INTRODUCTION TO FUZZY SET THEORY

by

Bart Kosko
University of Southern California

Fuzzy Logic Workshop
14 November 1990

Kosko, B., NEURAL NETWORKS AND FUZZY SYSTEMS, Volume I, Prentice-Hall, 1991

CHAPTER 1

NEURAL NETWORKS AND FUZZY SYSTEMS

From causes which appear similar, we expect similar effects. This is the sum total of all our experimental conclusions.

David Hume
An Inquiry Concerning Human Understanding

A learning machine is any device whose actions are influenced by past experiences.

Nils Nilsson
Learning Machines

Man is a species that invents its own responses. It is out of this unique ability to invent, to improvise, his responses that cultures are born.

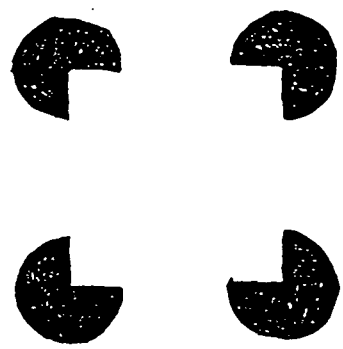
Ashley Montagu
Culture and the Evolution of Man

NEURAL AND FUZZY MACHINE INTELLIGENCE

This book examines how adaptive systems respond to stimuli. Systems map inputs to outputs, stimuli to responses. Adaptation or learning describes how data changes the system, how sample data changes system parameters, how training changes behavior.

Neural Pre-Attentive and Attentive Processing

The human visual system behaves as an adaptive system. Consider how it responds to this stimulus pattern:



What do we see when we look at the Kanizsa [1976] square? We see a square with bright interior. We see illusory boundaries. Or do we? We *recognize* a bright square. Technically we do not see it, because it is not there.

The Kanizsa square exists in our brain, not "out there" in physical reality on the page. Out there only four symmetric ink patterns stain the page.

In the terminology of eighteenth-century philosopher Immanuel Kant [1783-87], the four ink stains are *noumena*, "things in themselves." Light photons bounce off the noumena and stimulate our surface receptors, retinal neurons in this case. The noumena-induced sensation produces the Kanizsa-square *phenomenon* or perception in our brain. There would be no Kanizsa squares in the spacetime continuum without brains or brainlike systems to perceive them.

Today we understand many of the neural mechanisms of perception that Kant could only guess at. The realtime interaction of millions of competing and cooperating neurons produces the Kanizsa square illusion [Grossberg, 1987], and everything we "see."

We take for granted our high-speed, distributed, nonlinear, massively parallel pre-attentive processing. In our visual processing we pay no attention to how we segment images, enhance contrasts, or discount background luminosity. When we process sound we pay no attention to how our cochleas filter out high-frequency signals [Mead, 1989] or how our auditory cortex breaks continuous speech into syllables and words, compensates for rhythmic changes in speech duration, and detects and often corrects errors in pronunciation, grammar, and meaning. We likewise ignore our realtime pre-attentive processing in the other sense modalities, in smell, taste, touch, and balance.

We experience these pre-attentive phenomena, but we ignore them and cannot control or completely explain them. Natural selection has ensured only that we perform them, ceaselessly and fast.

Attention precedes recognition. We recognize segmented image pieces and parsed speech units. An emergent "search light," perhaps grounded in thalamic neurons [Crick, 1984], seems to selectively focus attention in as few as 70 to 100 milliseconds. We look, see, pay attention, then recognize.

Neural network theory studies both pre-attentive and attentive processing of stimuli. This leaves unaddressed the higher cognitive functions involved in reasoning, decision making, planning, and control. The asynchronous, nonlinear neurons and synapses in our brain perform these functions under uncertainty. We reason with scant evidence, vague concepts, heuristic syllogisms, tentative facts, rules of thumb, principles shot through with exceptions, and an inarticulable pantheon of inexact intuitions, hunches, suspicions, beliefs, estimates, guesses, and the like.

Natural selection evolved this uncertain cognitive calculus. Our cultural conditioning helps refine it. A fashionable trend in the West has been to denigrate this uncertainty calculus as illogical, unscientific, and nonrigorous. We even call it "fuzzy reasoning" or "fuzzy thinking." Modern philosophers [Churchland, 1981] often denigrate the entire cognitive framework as "folk psychology."

Yet we continue to use our fuzzy calculus. With it we run our lives, families, careers, industries, hospitals, courts, armies, and governments. In all these fields we employ the products of exact science, but as tools and decision aids. The final control remains fuzzy.

FUZZINESS AS MULTIVALUEDNESS

Fuzzy theory holds that all things are matters of degree. It mechanizes much of our "folk psychology." Fuzzy theory also reduces black-white logic and mathematics to special limiting cases of gray relationships. Along the way it violates black-white "laws of logic," in particular the law of noncontradiction *not-(A and not-A)* and the law of excluded middle *either A or not-A*, and yet resolves the paradoxes or antinomies [Kline, 1980] that these laws generate. Does the speaker tell the truth when he says he lies? Is set A a member of itself if A equals the set of all sets that are not members of themselves? Fuzziness also provides a fresh, and deterministic, interpretation of probability and randomness.

Mathematically fuzziness means multivaluedness [Rosser, 1952; Rescher, 1969] and stems from the Heisenberg position-momentum uncertainty principle in quantum mechanics [Birkhoff, 1936]. Three-valued fuzziness corresponds to truth, falsehood, and indeterminacy, or to presence, absence, and ambiguity. Multivalued fuzziness corresponds to degrees of indeterminancy or ambiguity, partial occurrence of events or relations.

Bivalent Paradoxes as Fuzzy Midpoints

Consider the bivalent paradoxes again. A California bumpersticker reads TRUST ME. Suppose instead a bumpersticker reads DON'T TRUST ME. Should we trust the driver? If we do, then, as the bumpersticker instructs, we do not. But if we don't trust the driver, then, again in accord with the bumpersticker, we do trust the driver. The classical liar paradox has the same form. Does the liar from Crete lie when he says that all Cretans are liars? If he lies, he tells the truth. If he tells the truth, he lies. Russell's barber is a man

in a town whose advertises his services with the logo "I shave all, and only, those men who don't shave themselves." Who shaves the barber? If he shaves himself, then according to his logo he does not. If he does not, then according to his logo he does. Consider the card that says on one side "The sentence on the other side is true," and says on the other side "The sentence on the other side is false."

The "paradoxes" have the same form. A statement S and its negation not-S have the same *truth-value* $t(S)$:

$$t(S) = t(\text{not-}S) \quad . \tag{1}$$

The two statements are both TRUE (1) or both FALSE (0). This violates the laws of noncontradiction and excluded middle. For bivalent truth tables remind us that negation reverses truth value:

$$t(\text{not-}S) = 1 - t(S) \quad . \tag{2}$$

So (1) reduces to

$$t(S) = 1 - t(S) \quad . \tag{3}$$

If S is true, if $t(S) = 1$, then $1 = 0$. $t(S) = 0$ also implies the contradiction $1 = 0$.

The fuzzy or multivalued interpretation accepts the logical relation (3) and, instead of insisting that $t(S) = 0$ or $t(S) = 1$, simply solves for $t(S)$ in (3):

$$2\,t(S) = 1 \quad , \tag{4}$$

or

$$t(S) = \frac{1}{2} \quad . \tag{5}$$

So the "paradoxes" reduce to literal half-truths. They represent in the extreme the uncertainty inherent in every empirical statement and in many mathematical statements. Geometrically, the fuzzy approach places the paradoxes at the midpoint of the 1-dimensional

unit hypercube [0, 1]. More general paradoxes reside at the midpoint of n-dimensional hypercubes, the unique point equidistant to all 2^n vertices.

Multivaluedness also resolves the classical *sorites* paradoxes. Consider a heap of sand. Is it still a heap if we remove one grain of sand? How about two grains? Three? If we argue bivalently by induction, we eventually remove all grains and still conclude that a heap remains, or that it has suddenly vanished. No single grain takes us from heap to non-heap. The same holds if we pluck out hairs from a nonbald scalp or remove 5%, 10%, or more of the molecules from a table or brain. We transition gradually, not abruptly, from a thing to its opposite. Physically we experience degrees of occurrence. In terms of statements about the physical processes, we arrive again at degrees of truth.

Suppose there are n grains of sand in the heap. Removing one grain leaves $n - 1$ grains and a truth value $t(S_{n-1})$ of the statement S_{n-1} that the $n - 1$ sand grains are a heap. The truth value $t(S_{n-1})$ obeys $t(S_{n-1}) < 1$ in general. $t(S_{n-1})$ may be close to unity, but we have some nonzero doubt d_{n-1} about the truth of the matter. (The argument still holds if there exist no doubting creatures in the universe.) For instance [Gaines, 1983],

$$t(S_n) = 1 - d_n \quad , \tag{6}$$

where $0 \leq d_n \leq d_{n-1} \leq \ldots \leq d_{n-m} \leq \ldots \leq 1$. So $t(S_{n-m})$ approaches zero as m increases to n. If we argue inductively, we can interpret the overall inference as the forward chain "(If S_n, then S_{n-1}) and (If S_{n-1}, then S_{n-2}) and ... and (If S_1, then S_0)." If we multiplicatively interpret the conjunction operator, then

$$t(S_n \longrightarrow S_{n-m}) = \prod_{k=0}^{m} (1 - d_{n-k}) \quad . \tag{7}$$

If we interpret the conjunction operator as the minimum operator, as discussed in the homework problems at the end of the chapter, then

$$t(S_n \longrightarrow S_{n-m}) = \min(1 - d_n, \ldots, 1 - d_{n-m}) \quad . \tag{8}$$

$$= 1 - \max(d_n, \ldots, d_{n-m}) \quad . \tag{9}$$

In both cases the implication truth value $t(S_n \longrightarrow S_0)$ equals zero (or some small number). We pay a truth-value fee for each application of *modus ponens*, of concluding B from A and $A \longrightarrow B$. The overall inference is vacuous. This reflects the everyday epistemological precept that the longer an explanation, the less we tend to trust it.

Fuzziness in the Twentieth Century

Logical paradoxes and the Heisenberg uncertainty principle led to the development of continuous or "fuzzy" logic in the 1920s and 1930s. Quantum theorists allowed for indeterminacy by including a third or middle truth value in the bivalent logical framework. The next step allowed degrees of indeterminacy, viewing TRUE and FALSE as the two limiting cases of the spectrum of indeterminacy.

Polish logician Jan Lukasiewicz [Rescher, 1969] first formally developed a three-valued logical system in the early 1930s. Lukasiewicz extended the range of truth values from {0, 1/2, 1} to all rational numbers in [0, 1], and finally to all numbers in [0, 1] itself. Logics that use the general truth function t: {Statements} $\longrightarrow$ [0, 1] define continous or "fuzzy" logics. Logicians refer to this system as L_1. The exercises at the end of the chapter develop Lukasiewicz's fuzzy logic.

In the 1930s quantum philosopher Max Black [1937] applied continuous logic componentwise to sets or lists of elements or symbols. Historically, Black drew the first fuzzy-set membership functions. Black called the uncertainty of these structures *vagueness*. Anticipating Zadeh's fuzzy set theory, each element in Black's multivalued sets and lists behaved as a statement in a continuous logic.

In 1965 systems scientist Lotfi Zadeh [1965] published the paper "Fuzzy Sets" that formally developed multivalued set theory, introduced the term fuzzy into the technical literature, and inaugurated a second wave of interest in multivalued mathematical structures, from systems to topologies. The recent emergence of fuzzy commerical products, as

well as new theory, has generated a third wave of interest in multivalued systems.

Zadeh extended the bivalent **indicator function** I_A of nonfuzzy subset A of X,

$$I_A(x) = \begin{cases} 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A \end{cases} \tag{10}$$

to a multivalued indicator or **membership function** $m_A : X \longrightarrow [0, 1]$. This allows us to combine such multivalued or fuzzy sets with the pointwise operators of indicator functions:

$$I_{A \cap B}(x) = \min(I_A(x), I_B(x)) \quad , \tag{11}$$

$$I_{A \cup B}(x) = \max(I_A(x), I_B(x)) \quad , \tag{12}$$

$$I_{A^c}(x) = 1 - I_A(x) \quad , \tag{13}$$

$$A \subset B \quad \text{iff} \quad I_A(x) \leq I_B(x) \text{ for all } x \text{ in } X \quad . \tag{14}$$

The membership value $m_A(x)$ measures the elementhood or degree to which element x belongs to set A:

$$m_A(x) = \text{Degree}(x \in A) \quad . \tag{15}$$

Just as the individual indicator values $I_A(x)$ behave as statements in bivalent propositional calculus, membership values $m_A(x)$ correspond to statements in a continuous logic. If A defines a fuzzy subset of the real line, as in Figure 1.7 below, then in principle we can graph $m_A : R \longrightarrow [0, 1]$ in two dimensions. In practice indicator functions I_A graph as step functions or rectangular pulses on the real line.

Sets as Points in Cubes

Fuzziness prevents logical certainty at the level of black-white axioms. This seems unsettling to some [Quine, 1981] and liberating to others.

At the system level fuzziness allows us to build computer chips and systems that "intelligently" control subways, automobile systems, and numerous consumer electronic and other devices. At this level fuzzy processing may resemble neural processing.

Neural networks and fuzzy systems process inexact information and process it inexactly. Neural networks recognize ill-defined patterns without an explicit set of rules. Fuzzy systems estimate functions and control systems with partial descriptions of system behavior. Experts may provide this heuristic knowledge, or, as we illustrate in Chapters 17 - 19, neural networks may adaptively infer it from sample data.

Neural and fuzzy systems share a more formal mathematical property. They share the same state space. A set of n neurons defines a sequence of n-dimensional continuous or "fuzzy" sets. The neurons emit bounded signals.

The neuronal signals range from some minimum value to some maximum value, say from 0 to 1. At each instant the n-vector of neuronal outputs defines a fuzzy unit or fit vector. Each fit value indicates the degree to which the neuron or element belongs to the n-dimensional fuzzy set.

The neuronal state space, the set of all possible neural outputs, equals the set of all n-dimensional fit vectors, the fuzzy power set. Both equal the unit hypercube $I^n = [0, 1]^n = [0, 1] \times \ldots \times [0, 1]$, the set of all vectors of length n and with coordinates in the unit interval $[0, 1]$. Chapter 17 discusses fuzzy systems and associative memories, which map unit cubes to unit cubes, fuzzy sets to fuzzy sets. We shall use this recent geometric view of *sets as points* [Kosko, 1987-90] throughout this book.

The 2^n vertices of I^n represent extremized neuronal-output combinations, as we often find in networks of competitive or laterally inhibitive neurons. Many feedback neural networks [Hopfield, 1984] drive initial states inside the unit cube to nearest vertices. These systems dynamically disambiguate fuzzy input descriptions by minimizing their fuzzy en-

tropy. The midpoint of the cube, where a fuzzy set A equals its own opposite A^c, has maximal fuzzy entropy, as we discuss in Chapter 16. The black-white vertices have minimal fuzzy entropy.

Proper fuzzy sets, nonvertex points, A violate the "laws" of noncontradiction and excluded middle: $A \cap A^c \neq \emptyset$ and $A \cup A^c \neq X$. In Chapter 16 we show that fuzzy entropy, the measure of fuzziness, balances the fuzzy count of the *overlap* $A \cap A^c$ and *underlap* $A \cup A^c$ in a simple ratio: $E(A) = \frac{M(A \cap A^c)}{M(A \cup A^c)}$.

