domain interval into a prespecified number of overlapping regions. The number of overlapping regions does not have to be the same for each variable, and the lengths of these overlapping regions can be equal or unequal. Each overlapping region is then labeled and is assigned a membership function. Resolution in prediction can be controlled by the coarseness of the fuzzy sets that are associated with $x(k)$. Membership functions could be of different types for different variables. Measured values of a variable are permitted to lie outside of the variable's domain interval, because if $x(k) > X^+$, then $\mu_X(x) = 1$ (see Fig. 23(b)).



(a)



(b)

Fig. 23. Construction to determine degrees of the elements of $\mathbf{x}^{(j)}$, when $n = 5$. (a) A time series and its intersection with the seven fuzzy sets shown in (b), and the six elements of the j th window of data; and (b) the six elements of the j th window of data projected up to intersect membership functions. In (a) the elements of $\mathbf{x}^{(j)}$ appear in chronological order, because the horizontal axis is time. In (b) the elements of $\mathbf{x}^{(j)}$ do not appear in chronological order, because the horizontal axis is amplitude of $x(k)$. The amplitude scales are different in (a) and (b).

Fuzzy rules are generated from the given data pairs using the following three-step procedure [75]:

- 1) Determine the degrees (i.e., the membership function values) of the elements of $\mathbf{x}^{(j)}$. As an example, in Fig. 23 we consider the case when $n = 5$. Examining Fig. 23b, we see that $x_1^{(j)}$ has degree 0.45 in B2 and 0.55 in B1, $x_2^{(j)}$ has degree 0.2 in S1 and 0.8 in S2, $x_3^{(j)}$ has degree 0.45 in S2 and 0.6 in S3, $x_4^{(j)}$ has degree 0.4 in S1 and 0.6 in CE, $x_5^{(j)}$ has degree 1.0 in S1, and $x_6^{(j)}$ has degree 0.3 in B3 and 0.7 in B2.
- 2) Assign each variable to the region with maximum degree, e.g., $x_1^{(j)}$ above is considered to be B1, $x_2^{(j)}$ is considered to be S2, $x_3^{(j)}$ is considered to be S3, $x_4^{(j)}$ is considered to be CE, $x_5^{(j)}$ is considered to be S1, and $x_6^{(j)}$ is considered to be B2.
- 3) Obtain one rule from one pair of desired input-output data, e.g., IF $x_1^{(j)}$ is B1 and $x_2^{(j)}$ is S2 and $x_3^{(j)}$ is S3 and $x_4^{(j)}$ is CE and $x_5^{(j)}$ is S1, THEN $y(j)$ is B2.

Because there can be lots of data, it is quite likely that there will be some *conflicting rules*, i.e., rules with the same antecedents but different consequents. We resolve this by assigning a degree, $D(R^{(j)})$, to each rule and accept only the rule from a conflict group that has maximum degree, where $D(R^{(j)}) \triangleq \mu_X(x_1^{(j)})\mu_X(x_2^{(j)})\dots\mu_X(x_n^{(j)})\mu_X(y(j))$. For our example, we find, from step one of our three-step procedure, that $D(R^{(j)}) = 0.55 \times 0.88 \times 0.6 \times 0.6 \times 1.0 \times 0.7 = 0.122$.

This three-step procedure is carried out for all $D - n$ training pairs. The results are at most $D - n$ linguistic rules of the form just obtained for $R^{(j)}$ in step 3. \square

A *multiple-antecedent multiple-consequent* (i.e., multi-input multi-output) *rule* can always be considered as a group of multi-input single-output rules [45, p. 426], which is why the literature concentrates on multi-input single-output rules. This follows from crisp logic, e.g., $(p \rightarrow (q_1 \wedge q_2 \wedge q_3)) \leftrightarrow ((p \rightarrow q_1) \wedge (p \rightarrow q_2) \wedge (p \rightarrow q_3))$.

It is also possible to cast “nonobvious” rules into the form of (33). Six such rules are summarized next, because it is very important for the reader to understand the power and flexibility of the generic rule structure in (33). The first five are adapted from [73].

- 1) **Incomplete IF Rules:** Suppose that we have created a rule base where there are p inputs, but some rules have antecedents that are only a subset of the p inputs, e.g., IF u_1 is F_1^l and u_2 is F_2^l and \dots and u_m is F_m^l , THEN v is G^l . Such rules are called *incomplete IF rules*, and apply regardless of u_{m+1}, \dots, u_p . They can be put into the format of the complete IF rule (33) by treating the unnamed antecedents, e.g., u_{m+1}, \dots, u_p , as elements of the fuzzy set INCOMPLETE (IN for short) where, by definition $\mu_{IN}(u) = 1$ for all $u \in R$, i.e., (IF u_1 is F_1^l and u_2 is F_2^l and \dots and u_m is F_m^l , THEN v is G^l) \leftrightarrow (IF u_1 is F_1^l and u_2 is F_2^l and \dots and u_m is F_m^l and u_{m+1} is IN \dots and u_p is IN, THEN v is G^l).
- 2) **Mixed Rules:** Not all rules use the “and” connective; some use the “or” connective, and some use a mixture of both connectives. Such rules are called *mixed rules*. These rules can be decomposed into a collection of equivalent rules, using standard techniques from crisp logic. Suppose, for example, we have the rule: IF u_1 is F_1^l and u_2 is F_2^l and \dots and u_m is F_m^l , or u_{m+1} is F_{m+1}^l and \dots and u_p is F_p^l , THEN v is G^l . This rule can be expressed as the following two rules: $R^{(1)}$: IF u_1 is F_1^l and u_2 is F_2^l and \dots and u_m is F_m^l , THEN v is G^l ; and, $R^{(2)}$: IF u_{m+1} is F_{m+1}^l and \dots and u_p is F_p^l , THEN v is G^l . Observe that both of these rules are Incomplete IF rules. See [70] for related discussions on nesting of rules.
- 3) **Fuzzy Statement Rules:** Some rules do not appear to have any antecedents; they are statements involving fuzzy sets; hence, they are called *fuzzy statement rules*. For example, “ v is G^l ” is such a rule. Clearly, this is an extreme case of an incomplete IF rule, and can therefore be formulated as: IF u_1 is IN and u_2 is IN and \dots and u_p is IN, THEN v is G^l .
- 4) **Comparative Rules:** Some rules are comparative, e.g., “The smaller the u , the bigger the v .” Such rules must first be reformulated as IF-THEN rules; this takes some experience. The preceding rule can be expressed as “IF u is S , THEN v is B ,” where S is a fuzzy set representing *smaller* and B is a fuzzy set representing *bigger*.
- 5) **Unless Rules:** Rules are sometimes stated using the connective “unless”; such rules are called *unless rules* and can be put into the format of (33) by using logical operations, including De Morgan’s Laws. For

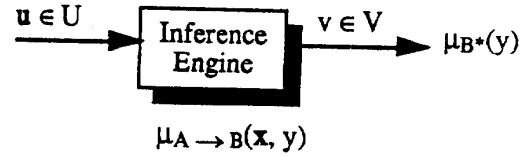


Fig. 24. Interpretation of fuzzy inference engine as a system. Again, remember that \mathbf{x} and y are the numerical values of the linguistic variables u and v , respectively.

example, the rule v is G^l unless u_1 is F_1^l and u_2 is F_2^l and \dots and u_p is F_p^l can first be expressed as IF not (u_1 is F_1^l and u_2 is F_2^l and \dots and u_p is F_p^l), THEN v is G^l . Using De Morgan’s Law, $\overline{A \cap B} = \overline{A} \cup \overline{B}$, this can be reexpressed as IF u_1 is not F_1^l or u_2 is not F_2^l or \dots or u_p is not F_p^l , THEN v is G^l . We treat “not F_i^l ” as a fuzzy set, and then decompose this *mixed rule* into a collection of p *incomplete IF rules* each of the form IF u_i is not F_i^l , THEN v is G^l , $i = 1, 2, \dots, p$.

- 6) **Quantifier Rules:** Rules sometimes include the quantifiers “some” or “all”; such rules are called *quantifier rules*. Because of the duality between propositional logic and set theory, rules with the quantifier “some” mean that we have to apply the union operator to the antecedents or consequents to which the “some” applies, whereas rules with the quantifier “all” mean we have to apply the intersection operator to the antecedents or consequents to which the “all” applies.

Of course, in practical applications, it is possible to have rules that combine nonobvious IF-THEN rules 1–6 in all sorts of ways.

B. Fuzzy Inference Engine

In the fuzzy inference engine (which is labeled “inference” in Fig. 2) fuzzy logic principles are used to combine fuzzy IF-THEN rules from the fuzzy rule base into a mapping from fuzzy input sets in $U := U_1 \times U_2 \times \dots \times U_p$ to fuzzy output sets in V . Each rule is interpreted as a fuzzy implication. With reference to (33), let $F_1^l \times F_2^l \times \dots \times F_p^l \triangleq A$ and $G^l \triangleq B$; then, $R^{(l)} : F_1^l \times F_2^l \times \dots \times F_p^l \rightarrow G^l = A \rightarrow B$. We treat the fuzzy inference engine as a system, one that maps fuzzy sets into fuzzy sets by means of $\mu_{A \rightarrow B}(\mathbf{x}, y)$; this is depicted in Fig. 24, which we recognize is the same as Fig. 14, except that *the input is now a vector, because our rules have multiple antecedents*.

In the rest of this paper we assume that our universes of discourse are discrete, and that each U_i ($i = 1, 2, \dots, p$) and G are finite, so that $R^{(l)}$ is given by a discrete multivariate membership function $\mu_{R^{(l)}}(\mathbf{x}, y)$, described by $\mu_{R^{(l)}}(\mathbf{x}, y) = \mu_{A \rightarrow B}(\mathbf{x}, y)$ where: $\mathbf{x} \triangleq \text{col}(x_1, x_2, \dots, x_p)$.

Consequently, $\mu_{R^{(l)}}(\mathbf{x}, y) = \mu_{R^l}[x_1, \dots, x_p, y]$ and

$$\begin{aligned} \mu_{R^{(l)}}(\mathbf{x}, y) &= \mu_{A \rightarrow B}(\mathbf{x}, y) \\ &= \mu_{F_1^l}(x_1) * \dots * \mu_{F_p^l}(x_p) * \mu_{G^l}(y) \end{aligned} \quad (34)$$

where it has been assumed that multiple antecedents are connected by *and*'s, and subsequently by *t*-norms, and, that only the product or minimum *t*-norms are used. The p -dimensional input to $R^{(l)}$ is given by the fuzzy set A_x whose membership function is (recall that the discrete fuzzy set A_x is written as $\sum \sum \dots \sum \mu_{A_x}(\mathbf{x})/\mathbf{x}$, where the p summations denote union operations):

$$\mu_{A_x}(\mathbf{x}) = \mu_{x_1}(x_1) * \dots * \mu_{x_p}(x_p) \quad (35)$$

where $X_k \subset U_k (k = 1, \dots, p)$ are the fuzzy sets describing the inputs. Each rule $R^{(l)}$ determines a fuzzy set $B^l = A_x \circ R^{(l)}$ in \mathbf{R} such that [see (25)]

$$\begin{aligned} \mu_{B^l}(y) &= \mu_{A_x \circ R^{(l)}}(y) \\ &= \sup_{\mathbf{x} \in A_x} [\mu_{A_x}(\mathbf{x}) * \mu_{A \rightarrow B}(\mathbf{x}, y)]. \end{aligned} \quad (36)$$

This equation is the input-output relationship in Fig. 2 between the fuzzy set that excites a one-rule inference engine and the fuzzy set at the output of that engine.

Example 17 and its associated Figs. 17 and 18 are made applicable to $\mu_{B^l}(y)$ in (36), by replacing: 1) the scalar antecedent x by the vector antecedent \mathbf{x} , and 2) $\mu_A(x')$ by $\mu_{A_x}(\mathbf{x}') = \mu_{X_1}(x'_1) * \dots * \mu_{X_p}(x'_p)$. When a min *t*-norm is used, then $\mu_{A_x}(\mathbf{x}') = \min[\mu_{X_1}(x'_1), \dots, \mu_{X_p}(x'_p)]$, whereas when a *product t*-norm is used, then $\mu_{A_x}(\mathbf{x}') = \mu_{X_1}(x'_1) \cdot \dots \cdot \mu_{X_p}(x'_p)$.

The final fuzzy set $B = A_x \circ [R^{(1)}, R^{(2)}, \dots, R^{(M)}]$, which is determined by all the rules in the rule base, is obtained by combining B^l and its associated membership function $\mu_{A_x \circ R^{(l)}}(y)$ for all $l = 1, 2, \dots, M$. Zadeh [85] connected rules using the word "else," one of whose definitions is *otherwise*. Lee [45] uses the connective "also" and has a discussion about a number of studies that were performed to determine the best way to connect rules. *Most people connect rules using a t-conorm* (i.e., the fuzzy union), and these seem to give very good results when our engineering implication operators are used, i.e., $B = B^1 \oplus B^2 \oplus \dots \oplus B^M$.

Lee [45] provides a rigorous proof that the sup-min or sup-product compositions and connective "also," interpreted as a max *t*-conorm, are commutative, i.e., $A_x \circ [R^{(1)}, R^{(2)}, \dots, R^{(M)}] = \cup_{i=1}^M A_x \circ R^{(i)}$. For additional discussions on connecting rules, see [18], [30], [31], [32], [51], [64].

There does not appear to be a unique or compelling theoretical reason for combining rules using a *t*-conorm. We have already seen that engineering applications require engineering implications; such applications may also require *engineering connectives*. Combining rules additively [36] is one such engineering connective. Kosko calls such a FLS (with an appropriate fuzzifier and defuzzifier) an *additive FLS*. Note, from our earlier definition of a *t*-conorm and our examples for them, that arithmetic addition is not a *t*-conorm (the algebraic sum $x \oplus y = x + y - xy$

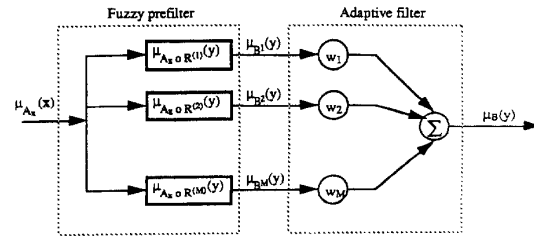


Fig. 25. An additive combiner which can be interpreted as an adaptive filter that is activated by fuzzy sets that are the output of a fuzzy prefilter.

is a *t*-conorm). Fig. 25 depicts the additive combiner. It resembles an adaptive filter whose inputs are the output fuzzy sets. The weights w_1, w_2, \dots, w_M of the combiner can be thought of as providing degrees of belief to each rule. It is conceivable, that we know that some rules are more reliable than others; such rules would be assigned a larger weight than less reliable rules. If such information is not known ahead of time, then we either set all the weights equal to unity or we use a training procedure to learn optimal values for the weights.

We will return to the question of how to combine rules, when we later discuss the defuzzifier block of Fig. 2, to see yet another method for engineering connectives.

Example 22: Let us return to the truck backing up Example 19, for which membership functions are given in Fig. 22 and a set of 27 IF-THEN rules are summarized in Fig. 21. Imagine that we are backing up the truck, and at some arbitrary time t_i the states of the truck are $\phi(t_i) = 140^\circ$ and $x(t_i) = 6$. Observe that $\phi(t_i)$ activates two fuzzy sets, $B1$ and $B2$, whereas $x(t_i)$ also activates two fuzzy sets, $S1$ and $S2$. Referring to the relational matrix in Fig. 21, this means that the following three rules are activated: $R^{(5,1)}$: IF ϕ is $B1$ and x is $S2$, THEN θ is $B2$; $R^{(5,2)}$: IF ϕ is $B1$ and x is $S1$, THEN θ is $B3$; and $R^{(6,2)}$: IF ϕ is $B2$ and x is $S1$, THEN θ is $B3$.

Let us now compute the output fuzzy set that is activated for each of these rules. As in Examples 15–17, we assume $\mu_{A_x}(\mathbf{x}') = 1$ for $\mathbf{x} = \mathbf{x}'$ and $\mu_{A_x}(\mathbf{x}) = 0$ for all other $\mathbf{x} \in U = U_\phi \times U_x$. It is informative to demonstrate these computations graphically. Later we will obtain formulas for them that can be programmed, so that they can be performed automatically. Fig. 26 depicts the activation of $R^{(5,1)}$ and its output fuzzy set, for either minimum or product inference. Figs. 27 and 28 depict the comparable quantities for $R^{(5,2)}$ and $R^{(6,2)}$, respectively. Finally, Fig. 29 depicts the overall output fuzzy set obtained using the max *t*-conorm.

A remaining question, motivated by Fig. 29, is "What numerical value will be used for the steering wheel control $\theta(t_i)$? The answer to this question is the subject of *defuzzification*. □

The graphical interpretation is easily extended from two antecedents to more than two antecedents, as in the case of the ball and beam Example 18, or the time-series prediction

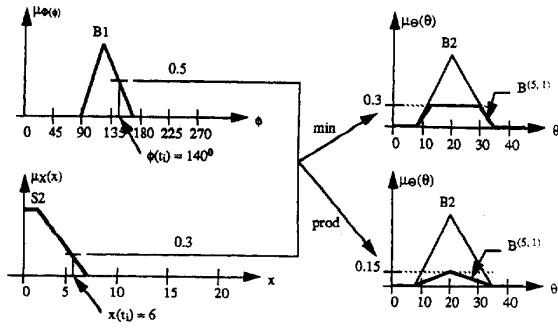


Fig. 26. Activation of $R^{(5,1)}$ and its output fuzzy set, $B^{(5,1)}$, for either minimum or product inference. Membership functions were taken from Fig. 22.

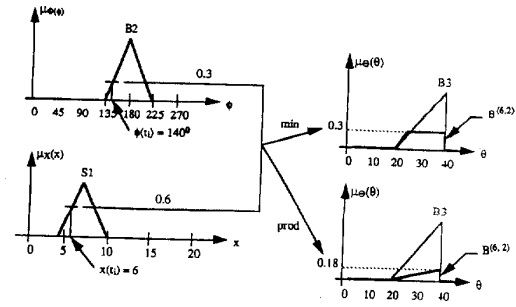


Fig. 28. Activation of $R^{(6,2)}$ and its output fuzzy set, $B^{(6,2)}$, for either minimum or product inference.

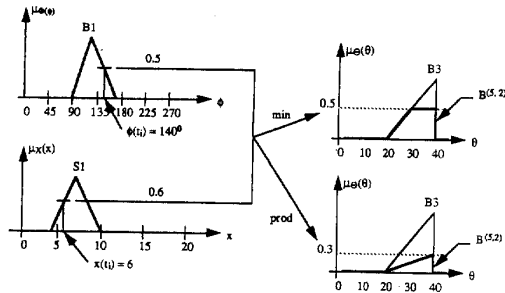


Fig. 27. Activation of $R^{(5,2)}$ and its output fuzzy set, $B^{(5,2)}$, for either minimum or product inference.

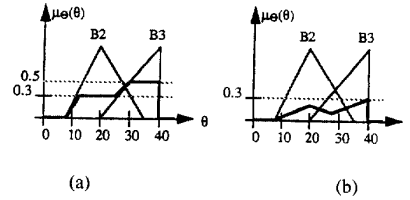


Fig. 29. The overall output fuzzy set obtained using the max t -conorm: (a) minimum inference, and (b) product inference.

Example 19 (when $n > 2$). A formula interpretation is described later.

C. Fuzzification

The *fuzzifier* maps a crisp point $\mathbf{x} = \text{col}(x_1, \dots, x_n) \in U$ into a fuzzy set A^* in U . The most widely used fuzzifier is the *singleton fuzzifier* which is nothing more than a fuzzy singleton, i.e., A^* is a fuzzy singleton with support \mathbf{x}' if $\mu_{A^*}(\mathbf{x}') = 1$ for $\mathbf{x} = \mathbf{x}'$ and $\mu_{A^*}(\mathbf{x}') = 0$ for all other $\mathbf{x} \in U$ with $\mathbf{x} \neq \mathbf{x}'$. When fuzzy input set A^* only contains a single element \mathbf{x}' , then the supremum operation in the sup-star composition (36) disappears, i.e., $\mu_{B^l}(y) = \mu_{A_x \circ R^{(l)}}(y) = \mu_{A \rightarrow B}(\mathbf{x}', y)$. Examples 15–17 demonstrated this simplification, and it is this (tremendous) simplification of the sup-star composition that is (in the opinion of this author) the reason for the popularity of singleton fuzzification.