There are 2^n bit vectors of length n. They define the vertices of I^n. So the vertices also represent the nonfuzzy power set of the n elements $x_1, \ldots, x_n$, the set of all nonfuzzy subsets of the n elements. The bit value 0 in the ith slot of a bit vector indicates the absence of element x_i in that subset. The bit value 1 indicates the presence of x_i in the subset. The bit vector (1 0 1 0 0) indicates the subset $\{x_1, x_3\}$ of set $\{x_1, x_2, x_3, x_4, x_5\}$.

Fit values equal the mebership values $m_A(x_i)$ discussed above. Fit values measure *partial set membership* or *degrees of elementhood.* The fit value 1/5 indicates that element x_i belongs only slightly to the fuzzy subset A. The fit value 1/2 indicates that x_i belongs to fuzzy set A as much as it does not—as much as it belongs to the complement fuzzy set A^c.

Consider the set X of two elements x_1 and x_2. The *power set* of X, denoted 2^X, contains the four subsets of X : $2^X = \{\emptyset, \{x_1\}, \{x_2\}, X\}$. These four nonfuzzy sets correspond to four bit vectors:

$$\begin{aligned} \emptyset &= (0\ 0) \\ \{x_1\} &= (1\ 0) \\ \{x_2\} &= (0\ 1) \\ X &= (1\ 1) \ . \end{aligned}$$

The fuzzy power set $F(2^X)$, which contains all continuum-many fuzzy subsets of X, corresponds to unit square. Figure 1.1 displays the fuzzy power set $F(2^X)$.

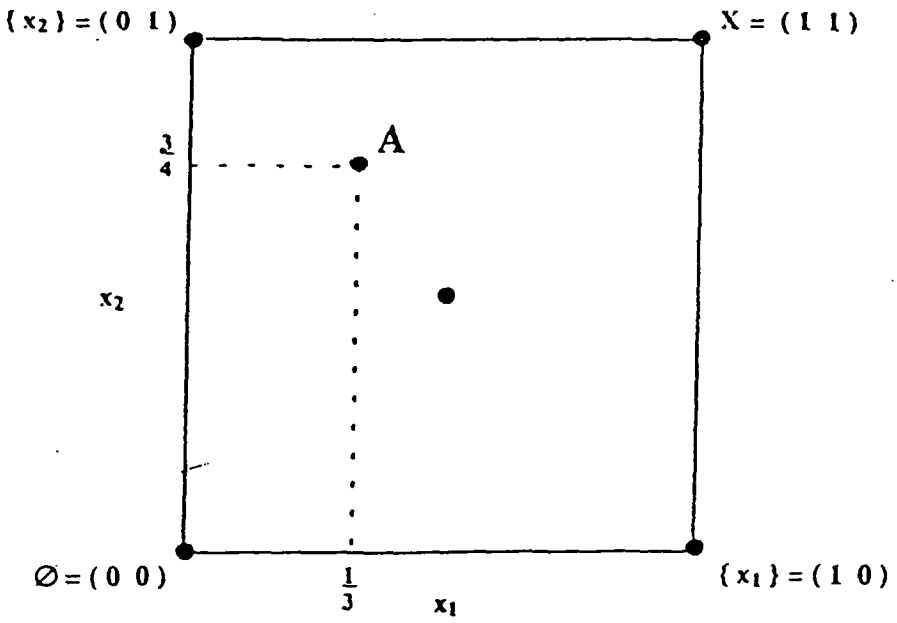


FIGURE 1.1 Fuzzy power set $F(2^X)$ of X corresponds to the unit square when $X = \{x_1, x_2\}$. The four nonfuzzy subsets in the nonfuzzy power set 2^X correspond to the four corners of the 2-cube. The fuzzy subset A correponds to the fit vector (1/3, 3/4) and to a point inside the 2-cube if $m_A(x_1) = 1/3$ and $m_A(x_2) = 3/4$. The midpoint M of the unit square corresponds to the maximally fuzzy set.

Figure 1.1 represents the fuzzy subset A as a point inside the 2-dimensional unit hypercube. If A has membership degrees or fit values $m_A(x_1) = 1/3$ and $m_A(x_2) = 3/4$—so x_1 belongs to A less than x_2 does—then A corresponds to the fit vector (1/3, 3/4).

The cube midpoint corresponds to the maximally fuzzy set M. The midpoint set M uniquely obeys the peculiar relation $M = M \cap M^c = M \cup M^c = M^c$, and so maximally violates the bivalent laws of noncontradiction and excluded middle. The classical paradoxes of logic and set theory correspond to midpoint phenomena. Note that the cube midpoint in Figure 1.1 is uniquely equidistant to all 2^2 vertices. The cube midpoint behaves as the black hole of set theory.

Subsethood and Probability

Elementhood represents a special case of *subsethood*. Subsethood measures the degree to which set A belongs to set B, the degree to which A is a subset of B. We denote this subsethood measure as $S(A, B)$:

$$S(A, B) = \text{Degree}(A \subset B) \quad . \tag{16}$$

Subsethood provides a unified set-theoretic framework for fuzziness and probability. For instance, in the simplest case A equals the singleton set $\{x_i\}$. Then the subsethood of $\{x_i\}$ in B equals the membership or elementhood value $m_B(x_i)$:

$$S(\{x_i\}, B) = m_B(x_i) \quad . \tag{17}$$

(17) follows directly from the Subsethood Theorem (22) below when we interpret $\{x_i\}$ as a bit vector with a 1 in the ith slot and 0s elsewhere.

Subsethood reveals the connection between fuzziness and randomness. Subsethood reduces probability to set theory. Randomness does not depend on the fuzziness or ambiguity of an event. It depends on the uncertainty between certain events. Randomness equals the uncertainty that arises when a nonfuzzy set B is partially contained in one of its own nonfuzzy subsets A. $S(A, B) = 1$ since A is a subset of B. But in general multivaluedness holds. The converse subsethood $S(B, A)$ is less than one but greater than zero:

$$0 < S(B, A) < 1 \quad . \tag{18}$$

Classical set theory implicitly forbids the strict inequalities in (18). The law of excluded middle dictates that every set either is or is not a subset of every other set. As a result, for centuries theorists have had to arbitrarily define probability as a frequency ratio or stipulate that it obeyed certain axioms. They could not *derive* probability from more fundamental concepts.

Fuzzy theory derives the axioms of the conditional probability measure $P(B|A)$,

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \quad , \tag{19}$$

the probability that B occurs given that A occurs, from the properties of the subsethood measure $S(A, B)$. If X defines the "sample space" of all elementary outcomes of an experiment, then X is a "sure event" since $P(X) = 1$. Then (19) implies that every probability $P(A)$ equals the conditional probability $P(A|X)$:

$$P(A) = P(A|X) \quad . \tag{20}$$

This identity reflects the general subsethood relationship

$$P(A) = S(X, A) \quad . \tag{21}$$

On the surface the subsethood relation (21) seems absurd. How can superset X belong to one of its own subsets? How can the whole be part of one of its own parts? X cannot *totally* belong to A unless $X = A$. But X can *partially* belong to A. The Subsethood Theorem in Chapter 16 proves that this partial containment depends directly on the overlap between X and A, the intersection $X \cap A$. Figure 1.2 illustrates the Pythagorean geometry of the Subsethood Theorem in three dimensions. The shaded hyper-rectangle defines $F(2^B)$, the fuzzy power set of B.

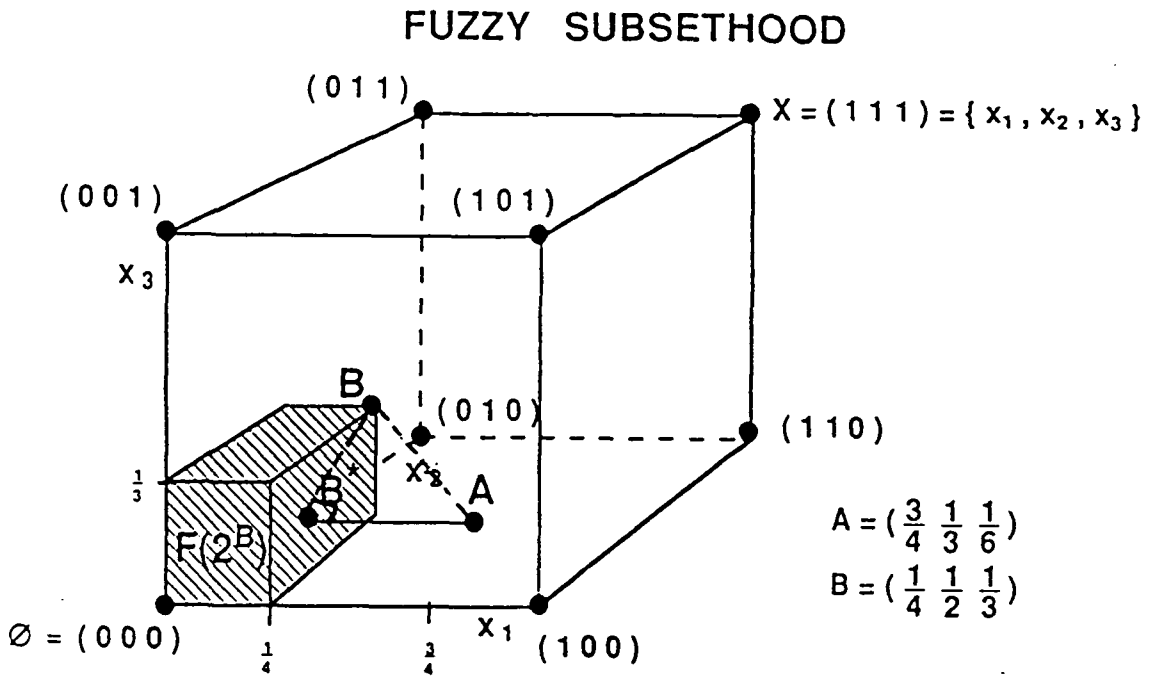


FIGURE 1.2 Subsethood Theorem in R^3. X contains 3 elements, x_1, x_2, and x_3, and 8 nonfuzzy subsets. Fuzzy subset $B = (1/4, 1/2, 1/3)$ contains infinitely many fuzzy subsets B' such that $S(B', B) = 1$. They define the shaded hyper-rectangle. $S(A, B) < 1$ since A lies outside the hyper-rectangle. The closer A to the hyper-rectangle, the larger the subsethood $S(A, B)$. B^* denotes the subset of B closest to A. B^* equals $A \cap B$ and uniquely defines an orthogonal or Pythagorean relationship between A and B.

The Subsethood Theorem relates $S(A, B)$ to the magnitudes of A, B, and $A \cap B$:

$$S(A, B) = \frac{M(A \cap B)}{M(A)} \quad . \tag{22}$$

The ratio in (22) resembles, behaves as, and generalizes the defining ratio (19) of conditional probability. $M(A)$ denotes the fuzzy count of fit vector A:

$$M(A) = a_1 + \ldots + a_n \quad . \tag{23}$$

$M(A)$ generalizes the classical cardinality count, which sums only 1s and 0s. In the infinite case appropriate integrals replace summations. (22) implies that the fuzzy entropy $E(A)$ of A equals the degree to which $A \cap A^c$ contains its own superset $A \cup A^c$: $E(A) = S(A \cup A^c, A \cap A^c)$.

In Figure 1.2, $A = (3/4, 1/3, 1/6)$ and $B = (1/4, 1/2, 1/3)$. Then the closest subset B^* to A that satisfies the total-subsethood condition

$$b_1^* \leq b_1, \ldots, b_n^* \leq b_n \quad , \tag{24}$$

corresponds to $B^* = (1/4\ 1/3\ 1/6)$, which also equals the pairwise minimum of A and B. (24) generalizes (14) above. As discussed in Chapter 16, the Subsethood Theorem ensures this in general:

$$B^* = A \cap B \quad . \tag{25}$$

(23) implies that $M(A) = 15/12 = 5/4$, and $M(A \cap B) = 3/4$. Then the Subsethood Theorem gives $S(A, B) = (3/4)/(5/4) = 3/5 = 60\%$.

Relative frequency provides the clearest example of between-set fuzziness. Suppose we flip a coin, draw balls from urns, or shoot at a target. The elementary events in X are trials. Each trial is successful or unsucessful. So X does not possess fuzzy subsets in its event space (its sigma-algebra). Each coin flips results in a head or a tail, not something in between. Suppose A defines the subset of successful trials. If X contains n trials, then A corresponds to a vertex of I^n and equals a bit vector of 1s and 0s. Suppose n_A successes out of n trials. 1s indicate successes, and 0s indicate failures. The event X equals total success, the bit vector of all 1s. X contains n successes. Then, since $A \cap X = A$, the Subsethood Theorem (22) gives

$$S(X, A) = \frac{n_A}{n} \quad . \tag{26}$$

Historically probability theorists have called the subsethood ratio in (26), or its limit, the "probability of success" or $P(A)$. This adds only a cultural tag. The success ratio n_A/n behaves no differently in its deterministic subsethood framework than it did in its "random" framework. The relative-frequency ratio still provides a stable estimate for probability values in our physical, engineering, economic, and gambling models. It still implies all the theorems it has always implied.

But we cannot derive the relative-frequency ratio from between-set relationships if we deny the strict inequality (18) and insist that subsethood is two-valued. Bivalence forces us to assume the ratio as a theoretical primitive.

Whether by design or by accident we have historically followed the bivalent path in mathematics for almost 3,000 years. Bivalence has simplified our formal frameworks but at a cost. It has led to logical paradoxes (bivalent contradictions), unexplained primitives, and "randomness" in a universe that seems to obey physical laws and where every event has causes.

THE DYNAMICAL SYSTEMS APPROACH TO MACHINE INTELLIGENCE: THE BRAIN AS A DYNAMICAL SYSTEM

Several engineering and scientific disciplines study how adaptive systems respond to stimuli. Electrical engineers study the topic as signal processing, nonlinear filtering, coding theory, circuit design, and adaptive control. Computer scientists study it as algorithm and automata theory, computer design, robotics, and artificial intelligence. Mathematicians study it as function approximation, statistical estimation, combinatorial optimization, and dynamical systems. Philosophers study it as epistemology, causality, and action. Biologists study it as neuroscience, biophysics, ecology, evolution, and population biology. Psychologists study it as reinforcement learning, psychometrics, and cognitive science. Economists study it as utility maximization, game theory, econometrics, and market equilibrium theory. Cultural anthropologists study it as culture.

We shall emphasize electrical engineering as we seek general principles of how adaptive

systems process information. We call these principles *machine intelligence* principles. We shall draw freely from the related fields of engineering and science.

The term *artificial intelligence* usually refers to the computer-scientific approach to machine intelligence. This approach emphasizes symbolic processing and tree search. AI has become the emblem for a popular computer-age view of the brain: *brain = computer.* This view ranges from classical science-fiction speculation (the computer HAL in *2001: A Space Odyssey*) to proposed space-based weapons systems.

We shall explore machine intelligence from a *dynamical-systems* viewpoint: *brain = dynamical system.* On this view a maple leaf falling to a potential-energy minimum on the ground better describes brain activity than does a computer executing instructions. The dynamical models we shall study are cast as large systems of differential or difference equations. The principles describe local or global interactions of nonlinear parallel processes.

Some of these machine-intelligence principles and mechanisms may explain natural phenomena and processes. Some already extend our theoretical and mathematical knowledge. But ultimately they should help us build smarter machines. They should give rise to new computational devices–electrical, optical, molecular, plasma, fluid, or other devices.

In this sense machine intelligence becomes an engineering discipline. Nearly a half century ago, Norbert Wiener [1948] outlined the first incarnation of such a machine-intelligence engineering. Wiener called it *cybernetics.*

We shall focus our analysis on artificial neural networks and fuzzy systems. These new, related systems represent broad classes of "machine-intelligent" adaptive systems. Chapters 2 - 6 describe neural network theory. Chapters 7 - 15 describe engineering applications of neural networks. Chapters 16 - 19 present a geometric theory of fuzzy sets and systems and its neural extension to adaptive fuzzy systems.

Neural and Fuzzy Systems as Function Estimators

Neural networks and fuzzy systems estimate input-output functions. Both are trainable

dynamical systems. Sample data shapes and "programs" their time evolution. Unlike statistical estimators, they estimate a function without a mathematical model of how outputs depend on inputs. They are *model-free* estimators. They "learn from experience" with numerical and, sometimes, linguistic sample data.

Neural and fuzzy systems encode sampled information in a parallel-distributed framework. Both frameworks are numerical. We can prove theorems to describe their behavior and limitations. We can implement neural and fuzzy systems in digital or analog VLSI circuitry or in optical-computing media, in spatial-light modulators and holograms.

Artificial neural networks consist of numerous, simple processing units or "neurons" that we can globally program for computation. We can program or train neural networks to store, recognize, and associatively retrieve patterns or database entries; to solve combinatorial optimization problems; to filter noise from measurement data; to control ill-defined problems; in summary, to estimate sampled functions when we do not know the form of the functions.

The human brain contains roughly 10^{11} or 100 billion neurons [Thompson, 1985]. That number approximates the number of stars in the Milky Way Galaxy, and the number of galaxies in the known universe. As many as 10^4 synaptic junctions may abut a single neuron. That gives roughly 10^{15} or 1 quadrillion synapses in the human brain. The brain represents an asynchronous, nonlinear, massively parallel, feedback dynamical system of cosmological proportions.

Artificial neural systems may contain millions of nonlinear neurons and interconnecting synapses. Future artificial neural systems may contain billions of real or virtual model neurons. In general no teacher supervises, stabilizes, or synchronizes these large-scale nonlinear systems.

Many feedback neural networks can learn new patterns and recall old patterns simultaneously, and ceaselessly. Supervised neural networks can learn far more input-output pairs, or stimulus-response associations, than the number of neurons and synapses in the network architecture. Since neural networks do not use a mathematical model of how a system's output depends on its input—since they behave as model-free estimators–we can apply the same neural network architecture, and dynamics, to a wide variety of problems.

Like brains, neural networks recognize patterns we cannot even define. We call this property *recognition without definition*. Who can define a tree, a pillow, or their own face to the satisfaction of a computer pattern-recognition system? These and most concepts we learn *ostensively*, by pointing out examples. We do not learn them as we learn the definition of a circle. We abstract these concepts from sample data, just as a child abstracts the color red from observed red apples, red wagons, and other red things, or as Plato abstracted triangularity from considered sample triangles.

Recognition without definition characterizes much intelligent behavior. It enables systems to generalize. Dogs, lizards, and slugs recognize multitudes of unforeseen, complex patterns without, of course, any ability to define them. Descriptive natural languages developed only yesterday in human evolution. Yet a great deal of modern philosophy, influenced by formal logic and behaviorist psychology, has insisted on concept definition preceding recognition or even discussion. Below we discuss how this insistence has helped shape the field of artificial intelligence and its emblem, the expert system.

Neural networks store pattern or function information with distributed encoding. They superimpose pattern information on the same associative-memory medium—on the many synaptic connections between neurons. Distributed encoding enables neural networks to complete partial patterns and "clean up" noisy patterns. So it helps neural networks estimate continuous functions.

Distributed encoding endows neural networks with fault tolerance and "graceful degradation." If we successively rip out handfuls of synaptic connections from a neural network, the network tends to smoothly degrade in performance, not abruptly fail. Computers and digital VLSI chips do not gracefully degrade when their components fail. Natural selection seems to have favored distributed encoding in brains, at least in sections of brains.

Neural networks, and brains, pay a price for distributed encoding: crosstalk. Distributed encoding produces crosstalk or interference between stored patterns. Similar patterns may clump together. New patterns may crowd out older learned patterns. Older patterns may distort newer patterns.

Crosstalk limits the neural network's storage capacity. Different learning schemes provide different storage capacities. The number of neurons bounds the number of patterns a

neural network can store reliably with the simplest unsupervised learning schemes. Even for more sophisticated supervised learning schemes, storage capacity ultimately depends on the number of network neurons and synapses, as well as on their function. *Dimensionality limits capacity.*

Biological neurons and synapses motivate the neural network's topology and dynamics. We interpret neurons as simple input-output functions, threshold switches for two-state neurons and asymptotic threshold switches for continuous neurons. We interpret synapses as adjustable weights. In neural analog VLSI chips [Mead, 1989], operational amplifiers model nonlinear neurons, and resistors model synapses.

The overall network behaves as an adaptive function estimator. Indeed commercial adaptive estimators are simple, usually linear, neural networks. These include antennae beam formers, high-speed modems, and echo-cancellers for long-distance telephone calls.

Neural Networks as Trainable Dynamical Systems

Neural networks *geometrize* computation. Network activity burrows a trajectory in a state space of large dimension, say R^n. Each point in the state space defines a snapshot of a possible neural network configuration.

The trajectory begins with a computational problem and ends with a computational solution. The user or the environment specifies the system's initial conditions, which define where the trajectory begins in the state space. In pattern learning, the pattern to be learned defines the initial conditions. In pattern recognition or recall, the pattern to be recognized defines the initial conditions.

Most of the trajectory corresponds to *transient* behavior or computations. Synaptic values gradually change to learn new pattern information. Neuronal outputs fluctuate.

The trajectory ends where the system reaches equilibrium, if it ever reaches equilibrium. In the simplest and rarest case, the equilibrium attractor is a fixed point of the dynamical system. Most popular neural networks converge to fixed points. In more complicated cases the equilibrium attractor is a limit cycle or limit torus. In Chapter 4 we discuss a crude

method for storing discrete time-varying patterns as limit cycles in feedback networks. The equilibrium attractors are *robust* or *structurally stable* if small perturbations do not distort or destroy them.

In general, and in most dynamical systems, the equilibrium attractor is *aperiodic* or chaotic. Once the network enters this region of the state space, it wanders forever without apparent structure or order. Yao and Freeman [1990] have used dynamical neural models and time-series data to argue that rabbit olfactory bulbs process odor information with chaotic attractors. As discussed in the homework problems, the function $x_{k+1} = c\, x_k\, (1 - x_k)$ behaves as a chaotic dynamical system for values of c near 4 and x values in the unit interval $[0,\ 1]$.

In Chapter 3 we discuss global Lyapunov functions for proving that certain feedback neural networks converge to fixed points from any initial conditions. Geometrically we can view the Lyapunov function as a surface sculpted by learned pattern information, as in Figure 1.3.

Figure 1.3 illustrates the geometry of fixed-point stability in feedback neural networks. Patterns behave as rocks on the rubber sheet of learning. The patterns, as well as "spurious" or unlearned patterns, dig out attractor basins in the state space and tend to rest at the local Lyapunov minimum of the attractor. The Lyapunov sheet changes shape as the system learns new patterns. Input patterns Q rapidly classify to nearest stored neighbors as if they were ball bearings rolling into local depressions in a gravity field. In a fixed-point attractor basins the state-trajectory balls stop at the local minima (or hover arbitrarily close to it). In limit-cycle attractors, the ball Q would rotate in an elliptical orbit inside the attractor basin. In limit-tori attractors, Q would cycle toroidally in the attractor basin, as if, in R^3, winding around the surface of a bagel. In chaotic attractors, Q would wander aperiodically within the attractor region.

In all these cases, the *number* of attractor basins does not affect the speed of convergence, the rate at which Q falls into the attractor basin. The dimensionality of the state space also does not in principle affect the convergence rate. In practice, Q converges exponentially quickly. This suggests that global stability may underlie our biological neural networks' ability to rapidly recognize patterns, generate answers, and exhibit appropriate

muscle reflexes independent of the amount of pattern information in our brains. Computer-type storage devices tend to slow as the number and complexity of patterns stored in them increases.

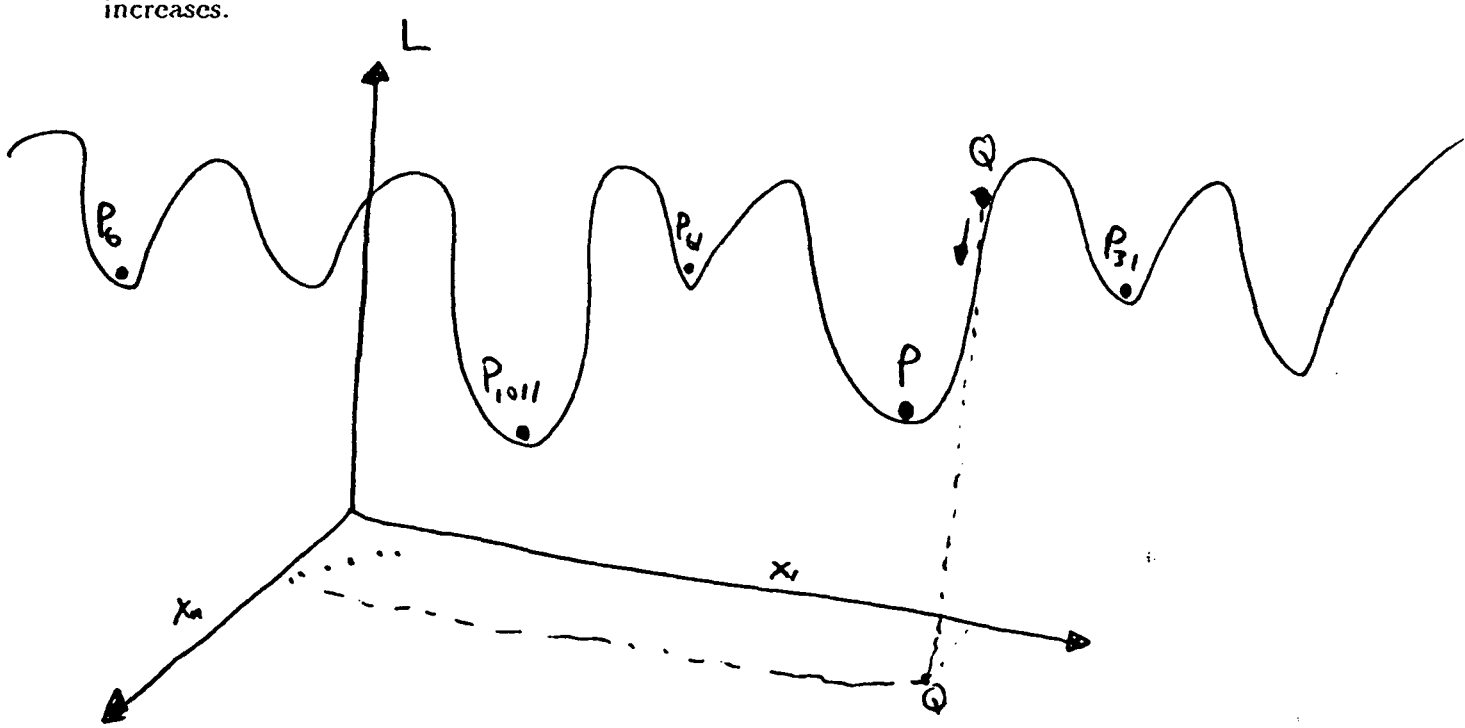


FIGURE 1.3 Global stability of a feedback neural network. Learning encodes the vector patterns P_1, P_2, ... by gradually sculpting a Lyapunov or "energy" surface in the augmented state space R^{n+1}. Input vector pattern Q rapidly "rolls" into the nearest attractor basin, where the system classifies Q as a learned pattern P or misclassifies Q as a spurious pattern. Q's descent rate does not depend on the number of stored patterns.

Mathematically we can describe the time evolution of the neural network by the (autonomous) dynamical system equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) \quad , \tag{27}$$

where the overdot denotes time differentiation. The state vector $\mathbf{x}(t)$ describes all neuronal and synaptic values of the neural network at time t. The neural network reaches *steady state* when

$$\dot{\mathbf{x}} = \mathbf{O} \quad , \tag{28}$$

holds indefinitely or until new stimuli perturb the system out of equilibrium. Neural computation seeks to identify the steady-state condition (28) with the solution of a computational problem, whether in pattern recognition, image segmentation, optimization, or numerical analysis.

We can locally linearize $\mathbf{f}$ by replacing $\mathbf{f}$ with its Jacobian matrix of partial derivatives $\mathbf{J}$. The eigenvalues of $\mathbf{J}$ describe the system's local behavior about an equilibrium point. For instance, if all eigenvalues have negative real parts, then the local equilibrium is a fixed point and the system converges to it exponentially quickly. More abstractly, generalized eigenvalues or *Lyapunov exponents* describe the underlying dynamical contraction and expansion that may produce chaos.

We can classify neural network models according as they learn with supervision (pattern-class information) and according as they contain closed synaptic loops or feedback. Figure 1.4 provides a rough taxonomy of several popular neural network models.

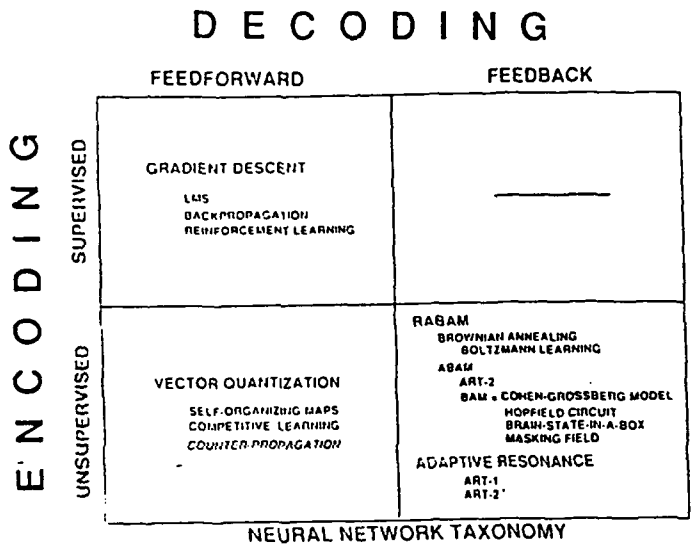


FIGURE 1.4 Taxonomy of neural network models.

Supervised feedforward models provide the most tractable, most applied neural models. We discuss these stochastic gradient systems in Chapter 5, and mention recent attempts to extend these supervised systems into the feedback domain. Unsupervised feedback models provide the most biologically plausible, but mathematically most complicated, models. These networks simultaneously learn and recall patterns. Both neurons and synapses change state when these systems learn and when they recall, recognize, or reconstruct pattern information. Chapter 6 proves global stability for many of these adaptive dynamical systems in the RABAM Theorem. Unsupervised feedforward neural networks tend to converge to locally sampled pattern-class centroids, as discussed in Chapters 4,6, and 9.

Fuzzy Systems and Applications

Fuzzy systems store banks of fuzzy associations or commonsense "rules." A fuzzy traffic controller might contain the fuzzy association "If traffic is heavy in this direction, then keep

the light greeen longer." Fuzzy phenomena admit degrees. Some traffic configurations are *heavier* than others. Some green-light durations are *longer* than others. The single fuzzy association (HEAVY, LONGER) encodes all these combinations.

Fuzzy systems are even newer than neural systems. Yet already engineers have successfully applied fuzzy systems in many commercial areas. Fuzzy systems "intelligently" automate subways; focus cameras and camcorders; tune color televisions, control automobile transmissions, cruise controllers, and emergency braking systems; defrost refrigerators and control air conditioners; automate washing machines and vacuum sweepers; guide robot-arm manipulators; invest in securities; control traffic lights, elevators, and cement mixers; recognize Kanji characters; select golf clubs; even arrange flowers.