Singleton fuzzification may not always be adequate, especially when data is corrupted by measurement noise. Nonsingleton fuzzification provides a means for handling such uncertainties totally within the framework of FLS's. A *nonsingleton fuzzifier* is one for which $\mu_{A^*}(\mathbf{x}') = 1$ and $\mu_{A^*}(\mathbf{x})$ decreases from unity as \mathbf{x} moves away from \mathbf{x}' . In nonsingleton fuzzification, \mathbf{x}' is mapped into a fuzzy number [28], i.e., a fuzzy membership function is associated with it. Examples of such membership functions are the

Gaussian and triangular. The broader these functions are, the greater is the uncertainty about \mathbf{x}' .

What happens to (36) in the case of nonsingleton fuzzification? Substituting (34) and (35) into (36), making use of the fact that all unions (denoted by \sum) in A_x and $R^{(l)}$, over i_k ($k = 1, \dots, p$) are over the same spaces, we can write the output membership function for the fuzzy set for the l th rule, as [53], [54]

$$\mu_{B^l}(y) = \sup_{\mathbf{x} \in U} \mu_{X_1}(x_1) * \dots * \mu_{X_p}(x_p) * \mu_{F_1^l}(x_1) * \dots * \mu_{F_p^l}(x_p) * \mu_{G^l}(y). \quad (37)$$

Since the supremum is only over $\mathbf{x} \in U$, then by the commutativity and monotonicity properties of a t -norm, we can rewrite $\mu_{B^l}(y)$ as:

$$\mu_{B^l}(y) = \mu_{G^l}(y) * \sup_{\mathbf{x} \in U} [\mu_{X_1}(x_1) * \dots * \mu_{X_p}(x_p) * \mu_{F_1^l}(x_1) * \dots * \mu_{F_p^l}(x_p)]. \quad (38)$$

This means that we only need to compute one supremum rather than m suprema (m is the number of discrete points in the universe of discourse of B). Because a t -norm is a two-place function from $[0, 1] \times [0, 1]$, we can consider every t -norm in (38) to be acting on a pair of membership functions; hence,

$$\mu_{B^l}(y) = \mu_{G^l}(y) * \sup_{\mathbf{x} \in U} [\mu_{X_1}(x_1) * \mu_{F_1^l}(x_1)] * \dots * [\mu_{X_p}(x_p) * \mu_{F_p^l}(x_p)]. \quad (39)$$

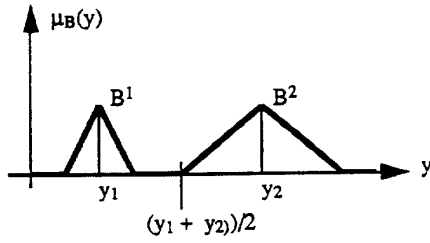


Fig. 30. Example for which mean of maximum defuzzification makes no sense.

By the monotonicity property of a t -norm, that supremum is attained when each term in braces in (39) attains its supremum.

Example 23: When the t -norm is the *product*, and all membership functions are Gaussian, then it is straightforward to carry out the supremum computations in (39) [53], [54]. The k th input fuzzy set and the corresponding rule antecedent fuzzy sets are assumed to have the following forms: $\mu_{X_k}(x_k) = \exp\{-1/2[(x_k - m_{X_k})/\sigma_{X_k}]^2\}$ and $\mu_{F_k^i}(x_k) = \exp\{-1/2[(x_k - m_{F_k^i})/\sigma_{F_k^i}]^2\}$. By maximizing the function

$$\mu_{Q_k^i}(x_k) \triangleq \mu_{X_k}(x_k)\mu_{F_k^i}(x_k) \quad (40)$$

we find that it is maximum at

$$x_{k,\max} = (\sigma_{X_k}^2 m_{F_k^i} + \sigma_{F_k^i}^2 m_{X_k}) / (\sigma_{X_k}^2 + \sigma_{F_k^i}^2). \quad (41)$$

In the special but important case when all input points for each input variable have the same level of uncertainty, the spreads of the input sets will be the same, in which case $\sigma_{X_k}^2$ in (41) is a constant. Usually we choose the mean of the fuzzy input sets, m_{X_k} , as the crisp measured input, x'_k ; hence, under these conditions, (41) simplifies to

$$x_{k,\max} = (\sigma_{X_k}^2 m_{F_k^i} + \sigma_{F_k^i}^2 x'_k) / (\sigma_{X_k}^2 + \sigma_{F_k^i}^2). \quad (42)$$

This formula can be interpreted as a prefiltering of the noisy data x'_k . A FLS has a built-in front-end mechanism for such prefiltering, namely the fuzzifier. A neural network does not.

Substituting (41) [or (42)] and (40) into (39), the latter becomes

$$\mu_{B^i}(y) = \mu_{G^i}(y) \prod_{k=1}^p \mu_{Q_k^i}(x_{k,\max}). \quad (43)$$

When the uncertainty of the input becomes zero (i.e., $\sigma_{X_k}^2 = 0$), then (42) reduces to the singleton case, i.e., $x_{k,\max} = x'_k$, for which each $\mu_{X_k}(x_{k,\max} = x'_k) = \exp\{-1/2[(x'_k - x'_k)/\sigma_{X_k}]^2\} = 1$ ($k = 1, \dots, p$), so that $\mu_{Q_k^i}(x_{k,\max}) = \mu_{F_k^i}(x_{k,\max}) = \mu_{F_k^i}(x'_k)$.

D. Defuzzifier

Defuzzification produces a crisp output for our FLS from the fuzzy set that is the output of the inference

block in Fig. 2. Many defuzzifiers have been proposed in the literature; however, there are no scientific bases for any of them (i.e., no defuzzifier has been derived from a first principle, such as maximization of fuzzy information or entropy); consequently, defuzzification is an art rather than a science. Because we are interested in engineering applications of FL, one criterion for the choice of a defuzzifier is *computational simplicity*. This criterion has led to the following candidates for defuzzifiers:

- 1) *Maximum Defuzzifier:* This defuzzifier examines the fuzzy set B and chooses as its output the value of y for which $\mu_B(y)$ is a maximum. It can lead to peculiar results or can get hung up. The former occurs for the situation depicted in Fig. 29(b), in which the maximum defuzzifier would choose $\theta = 40$ as the output of the FLS; this value totally ignores the fact that $\mu_B(\theta)$ is distributed from $\theta = 8$ to $\theta = 40$. The latter occurs for the situation depicted in Fig. 29(a), in which the maximum of $\mu_B(\theta)$ occurs for a range of θ values rather than at a unique point.
- 2) *Mean of Maxima Defuzzifier:* This defuzzifier examines the fuzzy set B and first determines the values of y for which $\mu_B(y)$ is a maximum. It then computes the mean of these values as its output. Unfortunately, it can also lead to some peculiar results. If the maximum value of $\mu_B(y)$ only occurs at a single point, then the mean of maximum defuzzifier reduces to the maximum defuzzifier, and our discussion about Fig. 29(b) applies to it. Fig. 30 depicts a situation where $\mu_B(y)$ is described by two separated triangles that both have the same peak amplitudes. The mean of the maximum defuzzifier assigns a value to the output of the FLS midway between the two triangles, at which point the membership function $\mu_B(y)$ has zero value. This makes no engineering sense.
- 3) *Centroid Defuzzifier:* This defuzzifier determines the center of gravity (centroid), \bar{y} , of B and uses this value as the output of the FLS. From calculus, we know that

$$\bar{y} = \left[\int_S y \mu_B(y) dy \right] / \left[\int_S \mu_B(y) dy \right] \quad (44)$$

where S denotes the support of $\mu_B(y)$. Frequently, S is discretized, so that \bar{y} can be approximated by the following formula which uses summations instead of integrations:

$$\bar{y} = \left[\sum_{i=1}^I y_i \mu_B(y_i) \right] / \left[\sum_{i=1}^I \mu_B(y_i) \right]. \quad (45)$$

The centroid defuzzifier is unique; however, it is usually difficult to compute. For an interpretation of \bar{y} as a conditional expectation, see [40].

Pacini and Kosko [56] prove that for product inference and additive combining of rules, \bar{y} can be computed using centroid information about the indi-

vidual M rules. While this result does not extend to other t -norms and t -conorms, it does provide some ad hoc justification for what is probably the most widely used form of defuzzification, height defuzzification.

- 4) *Height Defuzzifier*: Let \bar{y}^l denote the center of gravity of the fuzzy set B^l (which is associated with the activation of rule $R^{(l)}$). This defuzzifier first evaluates $\mu_{B^l}(y)$ at \bar{y}^l and then computes the output of the FLS as

$$y_h = \frac{\left[\sum_{l=1}^M \bar{y}^l \mu_{B^l}(\bar{y}^l) \right]}{\left[\sum_{l=1}^M \mu_{B^l}(\bar{y}^l) \right]}. \quad (46)$$

It is very easy to use (46) because the centers of gravity of commonly used membership functions are known ahead of time. Regardless of whether minimum (Fig. 17) or product (Fig. 18) inference is used, the center of gravity of B^l for a symmetric triangular consequent membership function is at the apex of the triangle; a Gaussian consequent membership function is at the center value of the Gaussian function; and, a symmetric trapezoidal membership function is at the midpoint of its support. Equation (46) is easily derived from calculus applied to the situation that is depicted in Fig. 31. Although (46) and (45) look alike, they are different.

Although (46) is easy to use, it suffers from a deficiency that is not at all obvious to the newcomer. Whereas y_h makes use of the entire shape of each antecedent's membership function, because this information is embodied in $\mu_{B^l}(\bar{y}^l)$, it does not make use of the entire shape of the consequent membership function. It only uses the center of the support, \bar{y}^l , of the consequent membership function. Regardless of whether or not the consequent membership function is very narrow, which indicates a very strong belief in that rule, or is very broad, which indicates much less belief in that rule, the height defuzzifier gives the same result. This has led to our last defuzzifier, the modified height defuzzifier.