Most of these applications originated in Japan, though fuzzy products are sold and applied throughout the world. Until very recently, Western scientists, engineers, and mathematicians have overlooked, discounted, or even attacked early versions of fuzzy theory, usually in favor of probability theory. Below, and especially in Chapter 16, we examine this philosophical resistance in more detail and present a new geometrical theory of continuous or "fuzzy" sets and systems.

Fuzzy systems "reason" with parallel associative inference. When asked a question or given an input, a fuzzy system fires each fuzzy rule in parallel, but to different degree, to infer a conclusion or output. Thus fuzzy systems reason with sets, "fuzzy" or continuous sets, instead of bivalent propositions. This generalizes the Aristotelian logical framework that still dominates science and engineering. In one second a digital fuzzy VLSI chip may execute thousands, perhaps millions, of these parallel-associative set inferences. We measure such chip performance in FLIPS, fuzzy logical inferences per second.

Fuzzy systems estimate sampled functions from input to output. They may use linguistic (symbolic) or numeric samples. An expert may articulate linguistic associations such as (HEAVY, LONGER). Or a fuzzy system may adaptively infer and modify its fuzzy associations from representative numerical samples.

In the latter case, neural and fuzzy systems naturally combine. The combination resembles an adaptive system with sensory and cognitive components. Neural parameter estimators embed directly in an overall fuzzy architecture. Neural networks "blindly"

generate and refine fuzzy rules from training data. Chapters 17-19 describe and illustrate these adaptive fuzzy systems.

Adaptive fuzzy systems learn to control complex processes very much as we do. They begin with a few crude rules of thumb that describe the process. Experts may give them the rules. Or they may abstract the rules from observed expert behavior. Successive experience refines the rules and, usually, improves performance.

Chapter 18 applies this adaptive cognitive process to backing up a truck-and-trailer rig to a loading dock. (A supervised neural system can also solve this problem, though at much greater computational cost. So far the truck-and-trailer dynamical system has eluded mathematical characterization.) The fuzzy system quickly learns a set of governing fuzzy rules as it samples actual truck-and-trailer trajectories. Additional training samples improve only marginally the fuzzy system's performance. This property is better experienced than explained. As an exercise for the reader, you might try backing your car into the same parking space five times from five different starting positions.

INTELLIGENT BEHAVIOR AS ADAPTIVE MODEL-FREE ESTIMATION

Below we discuss neural and fuzzy systems in more detail. First we examine the properties neural and fuzzy systems share with us and, more broadly, with all intelligent systems. These properties reduce to the single abstract property of adaptive model-free function estimation: *Intelligent systems adaptively estimate continuous functions from data without specifying mathematically how outputs depend on inputs.* We now elaborate this thesis.

A function f, denoted $f: X \rightarrow Y$, maps an input domain X to an output range Y. For every element x in the input domain X, the function f uniquely assigns the element y in the output range Y. We denote this unique assignments as $y = f(x)$. $f(x) = x^3$ defines a cubic function. $f(x_1, x_2, x_3) = (x_1, x_2, x_1^2 - x_2^2)$ defines a "saddle" or hyperbolic-paraboloid vector function in physical or 3-dimensional space R^3. Pressure is

a function of temperature, mass of energy ($e = m\ c^2$), gravity of mass, erosion of gravity, consumption of income. Functions define causal hypotheses. Science and engineering paint our pictures of the universe with functions.

Humans, animals, reptiles, amphibians, and others also estimate functions. We all respond to stimuli. We associate responses with stimuli. We associate actions with scenarios, class labels with patterns, effects with causes. Equivalently, we map stimuli to responses.

Mathematically, all these systems transform inputs to outputs. The transformation defines the input-output function $f : X \rightarrow Y$. Indeed the transformation defines the system. We can operatively characterize any system—atomic, molecular, biological, ecological, economic or legal, geological, galactic—by how it transforms input quantities into output quantities.

We call system behavior "intelligent" if the system emits appropriate, problem-solving responses when faced with problem stimuli. The system may use an associative memory embedded in the resistive network of an analog VLSI chip or embedded in the synaptic webs of its brain. Or the system may use a mathematical algorithm to search a decision tree, as in computer chess programs.

Generalization and Creativity

Intelligent systems also *generalize*. Their behavioral repertoires exceed their experience. Eighteenth-century philosopher David Hume saw why: Intelligent systems associate similar responses with similar stimuli. Small input changes produce small output changes. Hence they estimate *continuous* functions. The pilot lands the airplane at night the same way if only a few of the runway lights are out or if the new runway differs only slightly from more familiar runways. The leopard stalks like prey in like ways in like circumstances. Each minnow in a school smoothly adjusts its swimming behavior to the position of its smoothly moving neighbors.

Function continuity accounts for much novel or creative behavior, if not all of it. We call system behavior "novel" if the system emits appropriate responses when faced with new or

unexpected stimuli. "Novel ideas," says behaviorist psychologist B.F. Skinner [1953], are "responses never made before under the same circumstances....Novel contingencies generate novel forms of behavior." Usually these new stimuli resemble known or learned stimuli, and our responses usually resemble known responses.

Geometrically, when systems generalize or "create" they map stimulus balls to response balls. Consider a known stimulus-response pair (x, y). Stimulus x defines a point in the stimulus space S, the set of all possible stimuli for the problem at hand. In practice S often corresponds to the real Euclidean vector space R^n. Response y defines a point in the response space R, which may correspond to R^p.

Now imagine a stimulus ball $\mathbf{B_x}$ centered about stimulus x and a response ball $\mathbf{B_y}$ centered about response y. All the stimuli x' in $\mathbf{B_x}$ resemble stimulus x. The closer stimulus x' is to stimulus x, and hence the smaller the distance $d(\mathbf{x}', \mathbf{x})$, the more x' resembles x. The responses y' in $\mathbf{B_y}$ behave similarly.

Suppose $\mathbf{y} = f(\mathbf{x})$ for some unknown continuous function $f : R^n \rightarrow R^p$. The function f defines the sampled system. Suppose further that f generates the response ball from the stimulus ball: $\mathbf{B_y} = f(\mathbf{B_x})$. So for every similar response y' in $\mathbf{B_y}$, we can find some similar stimulus x' in $\mathbf{B_x}$ such that $\mathbf{y}' = f(\mathbf{x}')$. Formally f maps the stimulus ball *onto* to the response ball.

(We use the term "ball" loosely. Technically, $f(\mathbf{B_x})$ need not define an open ball in R^p. Thus we measure $\mathbf{B_y}$ with a volume measure below in (29). The *Open Mapping Theorem* in real analysis [Rudin, 1974] implies that all bounded onto linear transformations f map the open ball $\mathbf{B_x}$ to some set in R^p that contains the open ball $\mathbf{B_y}$, where $\mathbf{y} = f(\mathbf{x})$. At best we can only locally approximate most system transformations f as linear transformations.)

Then we can measure the creativity $C_{B_x}(f)$ of system f, given the stimulus ball $\mathbf{B_x}$, by the volume ratio

$$C_{B_x}(f) = \frac{V(\mathbf{B_y})}{V(\mathbf{B_x})} \quad , \tag{29}$$

where the V operator (Lebesgue measure) measures ball volume in R^n or R^p. C_{B_x}, crude as it is, captures many intuitions. It also resembles a spectral transfer function.

Consider the extreme cases of infinite and zero creativity. For a fixed nondegenerate response ball $\mathbf{B_y}$, as the stimulus ball $\mathbf{B_x}$ contracts to $\mathbf{x}$, the creativity measure $C_{B_x}(f)$ increases to infinity. (The point $\mathbf{x}$ has zero volume.) $C_{B_x}(f)$ also increases to infinity if the stimulus ball is constant and nondegenerate but the response ball $\mathbf{B_y}$ expands without bound as its radius approaches infinity. In both cases an infinitely creative system emits infinitely many responses when presented with, in the first case, a vanishingly small number of stimuli or, in the second case, a fixed set of stimuli.

Infinite creativity need not represent infinite problem solving. The reinforcing environment selects "solutions" from our varied or creative responses. Most creative solutions are impractical. We can emit creative responses without solving problems or contributing to our genetic fitness. Sometimes we call these responses "art" or "play."

At the other extreme, zero creativity occurs when the response ball $\mathbf{B_y}$ vanishes or when the stimulus ball expands without bound as its radius grows to infinity. In the first case the system f is a constant function. It maps all stimuli in $\mathbf{B_x}$ to a single value $\mathbf{y}$ in R^p. Such an f is "dumb" or "dull." In the second case, for an infinite-radius stimulus ball $\mathbf{B_x}$, the stimuli overwhelm the system's response repertoire. Such systems resemble classical pattern-recognition devices that are sensitive only to well-defined, well-centered patterns (faces, zip codes, bar codes).

Small variations in input provide the simplest novel stimuli. The physical or cultural environment may produce these variations. Or we may systematically produce them as grist for our analytical mill. We may vary stimuli to solve a crossword puzzle, to fit physical variables to astronomical data, or to formulate and resolve a mathematical conjecture.

We are all forward-looking creatures. We tend not to see the gradual causal chains that precede our every action, idea, and innovation. Even Beethoven's Fifth Symphony appears less a discontinuity when we examine Beethoveen's notebooks and a variety of preceding musical compositions by him and by other composers.

Variation and selection drive biological and cultural evolution. Physical and cultural environments drive the selection process. Function continuity, and other factors, drive variation.

Nature and man experiment with local variations of input parameters. This generates

local variations of output parameters. Then selection processes filter the new outputs. More accurately, they filter the corresponding new systems. We call the new systems "winners" or "fit" if they pass through the selection filters, "losers" or "unfit" if they do not pass through.

Variation and selection *rates* may vary, especially over long stretches of geological or cultural time. Different perturbed processes unfold at different speeds. So some evolutionary stretches appear more "punctuated" than others [Gould, 1980]. This means some measures of change—ultimately time derivatives—are nonlinear. It does not mean that the underlying input-output functions are discontinuous.

Learning as Change

Intelligent systems also *learn* or *adapt*. They learn new associations, new patterns, new functional dependencies. They sample the flux of experience and encode new information. They compress or quantize the sampled flux into a small, but statistically representative, set of prototypes or exemplars. Sample data changes system parameters.

"Learning" and "adaptation" are linguistic gifts from antiquity. They simply mean *parameter change*. The parameters may be numerical weights in an inner-product sum, average neurotransmitter release rates at synaptic junctions, or gene (allelle) frequencies at chromosonal loci in populations.

"Learning" usually applies to synaptic changes in brains or nervous systems, coefficient changes in estimation or control algorithms or devices, or resistor changes in analog VLSI circuitry. Sometimes we synonymously apply "adaptation" to the same changing parameters. In evolutionary theory "adaptation" applies to positive changes in gene frequencies [Wilson, 1975].

In all cases learning means change. Formally, a system learns if and only if the system parameter vector or matrix has a nonzero time derivative. In neural networks we usually represent the synaptic web by an adjacency or connection matrix M of numerical synaptic values. Then learning is *any change in any synapse*:

$$\dot{M} \neq 0 \ . \tag{30}$$

We can learn well or learn badly. But we cannot learn without changing, and we cannot change without learning.

Learning laws describe the synaptic dynamical system, how the system encodes information. They determine how the synaptic-web process unfolds in time as the system samples new information. This shows one way that neural networks compute with dynamical systems. Neural networks also identify neural activity with dynamical systems. This allows the systems to decode information.

In principle we can harness any dynamical system to encode and decode some information. We can view a kinetic swirl of molecules, a joint population of lynxes and rabbits, and a solar system as systems that transform input states to output states. Initial conditions and perturbations encode questions. Transient behavior computes answers. Equilibrium behavior provides answers. In the extreme case we can even view the universe as a dynamical-system "computer." A godlike entity may choose Big-Bang initial conditions, and there are infinitely many, to encode certain information or to ask certain questions. The dynamical system computes as the universe expands transiently. Universal equilibrium behavior could represent the computational output: a heat-death pattern or perhaps a periodic or chaotic oscillation of expansion and contraction.

Consider mowing a lawn of green grass. The mower "teaches" the lawn the short-grass pattern. The lawn consists of a parallel field of grass blades. Grass blades learn what they are cut. The lawn behaves as a semi-permanent, yet plastic, information storage medium. It tolerates faults and distributes cut patterns over large numbers of parallel units. We can mow our name in the lawn, and read or decode it from a rooftop. In principle we can encode all known information in a sufficiently big lawn. Eventually the lawn will forget this information if we do not resample comparable data, if we do not re-mow the lawn to a similar shape.

Ultimately learning provides only a means to some computational end. Neural networks learn patterns or functions or probability distributions to recognize future patterns, filter

future input streams of data, or solve future combinatorial optimization problems. Fuzzy systems learn associative rules to estimate functions or control systems. We climb the ladder of learning and kick it away when we reach the roof of computation. We care how the learned parameter performs in some computational system, not how it was learned, just as we applaud the piano recital and not the practice sessions.

Neural and fuzzy systems ultimately learn some unknown probability (subsethood) function $p(\mathbf{x})$. The probability density function $p(\mathbf{x})$ describes a distribution of vector patterns or signals $\mathbf{x}$, a few of which the neural or fuzzy system samples. When a neural or fuzzy system estimates a function $f: X \to Y$, it in effect estimates the joint probability density $p(\mathbf{x}, y)$. Then solution points $(\mathbf{x}, f(\mathbf{x}))$ should reside in high-probability regions of the input-output product space $X \times Y$.

We do not need to learn if we know $p(\mathbf{x})$. We could proceed directly to our computational task with techniques from numerical analysis, combinatorial optimization, calculus of variations, or any other mathematical discipline. The need to learn varies inversely with the quantity of information or knowledge.

Sometimes the patterns cluster into exhaustive decision classes $D_1, \ldots, D_k$. The decision classes may correspond to high-probability regions or "mountains." (If the pattern vectors are two-dimensional, then $p(\mathbf{x})$ defines a hilly surface in three-dimensional space R^3.) Then class boundaries correspond to low-probability regions or "valleys" on the probability surface.

Supervised learning uses class-membership information. *Unsupervised* learning does not. An unsupervised learning system processes each sample $\mathbf{x}$ but does not "know" that $\mathbf{x}$ belongs to class D_i and not to class D_j. Unsupervised learning uses unlabelled samples. Neither supervised nor unsupervised learning systems assume knowledge of the underlying probability density function $p(\mathbf{x})$.

Suppose we want to train a speech-recognition system at an international airport. We want the German lightbulb to light up when someone speaks German to the speech-recognition system, the Hindi lightbulb to light up when someone speaks Hindi, and so on. The system learns as we feed it training waveforms or spectrograms.

We supervise the learning if we label each training sample as German, Hindi, Japanese,

etc. We may do this to compute an error. If the English lightbulb lights up for a German sample, we may algorithmically punish the system for this misclassification.

An unsupervised system learns only from the raw training samples. We do not indicate language class labels. Unsupervised systems adaptively cluster like patterns with like patterns. The speech-recognition system gradually clumps German speech patterns together. In competitive learning, for instance, the system learns class centroids, centers of pattern mass.

Unsupervised learning may seem difficult and unreliable. But most learning is unsupervised, since we do not know accurately the labels of most sample data, especially in realtime processing. Every second our biological synapses learn without supervision on a single pass of noisy data.

SYMBOLS VS. NUMBERS: RULES VS. PRINCIPLES

We all share another property: We cannot articulate the mathematical rules that describe, if not govern, our behavior. We can ask a violinist how she plays, and she can tell us. But her answer will not be a mathematical function. In general her answer will not enable us to reproduce her behavior.

All lifeforms recognize vast numbers of patterns. The most primitive patterns relate to how an organism forages, avoids predators, and reproduces [Wilson, 1975].

On this planet only man articulates rules, and he articulates very few. We articulate some rules in grammar, common law, and science ("physical laws"). Yet all our natural languages, living and dead, and all our systems of law have culturally evolved without conscious design and not in accord with articulated principles [Hayek, 1973]. To some extent this also holds for our accumulated knowledge of medical, biological, and social science.

There have been exceptions, and the exceptions have helped create the field of artificial intelligence. Last century linguists developed the articulated language *Esperanto*. Mathematician Giuseppe Peano similarly devised the language *Interlingua*. A few fans still learn

and speak *Esperanto* and *Interlingua*, but far fewer speak them than speak Latin. This century computer scientists have consciously created the many computer programming languages. Today programmers frequently use *C*, Pascal, and even Fortran, and infrequently use Algol and Jovial.

Computer scientists developed artificial intelligence in large part around the computer language Lisp, or *list processing*, and more recently around Prolog, or *logic programming*. Lisp and Prolog process symbols and lists of symbols. Symbolic logic, the bivalent propositional and predicate calculi, underlies their processing structure.

Expert System Knowledge as Rule Trees

AI systems store and process propositional *rules*. The rules are logical implications: IF A, THEN B. They associate actions B with conditions A. The rule antecedents and consequents correspond to step functions defined on their universes of discourse. One part of the input space activates or "fires" A as true, and the other part does not activate A.

Collections of rules define "knowledge bases" or "rulebases." The rule $A \rightarrow B$ locally structures the knowledge of A and B as a logical implication. The knowledge base globally structures the rules as an acyclic tree (or forest). The logical-implication paths $A \rightarrow B \rightarrow C \rightarrow D \rightarrow \ldots$ flow from the tree's root nodes or antecedents to its leaf nodes or consequents. The term *knowledge base* stems from the computer-scientific term *database*. Because of the tree structure of knowledge bases, we might more accurately call them *knowledge trees*. Chapter 4 discusses fuzzy cognitive maps, which use feedback and vector-matrix operations to convert knowledge trees to knowledge networks.

Knowledge engineers search the knowledge tree to enumerate logical paths. Path enumeration defines the *inference* process. Forward-chaining inference proceeds from knowledge-tree antecedents to consequents. Backward-chaining inference proceeds from consequents or observations to plausible antecedents or hypotheses. Forward-chaining inference answers what-if question. It derives effects from causes. Backward-chaining inference answers why or how-come questions. It suggests causes for observed effects. Path-

enumeration complexity increases nonlinearly with the number of rules stored. Realtime path enumeration in large knowledge trees may be combinatorially prohibitive, requiring heuristic or approximate search strategies [Pearl, 1984].

Knowledge engineers acquire, store, and process the bivalent rules as symbols, not as numerical entities. This often allows knowledge engineers to rapidly acquire structured knowledge from experts and to efficiently process it. But it forces experts to articulate the propositional rules that approximate their expert behavior, and this they can rarely do.

Symbolic vs. Numeric Processing

Symbolic processing fits naturally in the brain-as-computer framework. Language strings model thoughts or shortterm memory. Rules and relations between language strings model longterm memory. Programming replaces learning. Logical inference replaces time evolution and nonlinear dynamics. Feedforward flow through knowledge trees replaces feedback equilibria.

But we cannot take the derivative of a symbol. We require a sufficiently continuous function. Symbol processing precludes mathematical analysis in the traditional senses of engineering and the physical sciences. The symbolic framework allows us to quickly represent structured knowledge as rules, but prevents us from directly applying the tools of numerical mathematics and from directly implementing AI systems in large-scale integrated circuits.

Figure 1.5 provides a taxonomy of model-free estimators. The taxonomy divides the knowledge type into structured (rule-like) and unstructured types and divides the framework into symbolic or numeric. All entries define model-free estimators because users need not state how outputs mathematically depend on inputs.

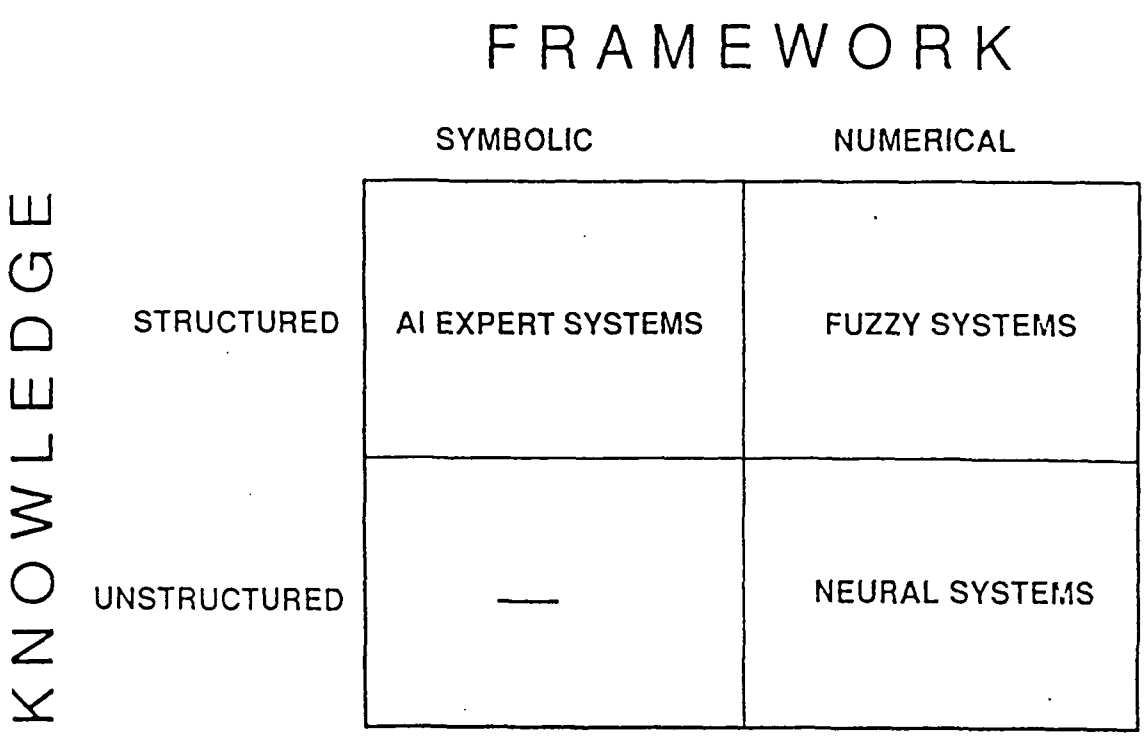


FIGURE 1.5 Taxonomy of model-free estimators. User need not state how system outputs explicitly depend on inputs.

Figure 1.5 outlines the advantages and disadvantages of machine-intelligent systems. AI expert systems exploit structured knowledge, when knowledge engineers can acquire it, but store and process it outside the analytical and computational numerical framework.

Neural networks exploit their numerical framework with theorems, efficient numerical algorithms, and analog and digital VLSI implementations. But neural networks cannot directly encode structured knowledge. They superimpose several input-output samples (x_1, y_1), (x_2, y_2), ..., (x_m, x_m) on a black-box web of synapses. Unless we check all input-output cases, we do not know what the neural system has learned, and in general we do not know what it will forget when it superimposes new samples (x_k, y_k) atop the old. We cannot directly encode the commonsense traffic-light rule "If traffic is heavy in one direction, keep the light green longer in that direction." Instead we must present the system with a sufficiently large set of input-output pairs, combinations of numerical traffic-density measurements and green-light duration measurements.

Fuzzy Systems as Structured Numerical Estimators

Fuzzy systems directly encode structured knowledge but in a numerical framework. We enter the fuzzy association (HEAVY, LONGER) as a single entry in a FAM-rule matrix. Each entry defines a fuzzy associative memory (FAM) "rule" or input-output transformation. In Chapter 17 we discuss the fuzzy control of an inverted pendulum. Figure 1.6 shows a bank of FAM rules sufficient to control an inverted pendulum.

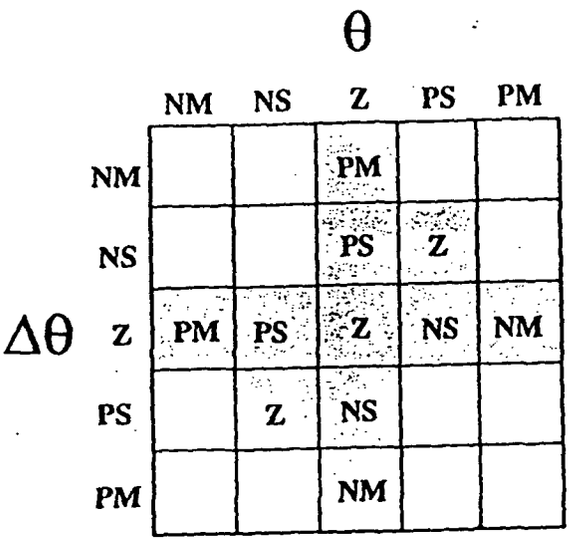


FIGURE 1.6 Bank of FAM rules to control an inverted pendulum. Each entry in the FAM matrix defines a fuzzy association between output fuzzy sets and paired input fuzzy sets.

θ, $\Delta\theta$, and v define fuzzy variables. Fuzzy variables θ and $\Delta\theta$ define the system's state variables. The angle fuzzy variable θ measures the angle the pendulum shaft makes with the vertical and ranges from -90 to 90. The angular velocity fuzzy $\Delta\theta$ variable measures

the instantaneous rate of change of angle values. In practice it measures the difference between successive angle values. Output fuzzy variable v measures the current to a motor controller that adjusts the pendulum shaft.

Each fuzzy variable can assume five fuzzy-set values: Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), and Positive Medium (PM). The entry at the center of the FAM matrix defines the steady-state FAM rule: "IF θ = ZE AND $\Delta\theta$ = ZE, THEN v = ZE."

We usually define the fuzzy-set values NM, ..., PM as trapezoids or triangles over regions of the real line. For the fuzzy angle variable θ, we can define ZE as a narrow triangle centered at the zero value in the interval [−90, 90]. Then the angle value 0 belongs to the fuzzy set ZE to degree 1. The angle values 3 and -3 may belong to ZE only to degree 0.6. Figure 1.7 shows seven trapezoidal fuzzy-set values assumed by fuzzy variable θ.

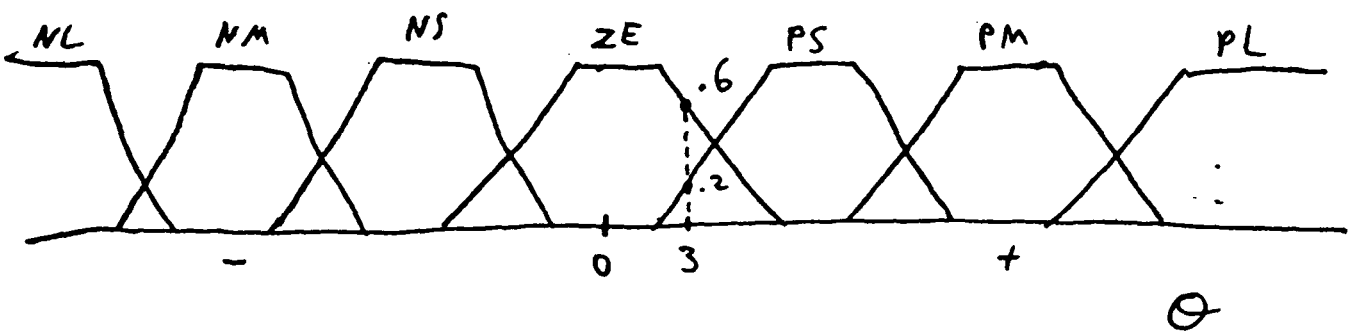


FIGURE 1.7 Seven trapezoidal fuzzy-set values assumed by fuzzy variable θ. Each value of θ belongs to each fuzzy set to some, but usually zero, degree. The exact value 3 belongs to the zero fuzzy number ZE to degree 0.6, to the positive small fuzzy number PS to degree .2, and to positive medium PM to

degree 0.

Fuzzy systems allow users to articulate linguistic FAM rules by entering values in a FAM matrix. Once a fuzzy engineer defines variables and fuzzy sets, the engineer can design a prototype fuzzy system in minutes.

Chapter 17 shows that a large neural-type matrix encodes each FAM rule. When fuzzy variables assume fuzzy subsets of the real line, as when we define ZE as a triangle centered about 0, then these associative matrices have uncountably infinite dimension. This endows each FAM rule with rich structure and "memory capacity." FAM systems do not add these matrices together, which avoids neural-type crosstalk.

A virtual representation scheme allows us to exploit the coding and capacity properties of these infinite matrices without actually writing them down. This holds for binary input-output FAMs (BIOFAMs), which includes all fuzzy systems used in commercial applications. BIOFAMs accept nonfuzzy scalar inputs, such as $\theta = 15$ and $\Delta\dot{\theta} = -10$, and generate nonfuzzy scalar outputs, such as $v = -3$.

Generating Fuzzy Rules With Product-Space Clustering

Neural networks can adaptively generate the FAM rules in a fuzzy system. We illustrate this in Chapters 17 - 20 with the new technique of unsupervised *product-space clustering.* Synaptic vectors quantize the input-output space. Clustered synaptic vectors track how experts associate appropriate responses with input stimuli. Each synaptic cluster estimates a FAM rule. The experts who generate the input-output data need not articulate the FAM rules. They need only behave as experts. The key geometric idea is *cluster equals rule.*

Consider the input-output product space of the inverted-pendulum system. There are two input variables and one output variable, so the input-output product space equals R^3 (in practice a three-dimensional sub-cube within R^3). Each input-output triple $(\theta, \Delta\theta, v)$ defines a point in R^3. The time evolution of the inverted-pendulum system defines a smooth curve or trajectory in R^3. As the fuzzy system stabilizes the inverted pendulum to its vertical position, the trajectory may spiral into the origin of R^3, where the above

steady-state FAM rule keeps the system in equilibrium until perturbed.

Each fuzzy variable can assume five fuzzy subsets of the x, y, or z coordinate axes of R^3. The Cartesian product of these fuzzy subsets defines 125 ($5 \times 5 \times 5$) *FAM cells* in the input-output product space R^3. Most system trajectories pass through only a few FAM cells. We show in Chapter 17 that these FAM cells equal FAM rules because the FAM cells equal fuzzy cartesian products, and the uncountably infinite entries in the associative matrices correspond to these cartesian products. So FAM rule equals associative (fuzzy Hebb) matrix, which equals fuzzy cartesian product, which equals FAM cell.

Unsupervised neural clustering algorithms efficiently track the density of input-output samples in FAM cells. We need only count the number of synaptic vectors in each FAM cell at any instant to estimate, and to weight, the underlying FAM rules used by the expert or physical process that generates the input-output data. This produces an *adaptive histogram* of FAM-cell occupation. Chapters 17 - 20 apply the adaptive product-space clustering methodology to inverted-pendulum control, backing up a truck-and-trailer in a parking lot, and realtime target tracking.

Suppose a system contains n fuzzy variables, and each fuzzy variable can assume m fuzzy-set values. This defines m^n FAM cells in the input-output product space R^n. Different fuzzy variables can assume different types and different numbers of fuzzy-set variables. So in general there are $m_1 \times \ldots \times m_n$ FAM cells. Suppose $n = m = 3$. Suppose the fuzzy sets are low, medium, and high and have bounded extent. Then a Rubik's cube represents the input-output product space partitioned into 27 FAM cells if the fuzzy sets do not overlap. In general FAM cells have nonempty but fuzzy intersection.