- 5) *Modified Height Defuzzifier* [22], [73]: As in height defuzzification, we let \bar{y}^l denote the center of gravity of the fuzzy set B^l . The modified height defuzzifier first evaluates $\mu_{B^l}(y)$ at \bar{y}^l and then computes the output of the FLS as

$$y_{mh} = \frac{\left[\sum_{l=1}^M \bar{y}^l \mu_{B^l}(\bar{y}^l) / \delta^{l^2} \right]}{\left[\sum_{l=1}^M \mu_{B^l}(\bar{y}^l) / \delta^{l^2} \right]} \quad (47)$$

where δ^l is a measure of the spread of the consequent for rule $R^{(l)}$. For triangular and trapezoidal membership functions, δ^l could be the support of the triangle or trapezoid, whereas, for Gaussian membership functions, δ^l could be its standard deviation. The modified height defuzzifier is also easy to use, although the δ^l parameters must be specified as well as \bar{y}^l and $\mu_{B^l}(\bar{y}^l)$.

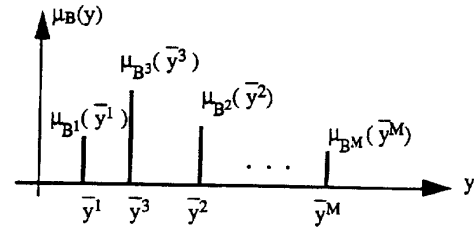


Fig. 31. Discrete masses that are located at the centroids of the membership functions for each rule. Each mass equals the peak value of the activated consequent membership function for $R^{(l)}$, $\mu_{B^l}(\bar{y}^l)$ (see (49)). An unactivated rule has zero value for $\mu_{B^l}(\bar{y}^l)$. Centroids do not have to appear in chronological order.

For additional discussions on defuzzifiers, see [20], [22], [69], [80], [81].

E. Possibilities

From our detailed discussions about the four elements which comprise the Fig. 2 FLS, we see that there are many possibilities to choose from. We must decide on the type of fuzzification (singleton or nonsingleton), functional forms for membership functions (triangular, trapezoidal, Gaussian, piecewise linear), parameters of membership functions (fixed ahead of time, tuned during a training procedure), composition (max-min, max-product), inference (minimum, product), and defuzzifier (centroid, height, modified height). Just choosing among the parenthetical possibilities leads to $2^{15} = 32\,768$ different FLS's. This demonstrates the richness of FLS's and that there is no such thing as *the* FLS.

F. Formulas for Specific FLS's: Fuzzy Basis Functions

The geometric interpretation we have provided for the inference block of our FLS (e.g., see Figs. 26–29) is informative; however, it does not provide us with a complete description of our FLS. For such a description, we need a mathematical formula that maps a crisp input \mathbf{x} into a crisp output $y = f(\mathbf{x})$. From Fig. 2, we see that such a formula can be obtained by following the signal \mathbf{x} through the fuzzifier, where it is converted into the fuzzy set $A_{\mathbf{x}}$, into the inference block, where it is converted into the fuzzy set B , and finally into the defuzzifier, where it is converted into $f(\mathbf{x})$. In order to write such a formula, we must make specific choices for fuzzifier, membership functions, composition, inference and defuzzifier.

Example 24: [71], [73] When we choose *singleton* fuzzification, *max-product* composition, *product inference*, and *height* defuzzification, leaving the choice of membership functions open, it is easy to show that

$$y = f_s(\mathbf{x}) = \frac{\left[\sum_{l=1}^M \bar{y}^l \prod_{i=1}^p \mu_{F_i^l}(x_i) \right]}{\left[\sum_{l=1}^M \prod_{i=1}^p \mu_{F_i^l}(x_i) \right]}. \quad (48)$$

To obtain (48), we started with (46) and substituted for $\mu_{B^l}(\bar{y}^l)$, where

$$\begin{aligned}\mu_{B^l}(\bar{y}^l) &= \mu_{A \rightarrow B}(\mathbf{x}', \bar{y}^l) = \left[\prod_{i=1}^p \mu_{F_i^l}(x'_i) \right] \mu_{G^l}(\bar{y}^l) \\ &= \prod_{i=1}^p \mu_{F_i^l}(x'_i),\end{aligned}\quad (49)$$

and we have assumed that membership functions are normalized, so that $\mu_{G^l}(\bar{y}^l) = 1$. Additionally, for notational simplicity, we have relabeled x'_i to x_i , so that we write $f_s(\mathbf{x}')$ as $f_s(\mathbf{x})$.

When we choose *singleton* fuzzification, *max-min* composition, *minimum* inference, and *height* defuzzification, then following the same procedure that we used to derive (48), we obtain

$$\begin{aligned}y &= f_s(\mathbf{x}) \\ &= \left[\sum_{l=1}^M \bar{y}^l \min_{i=1, \dots, p} \{ \mu_{F_i^l}(x_i) \} \right] / \\ &\quad \left[\sum_{l=1}^M \min_{i=1, \dots, p} \{ \mu_{F_i^l}(x_i) \} \right].\end{aligned}\quad (50)$$

When Gaussian membership functions are used, $\mu_{F_i^l}(x_i) = \exp\{-[(x_i - \bar{x}_i^l)/\sigma_i^l]^2\}$, where $i = 1, 2, \dots, p$ and $l = 1, 2, \dots, M$ (recall that p equals the dimension of \mathbf{x} , and M equals the number of rules).

Example 25: [53], [54] This is a continuation of Example 23. When we choose *nonsingleton* fuzzification, *max-product* composition, *product* inference, and the *Gaussian* membership functions for $\mu_{X_k}[x_k]$ and $\mu_{F_k^l}[x_k]$, then we know that $\mu_{B^l}(y)$ is given by (43). When we also choose *height* defuzzification, we must first determine which element from the fuzzy set B^l is going to be used by the defuzzifier. If, as assumed, $\mu_{G^l}(y)$ is Gaussian, then its value at $y = \bar{y}^l$ is unity; hence,

$$\mu_{B^l}(\bar{y}^l) = \prod_{k=1}^p \mu_{Q_k^l}(x_{k, \max}).\quad (51)$$

Substituting (51) into (46), we obtain

$$\begin{aligned}y &= f_{ns}(\mathbf{x}) \\ &= \left[\sum_{l=1}^M \bar{y}^l \prod_{k=1}^p \mu_{Q_k^l}(x_{k, \max}) \right] / \left[\sum_{l=1}^M \prod_{k=1}^p \mu_{Q_k^l}(x_{k, \max}) \right].\end{aligned}\quad (52)$$

serve the strong similarity between the *structures* of \bar{c} in (48) and $f_{ns}(\mathbf{x})$ in (52). \square

The FLS's in (48) and (52) can also be represented as drop the subscript "s" or "ns" on $f(\mathbf{x})$, for notational simplicity]

$$y = f(\mathbf{x}) = \sum_{l=1}^M \bar{y}^l \phi_l(\mathbf{x})\quad (53)$$

where $\phi_l(\mathbf{x})$ are called *fuzzy basis functions* (FBF's) [76] and are given by

$$\phi_l(\mathbf{x}) = \prod_{i=1}^p \mu_{F_i^l}(x_i) / \left[\sum_{l=1}^M \prod_{i=1}^p \mu_{F_i^l}(x_i) \right]\quad (54a)$$

for *singleton fuzzification*

$$\phi_l(\mathbf{x}) = \prod_{k=1}^p \mu_{Q_k^l}(x_{k, \max}) / \left[\sum_{l=1}^M \prod_{k=1}^p \mu_{Q_k^l}(x_{k, \max}) \right]\quad (54b)$$

for *nonsingleton fuzzification*

where $l = 1, 2, \dots, M$. We can now refer to our FLS as a *fuzzy basis function expansion*. Doing this is very useful, because it places a FLS into the more global perspective of function approximation. Remember though that the FBF's in (54) are valid only for very specific choices made about fuzzifier, membership functions, composition, inference and defuzzifier. Change any of these and (54a) and (54b) are no longer valid; but the interpretation of a FLS as a fuzzy basis function expansion still is. Formulas that are comparable to (54) can be derived for many other possibilities.

Although the index l on the FBF seems to be associated with a rule number, i.e., $l = 1, 2, \dots, M$, each FBF is affected by all of the rules because of the denominator in $\phi_l(\mathbf{x})$; hence, it is only partially correct to associate the j th FBF with the j th rule. Of course, if we add or remove a rule, thereby increasing or decreasing M , then we add or remove a FBF from the FBF expansion. It is in that sense that it is correct to associate the j th FBF with the j th rule.

The relationships between FBF's and other basis functions have been extensively studied in [29]. They are more general than radial basis functions, generalized radial basis functions, and hyper-basis functions [57]. For very special choices of their parameters and singleton fuzzification, they bare structural resemblance to generalized regression neural networks [63] and Gaussian sum approximations [2]. The latter two begin by assuming that the measured data is random and that an estimate is desired of another random quantity. This bares no resemblance to our starting point for a FLS where no assumption about randomness has been made.

The denominators in (54a) and (54b), which are a result of the height defuzzifier, serve to normalize the numerators of the FBF's. The numerators are radially symmetric; hence, one could also refer to our FBF's as normalized radial basis functions. Such basis functions were originally suggested by Moody and Darkin [52] as a means for sharing information across radial basis functions. Tao [68] has compared normalized and unnormalized radial basis functions, and demonstrated, by means of examples, the superiority of the former over the latter. It is important to remember that our FBF's are normalized not by abstraction, but rather by design of our overall FLS.

Example 26: What do the FBF's in (54a) and (54b) look like? We shall consider two situations, equally spaced and unequally spaced Gaussian antecedent membership functions and Gaussian fuzzy numbers. In order to visualize the FBF's in two dimensions, we choose $\dim \mathbf{x} = p =$

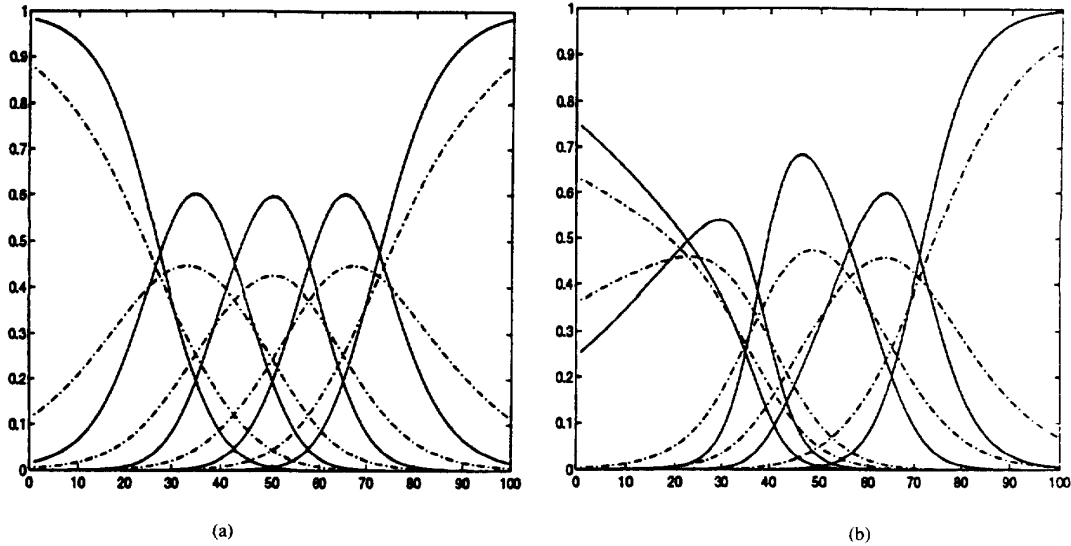


Fig. 32. Fuzzy basis functions for five rules in singleton (solid lines) and nonsingleton (dash-dot lines) FLS's. (a) Equally spaced and (b) unequally spaced Gaussian antecedent membership functions and Gaussian fuzzy numbers.