If we define n fuzzy variables, each with m fuzzy-set values, then there are 2^{m^n} possible fuzzy systems. Expert articulation, fuzzy engineering, and adaptive estimation produce only a small fraction of the total number 2^{m^n} of possible fuzzy systems. Different fuzzy-set definitions and different encoding or decoding strategies ("inferencing" techniques) produce different classes of 2^{m^n} possible fuzzy systems.

Fuzzy Systems as Parallel Associators

Fuzzy systems store and process FAM rules in parallel. Mathematically a fuzzy system maps points in an input product hypercube (possibly of infinite dimension) to points in an output hypercube. Fuzzy systems associate output fuzzy sets with input fuzzy sets, and so behave as associative memories. Unlike neural associative memories, fuzzy systems do not sum the associative matrices that represent FAM rules. *Neural networks sum throughputs. Fuzzy systems sum outputs.*

Summing outputs avoids crosstalk and achieves modularity. We can meaningfully look inside the black box of fuzzy model-free estimator. Figure 1.8 displays the generic fuzzy system architecture for a single-input, single-output FAM system.

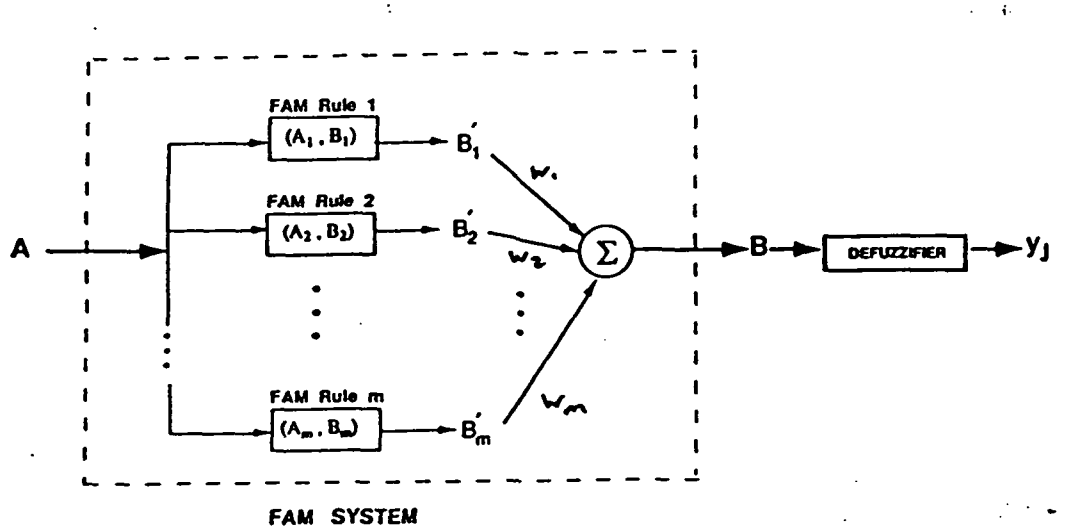


FIGURE 1.8 Fuzzy system architecture. The system maps input fuzzy sets A to output fuzzy sets B. The system stores separate FAM rules and in parallel fires each FAM rule to some degree for each input. Experts or adaptive

algorithms determine the FAM-rule weights w_j. Experts may use only $w_j = 1$ (articulates rule) or $w_j = 0$ (omits rule). Centroidal output converts fuzzy-set vector B to a scalar. In BIOFAM systems A defines a unit binary vector or delta pulse.

Fuzzy inference computes the output fuzzy sets B'_j, weights them with the scalar weights w_j, and sums them to produce the output fuzzy set B:

$$B = \sum_j w_j B'_j \quad . \tag{31}$$

In principle in (31) we sum over all m^n possible FAM rules since most rules have weight $w_i = 0$. Chapter 17 discusses the mechanism of the two types of fuzzy inference, correlation-product and correlation-minimum inference.

Adaptive fuzzy systems use sample data and neural or statistical algorithms to choose the coefficients w_j and thus to define the fuzzy system at each time instant. Adaptation changes the system structure. Geometrically, a time-varying between-cube mapping defines an adaptive fuzzy system. In the simplest case, if the input fuzzy sets define points in the unit hypercube I^n, and the output fuzzy sets define points in the unit hypercube I^p, then transformation S defines a fuzzy system if S maps I^n to I^p, $S : I^n \longrightarrow I^p$. Then S associates fuzzy subsets of the output space Y with fuzzy subsets of the input space X. So $S(A) = B$. S defines an adaptive fuzzy system if S changes with time:

$$\frac{dS}{dt} \neq 0 \quad . \tag{32}$$

BIOFAM systems convert the vector B into a single scalar output value $y \in Y$. We call this process defuzzification, although to defuzzify a fuzzy set formally means to round it off from some point in a unit hypercube to the nearest bit-vector vertex. Fuzzy engineers sometimes compute y as the mode y_{max} of the B distribution,

$$m_B(y_{max}) = sup \{m_B(y) : y \in Y\} \quad . \tag{33}$$

m_B denotes the fuzzy membership function $m_B : Y \longrightarrow [0, 1]$ that assigns fit values or

occurrence degrees to the elements of Y. If the output space Y equals a finite set of values $\{y_1, \ldots, y_p\}$, as in some computer discretizations, then we can replace the supremum in (33) with a maximum:

$$m_B(y_{\max}) = \max_j m_B(y_j) \quad , \tag{34}$$

The more popular centroidal defuzzification technique uses all, and only, the information in the fuzzy distribution B to compute y as the centroid $\bar{y}$ or center of mass of B:

$$\bar{y} = \frac{\int_{-\infty}^{\infty} y\, m_B(y)\, dy}{\int_{-\infty}^{\infty} m_B(y)\, dy} \quad , \tag{35}$$

provided the integrals exist. In practice we restrict fuzzy subsets to finite stretches of the real line. In Chapter 19 we prove that if the fuzzy variables assume only symmetric trapezoid-like fuzzy-set values, then (35) reduces to a simple discrete ratio. The numerator and denominator contain only m products. This discrete centroid trivializes the computational burden of defuzzification and admits direct VLSI implementation.

Figure 1.8 and equation (31) additively combine the weighted fuzzy sets B'_j. Earlier fuzzy systems [Mamdani, 1977] combined output fuzzy sets with pairwise maxima. Unfortunately, the maximum combination technique,

$$B = \max_j \min(w_j, B'_j) \quad , \tag{36}$$

based upon the so-called "extension principle" of classical fuzzy theory [Klir, 1988], tends to produce a uniform distribution for B as the number of combined fuzzy sets increases [Kosko, 1987]. A uniform distribution always has the same mode and centroid. So, ironically, as the number of FAM rules increases, system sensitivity decreases.

The additive combination technique (31) tends to invoke the fuzzy version of the Central Limit Theorem. The added fuzzy waveforms pile up to approximate a symmetric unimodal, or bell-shaped, membership function. Different fuzzy waveforms produce simi-

larly *shaped* output distributions B but centered about different places on the real line. We consistently observe this tendency towards a Gaussian membership function after summing only a few fuzzy waveforms. (Technically the CLT requires normalization by the square-root of the number of summed waveforms. Equation (31) does not normalize B because, for defuzzification, we care only about the relative values in B, the relative degrees of occurrence of output values.)

The maximum combination technique (36) forms the envelope of the weighted fuzzy sets B'_j. Then B resembles the silhouette of a desert-full of sand dunes. As the number of sand dunes increases, the silhouette becomes flatter. The additive combination technique (31) piles the sand dunes atop one other to form a sand mountain.

Fuzzy inference allows us to reason with sets as if they were propositions. The virtual-representation scheme for FAM rules greatly simplifies the fuzzy inference process if we use exact numerical inputs. Figure 1.9 illustrates the FAM (correlation-minimum) inference procedure derived in Chapter 17. We can apply this inference procedure in parallel to any number of FAM rules with any number of antecedent fuzzy-variable conditions.

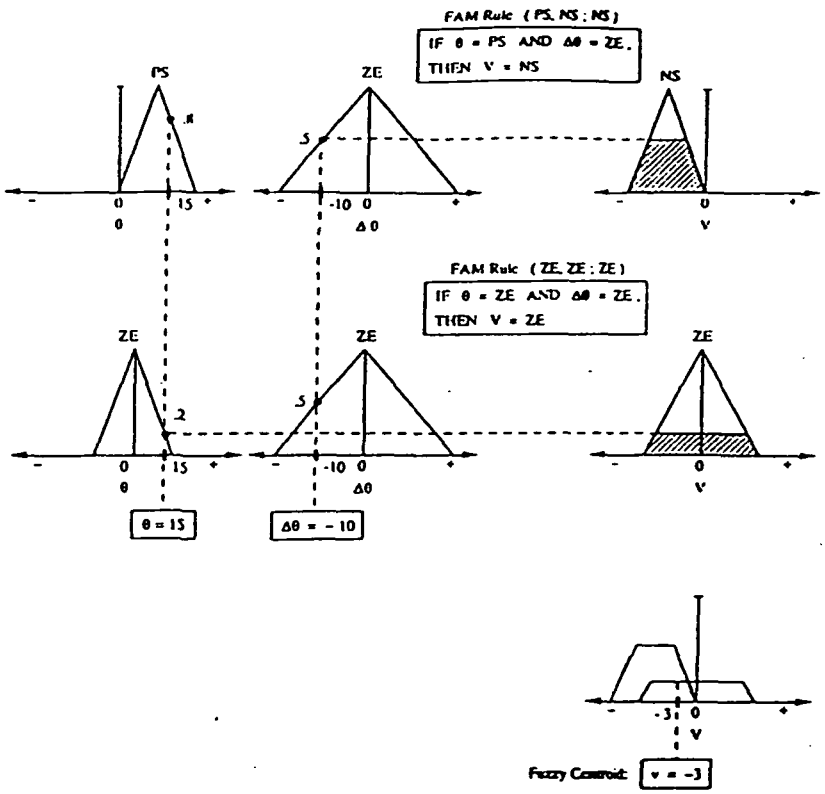


FIGURE 1.9 FAM inference procedure. The fuzzy system converts the numerical inputs, $\theta = 15$ and $\Delta\theta = -10$, into the numerical output $v = -3$. Since the FAM rules combine the antecedent terms with AND, the smaller of the two fit values scales the output fuzzy set. If the FAM rules combined antecedents disjunctively with OR, the larger of the fit values would scale the output fuzzy set.

Fuzzy Systems as Principle-Based Systems

AI expert systems chain through rules. Inference proceeds down, or up, branches of a decision tree. Except for chess trees or other game trees, in practice these search trees

are wider than they are deep. Shallow trees (or forests) can exaggerate the all-or-none effect of bivalent propositional rules. Relative to deeper trees, shallow trees use a smaller proportion of their stored knowledge when they inference. They are noninteractive.

Fuzzy systems are shallow but fully interactive. Every inference fires every FAM rule, itself a fuzzy expert system, to some degree. A similar property holds for the feedback fuzzy cognitive maps discussed in Chapter 4.

Consider an AI judge and a fuzzy judge. Opposing counsel present the same evidence and testimony to both judges. The AI judge rounds off the truth value of every key statement or alleged fact to TRUE or FALSE (1 or 0), opens a rule book, uses the true statements to activate or choose the antecedents of some of the rules, then logically chains through the rule tree to reach a decision. A more sophisticated AI judge may chain through the rule tree with uncertainty-factor algorithms or heuristic search algorithms.

The fuzzy judge weights the evidence to different degrees, say with fractional values in the unit interval [0, 1]. The fuzzy judge does not use a rule book. Instead the fuzzy judge determines to what degree the fuzzy evidence invokes a large set of vague legal *principles*. The fuzzy judge may cite case precedents to enunciate these principles or to illustrate their relative importance. The fuzzy judge reaches a decision by combining these fuzzy facts and fuzzy principles in an unseen act of intuition or judgement. If pressed, the fuzzy judge may defend or explain the decision by citing the salient facts and relevant legal principles, precedents, and perhaps rules. In general the fuzzy judge cannot articulate an exact legal audit trail of the decision process.

The distinction between the AI judge and the fuzzy judge reduces to the distinction between rules and principles. Recently legal theorists [Dworkin, 1968-77; Hayek, 1973] have focused on this distinction and challenged the earlier "positivist" legal theories of law as articulated rules [Kelsen, 1954; Hart, 1961].

Rules, as Dworkin [1977] says, apply "in an all-or-none fashion." Principles "have a dimension that rules do not—the dimension of weight or importance," and the court "cites principles as its justification for adopting and applying a new rule." Rules greatly outnumber principles. Principles guide while rules specify:

"Only rules dictate results, come what may. When a contrary result has been reached, the rule has been abandoned or changed. Principles do not work that way; they incline a decision one way, though not conclusively, and they survive intact when they do not prevail."

Rules tend to be black or white. They abruptly come into and out of existence. We post rules on signs, vote on them as propositions, and send them in memos: must be 18 to vote, open from 8 am to 5 pm, $500 fine for littering, office term lasts four years, can take only five sick days a year, and so on. Rules come and go as culture evolves.

Principles evolve as culture evolves. Most legal principles in the United States grew out of medeval British common law. Each year their character changes slightly, adaptively, as we apply them to novel circumstances. These principles range from very abstract principles, such as presumption of innocence or freedom of contract, to more behavioral principles, such as that no one can profit from a crime or you cannot challenge a contract if you acquiese to it and act on it.

Each principle admits a spectrum of exceptions. In each case a principle holds only to some, often slight, degree. Judges cite case precedents in effect to estimate the current weight of principles. All the principles "hang together" to some degree in each decision, just as all the fuzzy rules (principles) in Figure 1.5 contribute to some degree to the final inference or decision.

We often call AI expert systems rule-based systems because they consist of a bank or forest of propositional rules and an "inference engine" for chaining through the rules. The rule in rule-based emphasizes the articulated, expertly precise nature of the stored knowledge.

The AI precedent and modern legal theory suggest that we should call fuzzy systems **principle-based systems**. The fuzzy rules or principles indicate how entire clumps of output spaces associate with clumps of input spaces. Indeed FAM rules often behave as partial derivatives. Many applications require only a few FAM rules for smooth system control or estimation. In general AI rule-based systems would require vastly more precise rules to approximate the same system performance.

Adaptive fuzzy systems use neural (or statistical) techniques to abstract fuzzy principles from sampled cases and to gradually refine those principles as the system samples new cases. The process resembles our everyday acquisition and refinement of commonsense knowledge. Future machine-intelligent systems may match, then someday exceed, our ability to learn and apply the fuzzy commonsense knowledge—knowledge we can articulate only rarely and inexactly—that we use to run our lives and run our world.

REFERENCES

Birkhoff, G., von Neumann, J., "The Logic of Quantum Mechanics," *Annals of Mathematics*, vol. 37, no. 4, 823 - 843, October 1936.

Black, M., "Vagueness: An Exercise in Logical Analysis," *Philosophy of Science*, vol. 4, 427 - 455, 1937.

Churchland, P.M., "Eliminative Materialism and the Propositional Attitudes," *Journal of Philosophy*, vol. 78, no. 2, 67 - 90, February 1981.

Crick, F., "Function of the Thalamic Reticular Complex: The Searchlight Hypothesis," *Proceedings of the National Academy of Sciences*, vol. 81, 4586 - 4590, 1984.

Dworkin, R.M., "Is Law a System of Rules?" in *Essays in Legal Philosophy*, R.S. Summers, ed., Oxford University Press, 1968.

Dworkin, R.M., *Taking Rights Seriously*, Harvard University Press, 1977.

Grossberg, S., "Cortical Dynamics of Three-Dimensional Form, Color, and Brightness Perception: I. Monocular Theory," *Perception and Pschophysics*, vol. 41, no. 2, 87 - 116, 1987.

Gould, S.J., "Is a New and General Theory of Evolution Emerging?" *Paleobiology*, vol. 6, no. 1, 119 - 130, 1980.

Hart, H.L.A., *The Concept of Law*, Oxford University Press, 1961.

Hayek, F.A.,*Law, Legislation, and Liberty, volume I: Rules and Order*, University of Chicago Press, 1973.

Hopfield, J.J., "Neurons with Graded Response have Collective Computational Properties like Those of Two-State Neurons," *Proceedings of the National Academy of Sciences*, vol. 81, 3088 - 3092, 1984.

Kanizsa, G., "Subjective Countours," *Scientific American*, vol. 234, 48 - 52, 1976.

Kant, I., *Prolegemona to any Future Metaphysics*, 1783.

Kant, I., *Critique of Pure Reason*, second edition, 1787.

Kelsen, H., *General Theory of Law and State*, Harvard University Press, 1954.

Kline, M., *Mathematics: The Loss of Certainty*, Oxford University Press, 1980.

Klir, G.J., Folger, T.A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, 1988.

Kosko, B., "Fuzzy Entropy and Conditioning," *Information Sciences*, vol. 40, 165 - 174, 1986.

Kosko, B., *Foundations of Fuzzy Estimation Theory*, Ph.D. dissertation, Department of Electrical Engineering, University of California at Irvine, June 1987; Order Number 8801936, University Microfilms International, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

Kosko, B., "Fuzziness vs. Probability," *International Journal of General Systems*, vol. 17, no. 2, 211 - 240, 1990.

Mamdani, E.H., "Application of Fuzzy Logic to Approximate Reasoning Using Linguistic Synthesis," *IEEE Transactions on Computers*, vol. C-26, no. 12, 1182 - 1191, December 1977.

Mead, C., *Analog VLSI and Neural Systems*, Addison-Wesley, 1989.

Pearl, J., *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley, 1984.

Quine, W.V.O., "What Price Bivalence?" *Journal of Philosophy*, vol. 78, no. 2, 90 - 95, February 1981.

Rescher, N., *Many-valued Logic*, McGraw Hill, New York, 1969.

Rosser, J.B., Turquette, A.R., *Many-Valued Logics*, North-Holland, 1952.

Rudin, W., *Real and Complex Analysis*, McGraw Hill, 1974.

Skinner, B.F., *Science and Human Behavior*, Macmillan, 1953.

Thompson, R.F., *The Brain: An Introduction to Neuroscience*, New York: W.H. Freeman and Company, 1985.

Wiener, N., *Cybernetics: Control and Communication in the Animal and the Machine*, MIT Press, 1948.

Wilson, E.O., *Sociobiology: The New Synthesis*, Harvard University Press, 1975.

Yao, Y., Freeman, W.J., "Model of Biological Pattern Recognition with Spatially Chaotic Dynamics," *Neural Networks*, 153 - 170, vol. 3, no. 2, 1990.

Zadeh, L.A., "Fuzzy Sets," *Information and Control*, vol. 8, 338 - 353, 1965.

PROBLEMS

1. Lukasiewicz's continuous or "fuzzy" logic (L_1 logic) uses a continuous-valued truth function $t : S \longrightarrow [0, 1]$ defined on the set S of statements. Lukasiewicz defined the generalized conjunction (AND), disjunction (OR), negation (NOT) operators respectively as

$$t(A \text{ AND } B) = \min(t(A), t(B)) ,$$

$$t(A \text{ OR } B) = \max(t(A), t(B)) ,$$

$$t(\text{NOT-}A) = 1 - t(A) ,$$

for statements A and B. Prove the generalized noncontradiction-excluded-middle law:

$$t(A \text{ AND } \sim A) + t(A \text{ OR } \sim A) = 1 .$$

This equality implies that the classical bivalent law of noncontradiction, $t(A \text{ AND } \sim A) = 0$, holds if and only if the classical bivalent law of excluded middle, $t(A \text{ OR } \sim A) = 1$, holds. Note that in the case of bivalent "paradox," when $t(A) = t(\text{NOT-}A)$, the equality reduces to the equality $1/2 + 1/2 = 1$.

2. Let $t : S \longrightarrow [0, 1]$ be a continuous or "fuzzy" truth function on the set S of statements. Define the Lukasiewicz implication operator as the truth function $t_L(A \longrightarrow B) = \min(1, 1 - t(A) + t(B))$ for statements A and B. Then prove the following generalized fuzzy *modus ponens* inference rule:

$$t_L(A \longrightarrow B) = c$$
$$t(A) \geq a$$

Therefore $t(B) \geq \max(0,\ a + c - 1)$.

Hence if $t(A) = t_L(A \longrightarrow B) = 1$, then $t(B) = 1$, which generalizes classical bivalent *modus ponens*.

3. Use the continuous logic operations in Problem 2 to prove the following generalized fuzzy *modus tollens* inference rule:

$$t_L(A \longrightarrow B) = c$$
$$t(B) \leq b$$

Therefore $t(A) \leq \min(1,\ 1 - c + b)$.

Hence if $t_L(A \longrightarrow B) = 1$ and $t(B) = 0$, then $t(A) = 0$, which generalizes classical bivalent *modus tollens*.

4. Define the Gaines implication operator as

$$t_G(A \longrightarrow B) = \begin{cases} \min(1,\ t(B)/t(A)) & \text{if} \quad t(A) > 0 \\ 1 & \text{if} \quad t(A) = 0 \end{cases} .$$

Use the Gaines implication operator $t_G(A \longrightarrow B)$ to derive generalized fuzzy *modus ponens* and *modus tollens* inference rules. The conclusion of the inference rules should differ from the conclusions of the inference rules in Problems 2 and 3.

5. Exclusive-or (XOR) equals negated logically equivalence:

$$t(A \quad XOR \quad B) \quad = \quad 1 \; - \; t(A = B) \quad .$$

Equivalence equals biconditionality. Bivalent statements are equivalent if and only if the two statements have the same truth values. So the exclusive-or relation holds between two bivalent statements if and only if the two statements have opposite truth values.

Fuzzy statements can be equivalent to different degrees. But equivalence still equals biconditionality:

$$t(A = B) \quad = \quad t((A \longrightarrow B) \text{ AND } (B \longrightarrow A)) \quad .$$

Prove that if we use the Lukasiewicz implication operator, then exclusive-or equals the absolute difference (or l^1 or fuzzy Hamming distance) of the truth values $t(A)$ and $t(B)$:

$$t_L(A \quad XOR \quad B) \quad = \quad |t(A) \; - \; t(B)| \quad .$$

6. Set X contains n elements $x_1, \ldots, x_n$. So X contains 2^n nonfuzzy subsets A. Define the bivalent indicator function I_A of nonfuzzy set A as

$$I_A(x_i) \quad = \quad \begin{cases} 1 \text{ if } x_i \in A \\ 0 \text{ if } x_i \notin A \end{cases}$$

So I_A defines the mapping $I_A : X \longrightarrow \{0, 1\}$.

Suppose we extend I_A to a multivalued mapping by augmenting its range from $\{0, 1\}$ to $\{y_1, \ldots, y_m\}$, where $y_1 = 0$, $y_m = 1$, and $0 < y_j < 1$ if $1 < j < m$. Then I_A defines the mapping $I_A : X \longrightarrow \{y_1, \ldots, y_m\}$. How many multivalued

subsets does X have? In the 2-dimensional case, $X = \{x_1, x_2\}$, draw the planar lattice that describes the multi-dimensional power set of X, all its multi-dimensional subsets, when $m = 3$, and when $m = 5$.

7. Consider the discrete dynamical system

$$\begin{aligned} x_{k+1} &= f(x_k) \\ &= c\, x_k\, (1 - x_k) \quad , \end{aligned}$$

for x values in $[0, 1]$ and $0 < c \leq 4$. Many dynamical systems transition into chaos as we increase a control or gain parameter, such as c. Select $c = 3.5$ and use the two choices of initial conditions, $x_0 = .5$ and $x_0 = .51$, to generate $x_1, \ldots, x_{20}$. Plot the two trajectories on graph paper. Are they aperiodic (chaotic) or periodic? Does a difference of .01 in initial condition significantly affect the overall shape of the discrete trajectory?

Now repeat the above experiment but use the gain parameter $c = 3.9$ (or $c = 4$). No matter how close two initial conditions, in a chaotic dynamical system they always produce divergent trajectories. Does $c = 3.9$ produce chaos?

CHAPTER 16

FUZZINESS VERSUS PROBABILITY

So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.

. . . Albert Einstein

Fuzzy Sets and Systems

We now explore fuzziness as an alternative to randomness for describing uncertainty. We develop the new *sets-as-points* geometric view of fuzzy sets. This view identifies a fuzzy set with a point in a unit hypercube, a nonfuzzy set with a vertex of the cube, and a fuzzy system as a mapping between hypercubes. Chapter 17 examines fuzzy systems.

Paradoxes of two-valued logic and set theory, such as Russell's paradox, correspond to the midpoint of the fuzzy cube. We geometrically answer the fundamental questions of fuzzy theory—How fuzzy is a fuzzy set? How much is one fuzzy set a subset of another?—with the Fuzzy Entropy Theorem and the Fuzzy Subsethood Theorem.

We develop a new geometric proof of the Subsethood Theorem. A corollary shows that the apparently probabilistic relative frequency $\frac{n_A}{N}$ equals the deterministic subsethood $S(X, A)$, the degree to which the sample space X is contained in its subset A. So the

frequency of successful trials equals the degree to which all trials are successful. We examine recent Bayesian polemics against fuzzy theory in light of the new sets-as-points theorems.

An element belongs to a fuzzy set to some degree in [0, 1]. An element belongs to a nonfuzzy set all or none, 1 or 0. More fundamentally, one set is a subset of one of the set to some degree. Sets fuzzily contain subsets as well as elements. Subsethood generalizes elementhood. We shall argue that subsethood generalizes probability as well.

Fuzziness in a Probabilistic World

Is uncertainty the same as randomness? If we are not sure about something, is it only up to chance? Do the notions of likelihood and probability exhaust our notions of uncertainty?

Many people, trained in probability and statistics, believe so. Some even say so, and say so loudly. These voices often arise from the Bayesian camp of statistics, where probabilists view probability not as a frequency or other objective testable quantity, but as a subjective *state of knowledge.*

Bayesian physicist E. T. Jaynes [1979] says that "any method of inference in which we represent degrees of plausibility by real numbers, is necessarily either equivalent to Laplace's [probability], or inconsistent." He claims physicist R. T. Cox [1946] has proven this as a theorem, a claim we examine below.

More recently, Bayesian statistician Dennis Lindley [1987] issued an explicit challenge: "probability is the only sensible description of uncertainty and is adequate for all problems involving uncertainty. All other methods are inadequate."

Lindley directs his challenge in large part at *fuzzy theory*, the theory that *all things admit degrees*, but admit them deterministically. We accept the probabilist's challenge from the fuzzy viewpoint. We will defend fuzziness with new geometric first principles and will question the reasonableness and the axiomatic status of randomness. The new view is the *sets-as-points view* [Kosko, 1987] of fuzzy sets: A fuzzy set defines a point in a unit-hypercube, and a nonfuzzy set defines a corner of the hypercube.

Randomness and fuzziness differ conceptually and theoretically. We can illustrate some differences with examples. Others we can prove with theorems, as we show below.

Randomness and fuzziness also share many similarities. Both systems describe uncertainty with numbers in the unit interval [0, 1]. This ultimately means that both systems describe uncertainty numerically. Both systems combine sets and propositions associatively, commutatively, and distributively. The key distinction concerns how the systems jointly treat a thing A and its opposite A^c. Classical set theory demands $A \cap A^c = \emptyset$, and probability theory conforms: $P(A \cap A^c) = P(\emptyset) = 0$. So $A \cap A^c$ represents a probabilistically impossible event. But fuzziness begins when $A \cap A^c \neq \emptyset$.

Questions raise doubt, and doubt suggests room for change. So to commence the exposition, consider the following two questions, one fuzzy and the other probabilistic:

(i) Is it always and everywhere true that $A \cap A^c = \emptyset$?

(ii) Do we *derive* or *assume* the conditional probability operator

$$P(B|A) = \frac{P(A \cap B)}{P(A)} \text{ ?} \tag{1}$$

The second question may appear less fundamental than the first question, which asks whether fuzziness exists. The Entropy-Subsethood Theorem below shows that the first question reduces to the second questions: We measure the fuzziness of fuzzy set A when we measure how much the superset $A \cup A^c$ is a subset of its own subset $A \cap A^c$, a paradoxical relationship unique to fuzzy theory. In contrast, in probability theory the like relationship is impossible (has zero probability): $P(A \cap A^c | A \cup A^c) = P(\emptyset | X) = 0$, where X denotes the sample space or "sure event".

The conditioning or subsethood in the second question lies at the heart of Bayesian probabilistic systems. We may accept the absence of a first-principles derivation of $P(B|A)$. We can simply agree to take the ratio relationship as an axiom. But the new sets-as-points view of fuzzy sets *derives* its conditioning operator as a theorem from first principles. The history of science suggests that systems that hold theorems as axioms continue to evolve.

The first question asks whether we can logically or factually violate the law of noncontra-

diction—one of Aristotle's three "laws of thought" along with the laws of excluded middle, $A \cup A^c = X$, and identity, $A = A$. Set fuzziness occurs when, and only when, it is violated. Classical logic and set theory assume that we cannot violate the law of noncontradiction or, equivalently, the law of excluded middle. This makes the classical theory black or white. Fuzziness begins where Western logic ends—where contradictions begin.

Randomness vs. Ambiguity: Whether vs. How Much

Fuzziness describes *event ambiguity*. It measures the degree to which an event occurs, not whether it occurs. Randomness describes the uncertainty of *event occurrence*. An event occurs or not, and you can bet on it. The issue concerns the occurring event: Is it uncertain in any way? Can we unambiguously distinguish the event from its opposite?

Whether an event occurs is "random". To what degree it occurs is fuzzy. Whether an ambiguous event occurs—as when we say there is 20% chance of light rain tomorrow—involves compound uncertainties, the probability of a fuzzy event.

We regularly apply probabilities to fuzzy events: small errors, satisfied customers, A students, safe investments, developing countries, noisy signals, spiking neurons, dying cells, charged particles, nimbus clouds, planetary atmospheres, galactic clusters. We understand that, at least around the edges, some satisfied customers can be somewhat unsatisfied, some A students might equally be B+ students, some stars are as much in a galactic cluster as out of it. Events can transition more or less smoothly to their opposites, making classification hard near the midpoint of the transition. But in theory—in formal descriptions and in textbooks—the events and their opposites are black and white. A hill is a mountain if it is at least x meters tall, not a mountain if it is one micron less than x in height [Quine, 1981]. Every molecule in the universe either is or is not a pencil molecule, even those that hover about the pencil's surface.

Consider some further examples. The probability that this chapter gets published is one thing. The degree to which it gets published is another. The chapter may be edited

in hundreds of ways. Or the essay may be marred with typographical errors, and so on.

Question: Does quantum mechanics deal with the probability that an unambiguous electron occupies spacetime points? Or does it deal with the degree to which an electron, or an electron smear, occurs at spacetime points? Does $|\psi|^2\, dV$ measure the probability that a random-point electron occurs in infinitesimal volume dV? Or [Kosko, 1990] does it measure the degree to which a deterministic electron cloud occurs in dV? Different interpretation, different universe. Perhaps even existence admits degrees at the quantum level.