1, so that $\phi_l(\mathbf{x}) = \phi_l(x)$. We also choose the number of rules, M , equal to 5. Standard deviations for all Gaussian antecedent membership functions, as well as for the fuzzy input are set equal to 10.

In the equally spaced situation, we choose $m_{F_1^1} = 20$, $m_{F_1^2} = 35$, $m_{F_1^3} = 50$, $m_{F_1^4} = 65$, and $m_{F_1^5} = 80$. Fig. 32(a) depicts the five FBF's. Observe that the three interior FBF's are radially symmetric, whereas the two exterior FBF's are sigmoidal. These FBF's seem to combine the advantages of both radial basis functions, which are good at characterizing local properties, and sigmoidal neural networks, which are good at characterizing global properties. Observe, also, that the FBF's for nonsingleton fuzzification have longer tails and are broader than their singleton FBF counterparts. This means that more of them will be activated in the nonsingleton case than in the singleton case for a specific input value. For example, a vertical line at $x = 30$ in Fig. 32(a) intersects three of the singleton FBF's and four of the nonsingleton FBF's. *Input uncertainty activates more FBF's*, which means that decisions are more distributed in the nonsingleton case than in the singleton case.

Lest one believe that radial symmetry must occur for interior FBF's, we next consider the nonequally spaced situation, where we choose $m_{F_1^1} = 20$, $m_{F_1^2} = 25$, $m_{F_1^3} = 50$, $m_{F_1^4} = 62$, and $m_{F_1^5} = 80$. Fig. 32(b) depicts the five FBF's. Observe that the three interior FBF's are no longer radially symmetric, whereas the two exterior FBF's are still approximately sigmoidal. These figures should dispel the notion that fuzzy basis functions are radial basis functions. They are not; they are nonlinear functions of radial basis functions.

Equally spaced FBF's are possible only if the mean values (centers) of the antecedent membership functions

can be chosen by the designer. If these values are estimated by means of a training procedure, so that they adapt to the data that is associated with the rules, then unequally spaced FBF's are the norm rather than the exception. \square

Rules can come from numerical data or they can come from expert linguistic knowledge. Each rule contributes a basis function to the FBF expansion. It is convenient, therefore, to decompose $f_s(\mathbf{x})$ or $f_{ns}(\mathbf{x})$ into the sum of two terms, one associated with FBF's that are associated with rules that come from numerical data and the other that is associated with rules that come from linguistic information, i.e., $y = f(\mathbf{x}) = f_N(\mathbf{x}) + f_L(\mathbf{x})$. If we have a higher degree of belief in one set of rules over the other, then we can combine $f_N(\mathbf{x})$ and $f_L(\mathbf{x})$ in the following way: $y = f(\mathbf{x}) = \alpha f_N(\mathbf{x}) + (1 - \alpha) f_L(\mathbf{x})$ where $0 \leq \alpha \leq 1$. When $\alpha = 0$, then $y = f_L(\mathbf{x})$, which means, of course, that we are only using linguistic information. On the other hand, if $\alpha = 1$, then $y = f_N(\mathbf{x})$, which means we are only using numerical information. It is only when $0 < \alpha < 1$ that we are combining linguistic and numerical information. This is not the only way that we can combine linguistic and numerical information. One of its deficiencies is that it does not produce a strong coupling between the linguistic and numerical FBF's. Such coupling occurs when the denominators of all the basis functions are made dependent on both linguistic and numerical information. We illustrate how to do this next for singleton fuzzification. We begin by rewriting (53) as

$$\begin{aligned}
 y &= f(\mathbf{x}) \\
 &= \sum_{j=1}^M \bar{y}^j \phi_j(\mathbf{x}) = \sum_{i=1}^{M_N} \bar{y}_{N^i} \phi_{N,i}(\mathbf{x}) + \sum_{k=1}^{M_L} \bar{y}_{L^k} \phi_{L,k}(\mathbf{x})
 \end{aligned} \tag{55}$$

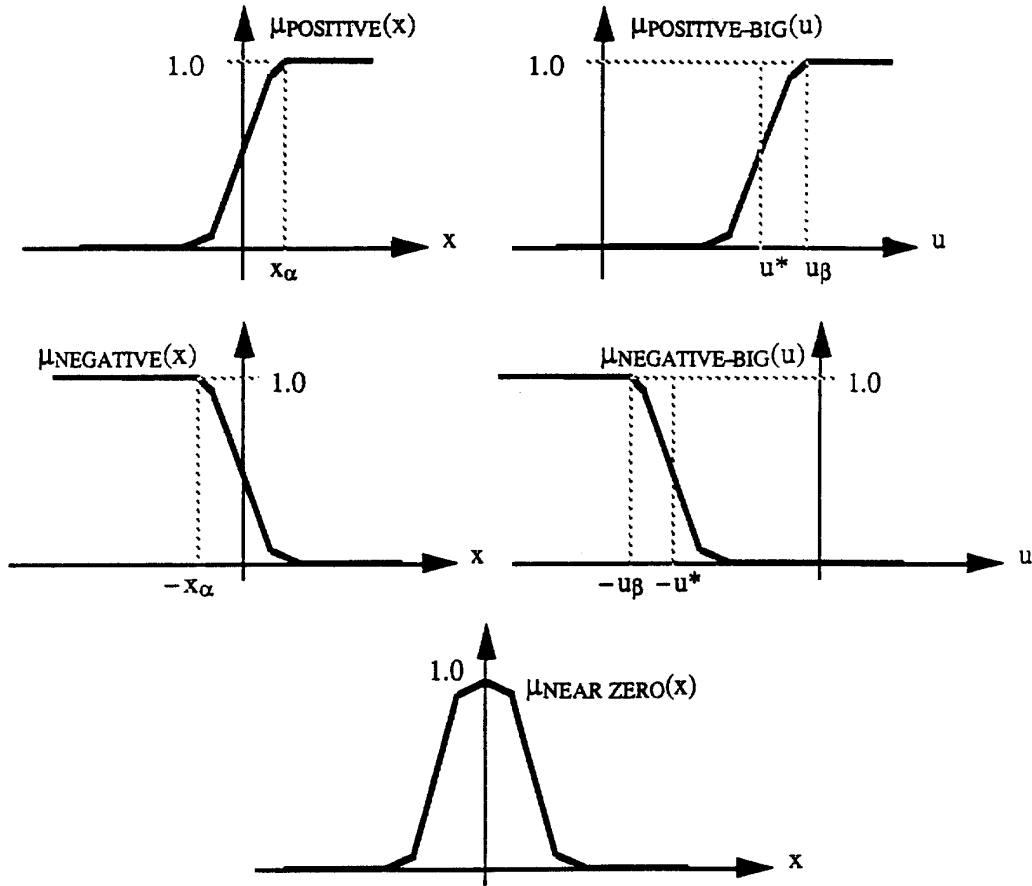


Fig. 33. Generic membership functions for the fuzzy sets *positive*, *positive big*, *negative*, *negative big*, and *near zero*. Step functions may be more appropriate for the fuzzy sets *positive* and *negative*.

where there are M_N FBF's associated with numerical data and M_L FBF's associated with linguistic information, and $M_N + M_L = M$. The FBF's are given by

$$\phi_{N,i}(\mathbf{x}) = \frac{\prod_{s=1}^p \mu_{F_s^i}(x_s)}{\left[\sum_{j=1}^M \prod_{s=1}^p \mu_{F_s^j}(x_s) \right]} \quad i = 1, 2, \dots, M_N \quad (56)$$

$$\phi_{L,k}(\mathbf{x}) = \frac{\prod_{s=1}^p \mu_{F_s^k}(x_s)}{\left[\sum_{j=1}^M \prod_{s=1}^p \mu_{F_s^j}(x_s) \right]} \quad k = 1, 2, \dots, M_L \quad (57)$$

Observe that the FBF's in (56) and (57) are normalized by information that is associated with both numerical and linguistic information, because their denominators depend on M , where $M = M_N + M_L$.

Example 27: Here we develop FBF's for the four linguistic rules in the ball and beam Example 18. Let us assume

that in addition to these four rules, we have D data training pairs; hence, there will be $M = D + 4$ FBF's. In order to simplify notation, let $x_1 = r(t)$, $x_2 = dr(t)/dt$, $x_3 = \theta(t)$, and $x_4 = d\theta(t)/dt$.

We need membership functions for the fuzzy sets *positive*, *positive big*, *near zero*, *negative*, and *negative big*. Fig. 33 depicts generic membership functions for all of these. Sigmoidal functions [e.g., $\mu(x) = 1/[1 + \exp(-\alpha x)]$] or shifted sigmoidal functions are used for $\mu_{\text{POSITIVE}}(x)$, $\mu_{\text{POSITIVE-BIG}}(u)$, $\mu_{\text{NEGATIVE}}(x)$, and $\mu_{\text{NEGATIVE-BIG}}(u)$, whereas a Gaussian function is used for $\mu_{\text{NEAR ZERO}}(x)$. The exact locations of the midpoint of the sigmoidal functions for $\mu_{\text{POSITIVE-BIG}}(u)$ and $\mu_{\text{NEGATIVE-BIG}}(u)$ depend on the domain of values over which u varies. Suppose that $u \in [-0.5, 0.5]$. Then the break point for the $\mu_{\text{POSITIVE-BIG}}(u)$ shifted sigmoidal function is chosen to be at $u^* = 0.4$, whereas the break point for the $\mu_{\text{NEGATIVE-BIG}}(u)$ shifted sigmoidal function is chosen to be at $u^* = -0.4$. Specific formulas for all these membership functions are left to the reader.

Let $\phi_{L,1}(\mathbf{x})$, $\phi_{L,2}(\mathbf{x})$, $\phi_{L,3}(\mathbf{x})$, and $\phi_{L,4}(\mathbf{x})$ be the FBF's that are associated with $R^{(1)}$, $R^{(2)}$, $R^{(3)}$, and $R^{(4)}$,

respectively; then from (57), we find, for example, that:

$$\phi_{L,1}(\mathbf{x}) = [\mu_{\text{POSITIVE}}(x_1)\mu_{\text{NEAR ZERO}}(x_2)\mu_{\text{POSITIVE}}(x_3) \mu_{\text{NEAR ZERO}}(x_4)] \left/ \sum_{j=1}^M \prod_{s=1}^4 \mu_{F_s^j}(x_s) \right. \quad (58a)$$

$$\phi_{L,4}(\mathbf{x}) = [\mu_{\text{NEGATIVE}}(x_1)\mu_{\text{NEAR ZERO}}(x_2)\mu_{\text{POSITIVE}}(x_3) \mu_{\text{NEAR ZERO}}(x_4)] \left/ \sum_{j=1}^M \prod_{s=1}^4 \mu_{F_s^j}(x_s) \right. \quad (58b)$$

where $M = D + 4$, and the $\mu_{F_s^j}(x_s)$, $j = 1, 2, 3, 4$, are easily identified with the numerator membership functions in each of the FBF's. We leave the construction of $\phi_{L,2}(\mathbf{x})$ and $\phi_{L,3}(\mathbf{x})$ to the reader.

Note that, although we have carefully described the membership functions in Fig. 33 for the control variable, u , they do not enter into the FBF's, because they are associated with the consequent of each rule. Examining the four rules and Fig. 33, we set \bar{y}_{L^k} in (55) ($M = 4$) to $\bar{y}_{L^1} = -u_\alpha$, $\bar{y}_{L^2} = u_\alpha$, $\bar{y}_{L^3} = u_\beta$, and $\bar{y}_{L^4} = -u_\beta$, respectively. It is only in the final formula for the FBF expansion that the control membership functions have any effect. \square

We have shown, therefore, that a FLS is a function approximator in which the basis functions of the approximator derive from either numerical or linguistic information. Each numerical rule leads to a fuzzy basis function, as does each linguistic rule. Unlike some of the classical basis functions (e.g., Laguerre polynomials, trigonometric functions) which are inherently orthogonal, fuzzy basis functions are not orthogonal. To date, FBF's are the only basis functions that can include numerical information as well as linguistic information; this makes them quite unique among all function approximation techniques.

G. Fuzzy Logic Systems Are Universal Approximators

How well does a FLS approximate an unknown function? This is an important question that is asked about all types of function approximations, including the popular feedforward neural network (FFNN). Cybenko [15], Hornik *et al.* [25], Hornik [24], as well as others (e.g., Blum [6]) demonstrated that a FFNN is a universal approximator, which means that a FFNN can uniformly approximate any real continuous nonlinear function to arbitrary degree of accuracy. Hornik *et al.* [25] used the Stone–Weirstrass theorem from real analysis to prove this result.

The same result has been proven using the Stone–Weirstrass theorem, by Wang and Mendel [76] and Wang [72] for a singleton FLS that uses product inference, product implication, Gaussian membership functions and height defuzzification. Kosko [37], [39] proved a similar result for an additive FLS, one that uses singleton fuzzification, centroid defuzzification, product inference and product implication (referred to by Kosko [36] as *correlation product inference*), using the concept of *fuzzy patches*. Mouzouris and Mendel [54] also do this, but for a nonsingleton FLS that uses a range of t -norms, *arbitrary* membership functions and

modified height defuzzification. Their proof does not use the Stone–Weirstrass theorem. See also [8] and [83].

Because no universal approximation theorem has been proved for arbitrary FLS's, we can expect to see many more FLS universal approximator theorems and proofs appearing in the FL literature, as has been the case for FFNN's.

A universal approximation theorem is an existence theorem. It helps to explain why FLS's are so successful in engineering applications; however, it does not tell us how to specify a FLS. The same is true for FFNN universal approximation theorems, which do not indicate how many layers of neurons should be used, how many neurons should be used in each layer, or how interconnected the neurons should be. Universal approximation theorems imply that by using *enough* layers, *enough* neurons in each layer, and *enough* interconnectivity, the FFNN can uniformly approximate any real continuous nonlinear function to arbitrary degree of accuracy.

We have already seen the enormous number of possibilities for FLS's. The design degrees of freedom that control the accuracy of a FLS are, number of inputs, number of rules and number of fuzzy sets for each input variable. Consider the i th input variable x_i , where $x_i \in U_i = [X_i^-, X_i^+]$. It is intuitively obvious that dividing the interval $[X_i^-, X_i^+]$ into 100 overlapping regions will lead to greater resolution, and consequently greater accuracy, than dividing the interval $[X_i^-, X_i^+]$ into, say, 10 overlapping regions. Kosko [37], [39] describes this in terms of smaller fuzzy patches versus larger fuzzy patches.

If there are p input variables, each of which is divided into r overlapping regions, then a complete fuzzy rule bank must contain p^r rules. As resolution parameter r increases, the size of the fuzzy rule bank becomes enormous (complex). There must, therefore, be a practical tradeoff between resolution and complexity. In actual practice, one almost never needs a complete fuzzy rule bank of p^r rules. This is because, in practical applications of FLS's there are large regions of the input space that are never seen during the actual operation of a system; hence, rules are not needed for such regions. In short, one important way to achieve high resolution and low complexity is to design the FLS using *representative data* that is collected for a specific application.

VI. DESIGNING FUZZY LOGIC SYSTEMS

Because of the large number of possibilities for FLS's, some guidelines are necessary for their practical designs. Linguistic rules are easily converted into their subset of FBF's, using fuzzy logic, as we have demonstrated in Example 27. Numerical rules, and their associated FBF's, must be extracted from numerical training data. Ultimately, after we have chosen the type of fuzzification, inference, implication, defuzzification, and membership functions, we must fix the parameters of the membership functions. Prior to 1992, all FLS's reported in the open literature fixed these parameters somewhat arbitrarily, e.g., the locations and spreads of the membership functions were chosen by

the designer independent of the numerical training data. Then, at the first IEEE Conference on Fuzzy Systems, held in San Diego, in 1992, three different groups of researchers (Wang and Mendel [77], Jang [27], and Horikawa *et al.* [23]) presented the same idea: *tune the parameters of a FLS using the numerical training data*. Since that time, quite a few adaptive training procedures have been published.

Space does not permit us to describe the different training approaches in detail. Here we very briefly outline two of them in connection with the following problem: we are given a collection of N input-output numerical data training pairs, $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \dots, (\mathbf{x}^{(N)} : y^{(N)})$, where \mathbf{x} is the vector input and y is the scalar output of our FLS. Our goal is to completely specify a FLS using the training data. See also [67] and [65].

For illustrative purposes, all designs assume singleton fuzzification, Gaussian membership functions, product inference and implication, and, height defuzzification; hence, our FLS is described by (53) and (54a). Note that the basic principles used in each design procedure carry over to many other FLS's.

To begin, we must explain how the training data can be interpreted as a collection of IF-THEN rules. Each rule is of the form IF u_1 is F_1^i and \dots and u_p is F_p^i , THEN v is G^i , $i = 1, 2, \dots, N$, where F_k^i are fuzzy sets, which are associated with the elements of $\mathbf{x}^{(i)}$, and are described by Gaussian membership functions, i.e., $\mu_{F_k^i}[x_k] = \exp\{-1/2[(x_k - m_{F_k^i})/\sigma_{F_k^i}]^2\}$, $k = 1, \dots, p$. Each design method establishes how to specify the parameters $m_{F_k^i}$ and $\sigma_{F_k^i}$ of these membership functions, as well as the centers of the consequent membership functions, the \bar{y}^i 's in (53), using the training pairs $(\mathbf{x}^{(1)} : y^{(1)}), (\mathbf{x}^{(2)} : y^{(2)}), \dots, (\mathbf{x}^{(N)} : y^{(N)})$.

In the least-squares design procedure [71], [73], [76] all of the parameters in the FBF's are fixed by the designer, and only the centers of the consequent membership functions, the \bar{y}^i 's in (53), are tuned. The number of FBF's (i.e., the number of rules), M , must also be specified. An orthogonal least-squares (OLS) procedure is used to select the most significant FBF's. The detailed formulas for the OLS procedure can be found in [73], [76]; they are based on the works of [10] and [11]. Linguistic information can be incorporated as a subset of the FBF's. The OLS procedure will then establish the simultaneous significance of linguistically based and data-based FBF's. For example, in the ball and beam example, the OLS procedure would establish whether or not the four linguistic rules contribute important FBF's. We have found that when there is not a lot of numerical training data, the linguistic information is very important; but, when there is a lot of numerical training data, linguistic rules become less important. The main drawback to this procedure is its computational intensity, due to the optimization of M . A second drawback is that it only seeks to optimize a subset of all the parameters that characterize a FLS.

In the backpropagation design procedure [71], [73], [77] all of the parameters in the FLS are optimized. There are

$M\bar{y}^i$ parameters, $M_p m_{F_k^i}$ parameters and $M_p \sigma_{F_k^i}$ parameters; hence, there are $M + 2M_p$ parameters that describe our FLS. For even modest values of M and p , $M + 2M_p$ can be a large number, e.g., 5 inputs and 50 rules lead to 550 parameters. One way to reduce the number of parameters is to assume $\sigma_{F_k^i} = \sigma$. Doing this reduces the number of parameters to $M + M_p + 1$. For our example, we would then have 301 parameters, which still may be a lot of parameters. Clearly, the dominant term in the number of parameters is the product M_p . In order to reduce the total number of parameters to a manageable number we must reduce the total number of rules, or the dimension of the input vector to the FLS, or both. The OLS design procedure can help to accomplish the former. Reducing the number of inputs, p , is very problem dependent, and is usually accomplished by trial and error. Finally, note that a feedforward neural network, which is also a universal approximator, also contains a lot of weights which must be optimized during a training procedure. Thousands of weights are not uncommon in practical applications.

The output of the FLS, when the k th training input vector, $\mathbf{x}^{(k)}$, is applied to it, is $f(\mathbf{x}^{(k)})$. The error between $f(\mathbf{x}^{(k)})$ and the desired output, $y^{(k)}$, is used as the basis for the backpropagation design procedure. Let $e_k \triangleq 1/2 \times [f(\mathbf{x}^{(k)}) - y^{(k)}]^2$, $k = 1, 2, \dots, N$. In the backpropagation design procedure, we fix the number of rules, M , and choose the design parameters, \bar{y}^i , $m_{F_k^i}$ and $\sigma_{F_k^i}$, such that e_k is minimized. This is done using a steepest descent algorithm in which the derivatives of e_k , with respect to all of the parameters, can be explicitly computed, because we can express e_k as an explicit function of the design parameters, using (53), (54a) and the formula for $\mu_{F_k^i}[x_k]$. In essence, this training procedure involves a forward and a backward flow of the data. In the forward flow, $f(\mathbf{x}^{(k)})$ is computed for a given $\mathbf{x}^{(k)}$. In the backward flow, $f(\mathbf{x}^{(k)}) - y^{(k)}$ is computed and propagated into the backpropagation equations for \bar{y}^i , $m_{F_k^i}$ and $\sigma_{F_k^i}$. Results in Chu and Mendel [12] demonstrate that a FLS can be trained much faster than a feedforward neural network.

Of course, a better backpropagation algorithm, or perhaps a totally different training algorithm, such as an extended Kalman filter [62], [26] or a Newton algorithm [78], may greatly speed up the training procedure for the neural network. Whatever better backpropagation algorithm, or different training algorithm that is used for a neural network can also be used by the FLS. Success of any neural network training algorithm depends on the initial values chosen for the weights. These weights have no physical meaning for the neural network; hence, they usually must be chosen randomly. The parameters of a FLS are associated with membership functions for physically meaningful quantities; hence, it is possible to obtain very good initial values for them.

C. Comments

It is possible to work with these two design methods in an iterative manner, thereby capturing the strong points of both methods. For example, by using the backpropagation

method first, we can obtain a good set of basis function parameters. Then the OLS design procedure can be used to reduce the number of basis functions, after which we can rerun the backpropagation design procedure in order to retune all of the FLS parameters.

The OLS and backpropagation design procedures are by no means the only ones that have been developed for tuning a FLS. Other procedures include: nearest neighborhood clustering procedure [73], which is useful when there is a lot of training data; determining membership function shapes that fit the input-output data, but for a FLS that is somewhat different from the one we have presented in this article [66]; and, supervised ellipsoidal learning [16] for tuning the parameters of an additive FLS. Parameters of a FLS can also be trained on-line using reinforcement learning [4]. Past issues of the *IEEE Transactions on Fuzzy Systems*, as well as the *Proceedings* of the 1992–1994 IEEE Conferences on Fuzzy Systems contain many other design procedures. See also [81] and the article by R. Jang in this issue of the *IEEE Proceedings*.

VII. CONCLUSIONS

We have demonstrated that a fuzzy logic system (FLS) is a nonlinear system that maps a crisp input vector into a crisp scalar output. We have provided mathematical formulas that describe this system, have shown that it can be expressed as a linear combination of fuzzy basis functions, and have explained that a FLS is a universal function approximator, which makes it a competitor to all other function approximators that share this property (e.g., feedforward neural networks). We have also demonstrated that the fuzzy basis function expansion is unique among all other basis function expansions, in that it can derive its basis functions in a unified manner from either numerical data (as can the other expansions) or linguistic knowledge (as can none of the other expansions).

The architecture of a FLS is determined by a careful understanding of fuzzy sets and fuzzy logic, and is rich with possibilities, i.e., there is no one FLS, there are many. As a user of a FLS, we must decide on the type of fuzzification (singleton or nonsingleton), functional forms for membership functions (triangular, trapezoidal, Gaussian, piecewise linear), parameters of membership functions (fixed ahead of time, tuned during a training procedure) composition (max-min, max-product), inference (minimum, product), and defuzzifier (centroid, height, modified height).

In order to derive our FLS we had to make the very strong but engineering-meaningful assumption of *causality*. Doing this caused us to deviate from the usual propositional logic definition of implication, so as to reach *engineering implications*, which were first introduced in the 1970's by engineers who successfully applied FL to control problems. These engineering implications lead to FLS's that work well in practice; however, they bare very little resemblance to more traditional logical implications. If this is viewed as a weakness of our FLS's, so be it. We prefer to view it as a triumph of engineering.

We conclude with a short comparison of FLS's and feedforward neural networks (FFNN's), because they can both be used to solve similar problems. Both are model free, i.e., all the information they are given is contained in some examples from which they are required to learn so as to give correct or successful outputs when new inputs are presented. They are given the same information and asked to perform the same task; but, linguistic knowledge can only be used by a FLS whereas it cannot yet be used by a FFNN. Such knowledge can be invaluable, especially if there is not a lot of numerical training data. Tuning a FLS can be done much faster than tuning a FFNN because the parameters of the FLS can be initialized smartly, whereas the parameters of FFNN must usually be initialized randomly. Finally, the fuzzification subsystem within the FLS lets us handle *uncertainty* in a very natural way, totally within the framework of FLS's. To date there does not seem to be a comparable way to handle uncertainty in a FFNN.

ACKNOWLEDGMENT

The author wishes to thank Joerg Hohensohn for producing Fig. 1, and George Mouzouris for producing Fig. 32. The author also wishes to thank the following people who reviewed the first draft of this paper at the request of the author: Lotfi Zadeh, Simon Haykin, Li-Xin Wang, John Yen, Stephen Chiu, George Mouzouris, Nilesh N. Karnik, and Joerg Hohensohn.

REFERENCES

- [1] C. B. Allendoerfer and C. O. Oakley, *Principles of Mathematics*. New York: McGraw-Hill, 1955.
- [2] D. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximations," *IEEE Trans. Automatic Control*, vol. AC-17, pp. 439–448, Aug. 1972.
- [3] B. Baets and E. Kerre, "The generalized modus ponens and the triangular fuzzy data model," *Fuzzy Sets and Syst.*, vol. 59, pp. 305–317, 1993.
- [4] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks*, vol. 3, pp. 724–740, Sept. 1992.
- [5] J. Bezdek and S. K. Pal, *Fuzzy Models for Pattern Recognition*. New York: IEEE Press, 1992.
- [6] E. K. Blum and L. K. Li, "Approximation theory and feedforward networks," *Neural Networks*, vol. 4, pp. 511–515, 1991.
- [7] P. P. Bonissone and K. S. Decker, "Selecting uncertainty calculi and granularity: An experiment in trading off precision and complexity," in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer, Eds. Amsterdam: Elsevier, 1986, pp. 217–247.
- [8] J. J. Buckley, "Universal fuzzy controllers," *Automatica*, vol. 28, pp. 1245–1248, 1992.
- [9] P. Cheeseman, "An inquiry into computer understanding," *Comp. Intell.*, vol. 4, pp. 57–142, 1988 (with 22 commentaries/replies).
- [10] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to nonlinear system identification," *Int. J. Contr.*, vol. 50, no. 5, pp. 1873–1896, 1989.
- [11] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, vol. 2, pp. 302–309, Feb. 1991.
- [12] P. Chu and J. M. Mendel, "First break refraction event picking using fuzzy logic systems," USC SIPI Rep. #221, Sep., 1992; also, *Proc. 2nd Int. Conf. on Fuzzy Systems*, Mar./Apr. 1993, San Francisco, CA; also, *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 255–266, 1994.

- [13] E. Cox, "Fuzzy fundamentals," *IEEE Spectrum*, pp. 58–61, Oct. 1992.
- [14] —, *The Fuzzy Systems Handbook*. Cambridge, MA: AP Professional, 1994.
- [15] G. Cybenko, "Approximation by superpositions of sigmoidal function," *Mathematics of Control, Signals, and Systems*, 1989.
- [16] J. A. Dickerson and B. Kosko, "Fuzzy function approximation with supervised ellipsoidal rules," *Proc. 1993 World Cong. on Neural Networks*, vol. 2, pp. 9–17, July 1993.
- [17] D. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. Orlando, FL: Academic, 1980.
- [18] D. Dubois and H. Prade, "A review of fuzzy set aggregation connectives," *Inf. Sciences*, vol. 36, pp. 85–121, 1985.
- [19] A. W. F. Edwards, *Likelihood*. London, UK: Cambridge Univ. Press, 1972.
- [20] D. P. Filev and R. R. Yager, "A generalized defuzzification method via BAD distributions," *Int. J. Intell. Syst.*, vol. 6, pp. 687–697, 1991.
- [21] J. Hauser, S. Sastry, and P. Kokotovic, "Nonlinear control via approximate input-output linearization: The ball and beam example," *IEEE Trans. Autom. Contr.*, vol. AC-37, pp. 392–398, Mar. 1992.
- [22] H. Hellendoorn and C. Thomas, "Defuzzification in fuzzy controllers," *J. Intell. Syst.*, vol. 1, pp. 109–123, 1993.
- [23] S. Horikawa, T. Furahashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with back-propagation algorithm," *IEEE Trans. Neural Networks*, vol. 3, pp. 801–806, Sept. 1992.
- [24] K. Hornik, "Some results on neural network approximation," *Neural Networks*, vol. 6, pp. 1069–1072, 1993.
- [25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
- [26] Y. Iiguni, H. Sakai, and H. Tokumaru, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter," *IEEE Trans. Signal Processing*, vol. 40, pp. 959–966, 1992.
- [27] J.-S. R. Jang, "Self-learning fuzzy controllers based on temporal back-propagation," *IEEE Trans. Neural Networks*, vol. 3, pp. 714–723, Sept. 1992.
- [28] A. Kaufman and M. M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications*. New York: Van Nostrand Reinhold, 1985.
- [29] Y. M. Kim and J. M. Mendel, "Fuzzy basis functions: Comparisons with other basis functions," USC SIPI Rep. No. 229, Jan. 1993; also, accepted for publication in *IEEE Trans. Fuzzy Systems*, 1995.
- [30] J. B. Kiszka, M. E. Kochanska, and D. S. Sliwiska, "The influence of some parameters on the accuracy of fuzzy model," in *Industrial Applications of Fuzzy Control*, M. Sugeno, Ed. Amsterdam: North-Holland, 1985, pp. 187–230.
- [31] —, "The influence of some parameters on the accuracy of a fuzzy model—Part I," *Fuzzy Sets Syst.*, vol. 15, pp. 111–128, 1985.
- [32] —, "The influence of some parameters on the accuracy of a fuzzy model—Part II," *Fuzzy Sets Syst.*, vol. 15, pp. 223–240, 1985.
- [33] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [34] S. G. Kong and B. Kosko, "Comparison of fuzzy and neural truck backer-upper control systems," *Proc. IJCNN-90*, vol. 3, pp. 349–358, June 1990.
- [35] B. Kosko, "Fuzziness versus probability," *Int. J. General Syst.*, vol. 17, pp. 211–240, 1990.
- [36] —, *Neural Network and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [37] —, "Fuzzy systems as universal approximators," *IEEE Int. Conf. Fuzzy Syst.*, pp. 1153–1162, 1992.
- [38] —, *Fuzzy Thinking: The New Science of Fuzzy Logic*. New York: Hyperion, 1993.
- [39] —, "Fuzzy systems as universal approximators," *IEEE Trans. Computers*, 1994; an early version appears in *Proc. 1st IEEE Int. Conf. on Fuzzy Systems*, pp. 1153–1162, Mar. 1992.
- [40] B. Kosko and J. A. Dickerson, "Function approximation with additive fuzzy systems," in *Theoretical Aspects of Fuzzy Control*, H. T. Nguyen, M. Sugeno, R. Tong, and R. Yager, Eds. New York: Wiley, 1994, chap. 12.
- [41] A. Lapedes and R. Farber, *Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling*, LA-UR-87-2662, Los Alamos, NM, 1987.
- [42] P. M. Larsen, "Industrial applications of fuzzy logic control," *Int. J. Man, Mach. Studies*, vol. 12, no. 1, pp. 3–10, 1980.
- [43] M. Laviolette and J. W. Seaman, Jr., "The efficacy of fuzzy representations and uncertainty," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 4–15, 1994.
- [44] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller, part I," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-20, no. 2, pp. 404–418, 1990.
- [45] —, "Fuzzy logic in control systems: Fuzzy logic controller, Part II," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-20, no. 2, pp. 419–435, 1990.
- [46] B. Y. Lindley, "Scoring rules and the inevitability of probability," *Int. Stat. Rev.*, vol. 50, pp. 1–26, 1982 (with seven commentaries/replies).
- [47] M. C. Mackey and L. Glass, "Oscillations and chaos in physiological control systems," *Science*, vol. 197, pp. 287–289, 1987.
- [48] J. Maiers and Y. S. Sherif, "Applications of fuzzy set theory," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-15, pp. 175–189, Jan./Feb. 1985.
- [49] E. H. Mamdani, "Applications of fuzzy algorithms for simple dynamic plant," *Proc. IEE*, vol. 121, pp. 1585–1588, 1974.
- [50] D. McNeill and P. Freilberger, *Fuzzy Logic: The Discovery of a Revolutionary Computer Technology and How it is Changing Our World*. New York: Simon and Schuster, 1992.
- [51] M. Mizumoto, "Comparison of various fuzzy reasoning methods," *Proc. 2nd IFSA Cong.*, Tokyo, Japan, pp. 2–7, July 1987.
- [52] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comp.*, vol. 1, pp. 281–294, 1989.
- [53] G. C. Mouzouris and J. M. Mendel, "Non-singleton fuzzy logic systems," *Proc. 1994 IEEE Conf. on Fuzzy Syst.*, Orlando, FL, June 1994.
- [54] —, *Non-Singleton Fuzzy Logic Systems: Theory and Application*, to be published.
- [55] D. Nguyen and B. Widrow, "The truck backer-upper: An example of self-learning in neural networks," *IEEE Control Syst. Magazine*, vol. 10, pp. 18–23, 1990.
- [56] P. Pacini and B. Kosko, "Comparison of fuzzy and Kalman filter target-tracking control systems," in *Neural Networks and Fuzzy Systems*. Englewood Cliffs, NJ: Prentice-Hall, pp. 379–406.
- [57] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, Sept. 1990.
- [58] R. F. Popoli, "Objective and subjective knowledge in estimation," Ph.D. dissertation, Univ. Southern California, Aug. 1989.
- [59] R. F. Popoli and J. M. Mendel, "Heuristically constrained estimation for intelligent signal processing," in *Advances in Geophysical Data Processing*, vol. 3, M. Siman and F. Aminzadeh, Eds. Greenwich, CT: JAI, 1989, pp. 107–134.
- [60] —, "Estimation using subjective knowledge with tracking applications," *IEEE Trans. Aerospace and Electron. Syst.*, vol. 29, pp. 610–623, 1993.
- [61] D. G. Schwartz, G. J. Klir, H. W. Lewis III, and Y. Ezawa, "Applications of fuzzy sets and approximate reasoning," *IEEE Proc.*, vol. 82, pp. 482–498, 1994.
- [62] S. Singhal and L. Wu, "Training feedforward networks with the extended Kalman filter," *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Processing*, pp. 1187–1190, May 1989.
- [63] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Networks*, vol. 2, pp. 568–576, 1991.
- [64] M. S. Stachowicz and M. E. Kochanska, "Fuzzy modeling of the process," *Proc. 2nd IFSA Congress*, Tokyo, Japan, pp. 86–89, July 1987.
- [65] M. Sugeno and G. T. Kang, "Fuzzy modeling and control of multilayer incinerator," *Fuzzy Sets and Syst.*, vol. 18, pp. 329–346, 1986.
- [66] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 7–31, 1993.
- [67] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control," *IEEE Trans. Syst., Man and Cybernetics*, vol. 15, pp. 116–132, 1985.

- [68] K. M. Tao, "A closer look at the radial basis function (RBF) networks," *Proc. 27th Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 1993.
- [69] T. Terano, K. Asai, and M. Sugeno, *Fuzzy Systems Theory and Its Applications*. New York: Academic, 1992.
- [70] N. Vadiee and M. Jamshidi, "A tutorial on fuzzy rule-based expert systems (FRBES) models. 1: Mathematical foundations," *J. Intell. and Fuzzy Syst.*, vol. 1, pp. 171-188, 1993.
- [71] L. X. Wang, "Analysis and design of fuzzy systems," Ph.D. dissertation, Univ. Southern California, Los Angeles, CA, 1992.
- [72] —, "Fuzzy systems are universal approximators," *Proc. IEEE Int. Conf. on Fuzzy Syst.*, San Diego, CA, 1992.
- [73] —, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [74] L. X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *Proc. 6th IEEE Int. Symp. on Intell. Contr.*, pp. 263-268, 1991.
- [75] —, "Generating fuzzy rules from numerical data, with applications," USC SIPI Rep. No. 169, 1991; also in *IEEE Trans. Systems, Man, and Cybern.*, vol. SMC-22, pp. 1414-1427, 1992.
- [76] —, "Fuzzy basis functions, universal approximation, and orthogonal least squares learning," *IEEE Trans. Neural Networks*, vol. 3, pp. 807-813, Sept. 1992.
- [77] —, "Back-propagation of fuzzy systems as nonlinear dynamic system identifiers," *Proc. IEEE Int. Conf. on Fuzzy Syst.*, pp. 1409-1418, San Diego, CA, 1992.
- [78] R. L. Watrous, "Learning algorithms for connectionist networks: Applied gradient methods of nonlinear optimization," *Proc. IEEE 1st Int. Conf. on Neural Networks*, vol. 2, pp. 619-627, June 1987.
- [79] R. R. Yager, "On the implication operator in fuzzy logic," *Inf. Sciences*, vol. 31, pp. 141-164, 1983.
- [80] R. R. Yager and D. P. Filev, "Template-based fuzzy systems modeling," *J. Intell. and Fuzzy Syst.*, vol. 2, pp. 39-54, 1994.
- [81] —, *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994.
- [82] J. Yen, R. Langari, and L. Zadeh, Eds., *Industrial Applications of Fuzzy Logic and Intelligent Systems*. New York: IEEE Press, 1995.
- [83] H. Ying, "Sufficient conditions on general fuzzy systems as function approximator," *Automatica*, vol. 30, pp. 521-525, 1994.
- [84] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.
- [85] —, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man and Cybern.*, vol. SMC-3, no. 1, pp. 28-44, 1973.

- [86] —, "The concept of a linguistic variable and its application to approximate reasoning," *Inf. Sciences*, vol. 8, pp. 199-249, and vol. 9, pp. 43-80, 1975.
- [87] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 2nd ed. Boston, MA: Kluwer, 1991.



Jerry M. Mendel (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the Polytechnic Institute of Brooklyn, Brooklyn, NY, in 1963.

He is Professor and Associate Chair of Electrical Engineering-Systems at the University of Southern California in Los Angeles, where he has been since 1974. His research interests include higher-order statistics, fuzzy logic, neural networks, applied-to-array processing, prediction of nonlinear time-series, and classification problems. He has published nearly 300 technical papers and is the author of five books, including *Lessons in Estimation Theory for Signal Processing, Communications and Control* (Prentice Hall, 1995) and *Maximum-Likelihood Deconvolution* (Springer-Verlag, 1990). He is the Editor of *A Prelude to Neural Networks: Adaptive and Learning Systems* as well as the author of *Kalman Filtering, and Other Digital Estimation Techniques* (IEEE Press). He was Editor of the IEEE Control System Society's IEEE TRANSACTIONS ON AUTOMATIC CONTROL and is presently an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS.

Dr. Mendel is a Distinguished Member of the IEEE Control Systems Society, a member of the IEEE Signal Processing Society, the Society of Exploration Geophysicists, the European Association for Signal Processing, Tau Beta Pi, Pi Tau Sigma, and Sigma Xi, and a registered Professional Control Systems Engineer in California. He was President of the IEEE Control Systems Society in 1986. He received the 1992 Signal Processing Society Paper Award for a paper on identification of nonminimum phase systems using higher-order statistics in the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING; the 1983 Best Transactions Paper Award for an article that appeared in the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING; the SEG 1976 Outstanding Presentation Award for a paper on maximum-likelihood deconvolution; a Phi Kappa Phi book award for his 1983 research monograph on seismic deconvolution; a 1985 Burlington Northern Faculty Achievement Award; a 1984 IEEE Centennial Medal; and the 1993 Service Award from the School of Engineering at USC.