Suppose there is 50% chance that there is an apple in the refrigerator (electron in a cell). That is one state of affairs, perhaps arrived at through frequency calculations or a Bayesian state of knowledge. Now suppose there is half an apple in the refrigerator. That is another state of affairs. Both states of affairs are superficially equivalent in terms of their numerical uncertainty. Yet physically, ontologically, they differ. One is "random", the other fuzzy.

Consider parking your car in a parking lot with painted parking spaces. You can park in any space with some probability. Your car will totally occupy one space and totally unoccupy all other spaces. The probability number reflects a frequency history or Bayesian brain state that summarizes which parking space your car will totally occupy. Alternatively, you can park in every space to some degree. Your car will partially, and deterministically, occupy every space. In practice your car will occupy most spaces to zero degree. Finally, we can use numbers in $[0, 1]$ to describe, for each parking space, the occurence probability of each degree of partial occupancy—probabilities of fuzzy events.

If we *assume* events are unambiguous, as in balls-in-urns experiments, there is no set fuzziness. Only "randomness" remains. But when we discuss the physical universe, every assertion of event ambiguity or nonambiguity is an empirical *hypothesis*. We habitually overlook this when we apply probability theory. Years of such oversight have entrenched the sentiment that uncertainty is randomness, and randomness alone. We systematically assume away event ambiguity. We call the partially empty glass empty and call the small number zero. This silent assumption of universal nonambiguity resembles the pre-relativistic assumption of an uncurved universe. $A \cap A^c = \emptyset$ is the "parallel postulate"

of classical set theory and logic, indeed of Western thought.

If fuzziness is a genuine type of uncertainty, if fuzziness exists, the physical consequences are universal, and the sociological consequence is startling: scientists, especially physicists, have overlooked an entire mode of reality.

Fuzziness is a type of deterministic uncertainty. Ambiguity is a property of physical phenomena. Unlike fuzziness, probability dissipates with increasing information. After the fact "randomness" looks like fiction. Yet many of the laws of science are time reversible, invariant if we replace time t with time $-t$. If we run the universe in reverse as if it were a video tape, where does the "randomness" go? There is as much ambiguity after a sample-space experiment as before. Increasing information specifies the degrees of occurrence. Even if science had run its course and all the facts were in, a platypus would remain only roughly a mammal, a large hill only roughly a mountain, an oval squiggle only roughly an ellipse. Fuzziness does not require that God plays dice.

Consider the inexact oval in Figure 16.1. Does it make more sense to say that the oval is *probably* an ellipse, or that it *is* a fuzzy ellipse? There seems nothing random about the matter. The situation is deterministic: All the facts are in. Yet uncertainty remains. The uncertainty arises from the simultaneous occurrence of two properties: to some extent the

inexact oval is an ellipse, and to some extent it is not an ellipse.

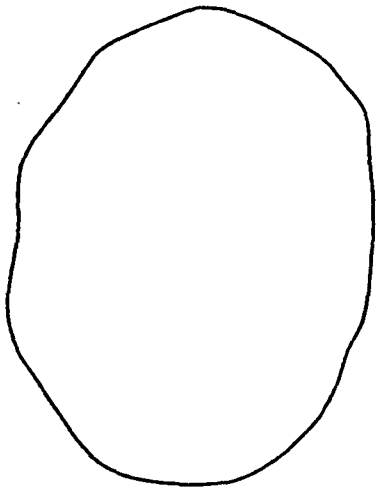


Figure 16.1 Inexact oval. Which statement better describes the situation: "It is probably an ellipse" or "It is a fuzzy ellipse"?

More formally, does $m_A(x)$, the degree to which element x belongs to fuzzy set A, equal the probability that x belongs to A? Is $m_A(x) = \text{Prob}\{x \in A\}$ true? Cardinality-wise, sample spaces cannot be too big. Else a positive measure cannot be both countably additive and finite, and thus in general cannot be a probability measure [Chung, 1974]. The space of all possible oval figures is too big, since there are more of these than real numbers. Almost all sets are too big for us to define probability measures on them, yet we can always define fuzzy sets on them.

Probability theory is a chapter in the book of finite measure theory. Many probabilists do not care for this classification, but they fall back upon it when defining terms [Kac, 1959]. How reasonable is it to believe that finite measure theory—ultimately, the summing of nonnegative numbers to unity—exhaustively describes the universe? Does it really describe *any* thing?

Surely from time to time every probabilist wonders whether probability describes anything real. From Democritus to Einstein, there has been the suspicion that, as David Hume [1748] put it, "though there be no such thing as *chance* in the world, our ignorance of the real cause of any event has the same influence on the understanding and begets a like species of belief." When we model noisy processes by extending differential equations to stochastic differential equations, as in Chapters 4-6, we introduce the formalism only as a working approximation to several underlying unspecified processes, processes that presumably obey deterministic differential equations. In this sense conditional expectations and martingale techniques might seem reasonably applied, for example, to stock options or commodity futures phenomena, where the behavior involved consists of aggregates of aggregates of aggregates. The same techniques seem less reasonably applied to quarks, leptons, and void.

The Universe as a Fuzzy Set

The world, as Wittgenstein [1922] observed, is everything that is the case. In this spirit we can summarize the ontological case for fuzziness: *The universe consists of all subsets of the universe.* The only subsets of the universe that are not in principle fuzzy are the constructs of classical mathematics. The integer 2 belongs to the even integers, and does not belong to the odd or negative integers. All other sets—sets of particles, cells, tissues, people, ideas, galaxies—in principle contain elements to different degrees. Their membership is partial, graded, inexact, ambiguous, or uncertain.

The same universal circumstance holds at the level of logic and truth. The only logically true or false statements—statements S with truth value $t(S)$ in $\{0, 1\}$-are tautologies, theorems, and contradictions. If statement S describes the universe, if S is an *empirical* statement, then $0 < t(S) < 1$ holds by the canons of scientific method and by the lack of a single demonstrated factual statement S with $t(S) = 1$ or $t(S) = 0$. Philosopher Immanuel Kant [1787] wrote volumes in search of factually true logical statements and logically true factual statements.

Logical truth differs in kind from factual truth. "2 = 1 + 1" has truth value 1. "Grass is green" has truth value less than 1 but greater than 0. This produces the math/universe crisis Einstein laments in his quote at the beginning of this chapter. Scientists have imposed a two-valued mathematics, shot through with logical "paradoxes" or antinomies [Kline, 1980], on a multivalued universe. Last century John Stuart Mill [1843] argued that logical truths represent limiting cases of factual truths. This accurately summarized the truth-value distinction between $0 < t(S) < 1$ and $t(S) = 0$ or $t(S) = 1$ but, cast in linguistic form, seems not to have persuaded modern philosophers. The Heisenberg uncertainty principle, with its continuum of indeterminacy, forced multivaluedness on science, though few Western philosophers [Quine, 1981] have accepted multivaluedness. Lukasiewicz, Gödel, and Black [Rescher, 1969] did accept it and developed the first continuous or "fuzzy" logic and set systems.

Fuzziness arises from the ambiguity or vagueness [Black, 1937] between a thing A and its opposite A^c. If we do not know A with certainty, we do not know A^c with certainty either. Else by double negation we would know A with certainty. This ambiguity produces nondegenerate *overlap*: $A \cap A^c \neq \emptyset$, which breaks the "law of noncontradiction." Equivalently, it also produces nondegenerate *underlap* [Kosko,1986b]: $A \cup A^c \neq X$, which breaks the "law of excluded middle." Here X denotes the ground set or universe of discourse. (Probabilistic or stochastic logics [Gaines, 1983] do not break these laws: $P(A \text{ and not-}A) = 0$ and $P(A \text{ or not-}A) = 1$.) Formally, probability measures cannot take fuzzy sets as arguments. We must first quantize, round off, or defuzzify the fuzzy sets to the nearest nonfuzzy sets.

THE GEOMETRY OF FUZZY SETS: SETS AS POINTS

It helps to see the geometry of fuzzy sets when we discuss fuzziness. To date researchers have overlooked this visualization. Instead they have interpreted fuzzy sets as generalized indicator or membership functions [Zadeh, 1965], mappings m_A from domain X to range $[0, 1]$. But functions are hard to visualize. Fuzzy theorists [Klir, 1988] often picture

membership functions as two-dimensional graphs, with the domain X represented as a one-dimensional axis. The geometry of fuzzy sets involves both the domain $X = \{x_1, \ldots, x_n\}$ and the range $[0, 1]$ of mappings $m_A : X \longrightarrow [0, 1]$. The geometry of fuzzy sets aids us when we describe fuzziness, define fuzzy concepts, and prove fuzzy theorems. Visualizing this geometry may by itself provide the most powerful argument for fuzziness.

An odd question reveals the geometry of fuzzy sets: What does the fuzzy power set $F(2^X)$, the set of all fuzzy subsets of X, look like? It looks like a cube. What does a fuzzy set look like? A point in a cube. The set of all fuzzy subsets equals the unit hypercube $I^n = [0, 1]^n$. A fuzzy set is any point [Kosko, 1987] in the cube I^n. So (X, I^n) defines the fundamental measurable space of (finite) fuzzy theory. We can teach much of the theory of fuzzy sets—more accurately, the theory of *continuous* sets—on a Rubik's cube.

Vertices of the cube I^n define nonfuzzy sets. So the ordinary power set 2^X, the set of all 2^n nonfuzzy subsets of X, equals the Boolean n-cube B^n : $2^X = B^n$. Fuzzy sets fill in the lattice B^n to produce the solid cube I^n : $F(2^X) = I^n$.

Consider the set of two elements $X = \{x_1, x_2\}$. The nonfuzzy power set 2^X contains four sets: $2^X = \{\emptyset, X, \{x_1\}, \{x_2\}\}$. These four sets correspond respectively to the four bit vectors (0 0), (1 1), (1 0), and (0 1). The 1s and 0s indicate the presence or absence of the ith element x_i in the subset. More abstractly, we can uniquely define each subset A as one of the two-valued membership functions $m_A : X \longrightarrow \{0, 1\}$.

Now consider the fuzzy subsets of X. We can view the fuzzy subset $A = (\frac{1}{3} \ \frac{3}{4})$ as one of the continuum-many continuous-valued membership functions $m_A : X \longrightarrow [0, 1]$. Indeed this corresponds to the classical Zadeh [1965] *sets-as-functions* definition of fuzzy sets. In this example element x_1 belongs to, or fits in, subset A a little bit—to degree $\frac{1}{3}$. Element x_2 has more membership than not at $\frac{3}{4}$. Analogous to the bit vector representation of finite (countable) sets, we say that the *fit vector* $(\frac{1}{3} \ \frac{3}{4})$ represents A. The element $m_A(x_i)$ equals the ith *fit* [Kosko, 1986b] or *fuzzy unit* value. The sets-as-points view then geometrically represents the fuzzy subset A as a point in I^2, the unit square, as in Figure 16.2.

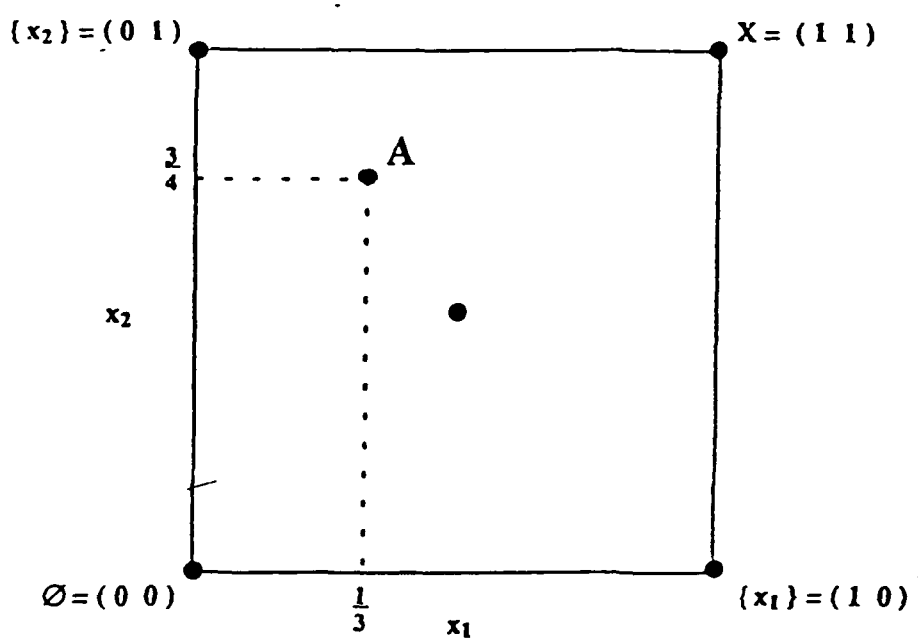


Figure 16.2 Sets as points. The fuzzy subset A is a point in the unit 2-cube with coordinates or fit values $(\frac{1}{3} \ \frac{3}{4})$. The first element x_1 fits in or belongs to A to degree $\frac{1}{3}$, the element x_2 to degree $\frac{3}{4}$. The cube consists of all possible fuzzy subsets of two elements $\{x_1, x_2\}$. The four corners represent the power set 2^X of $\{x_1, x_2\}$.

The midpoint of the cube I^n is maximally fuzzy. All its membership values equal $\frac{1}{2}$. The midpoint is unique in two respects. First, the midpoint is the only set A that not only equals its own opposite A^c but equals its own overlap and underlap as well:

$$A = A \cap A^c = A \cup A^c = A^c \quad . \tag{2}$$

Second, the midpoint is the only point in the cube I^n equidistant to each of the 2^n vertices of the cube. The nearest corners are also the farthest. Figure 16.2 illustrates this

metrical relationship.

We combine fuzzy sets pairwise with minimum, maximum, and order reversal, just as we combine nonfuzzy sets. So we combine set elements with the operators of Lukasiewicz continuous logic [Rescher, 1969]. We define fuzzy set intersection fitwise by pairwise minimum (picking the smaller of the two elements), union by pairwise maximum, and complementation by order reversal:

$$m_{A \cap B} = \min(m_A, m_B) \quad , \tag{3}$$

$$m_{A \cup B} = \max(m_A, m_B) \quad , \tag{4}$$

$$m_{A^c} = 1 - m_A \quad . \tag{5}$$

For example:

$$\begin{aligned} A &= (1 \quad .8 \quad .4 \quad .5) \\ B &= (.9 \quad .4 \quad 0 \quad .7) \\ A \cap B &= (.9 \quad .4 \quad 0 \quad .5) \\ A \cup B &= (1 \quad .8 \quad .4 \quad .7) \\ A^c &= (0 \quad .2 \quad .6 \quad .5) \\ A \cap A^c &= (0 \quad .2 \quad .4 \quad .5) \\ A \cup A^c &= (1 \quad .8 \quad .6 \quad .5) \end{aligned}$$

The overlap fit vector $A \cap A^c$ in this example does not equal the vector of all zeroes, and the underlap fit vector $A \cup A^c$ does not equal the vector of all ones. This holds for all properly fuzzy sets, all points in I^n other than vertex points. Indeed the min-max

definitions give at once the following fundamental characterization of fuzziness as nondegenerate overlap and nonexhaustive underlap.

Proposition. A is properly fuzzy iff $A \cap A^c \neq \emptyset$
iff $A \cup A^c \neq X$.

The proposition says that Aristotle's laws of noncontradiction and excluded middle hold, but they hold only on a set of measure zero. They hold only at the 2^n vertices of I^n. In all other cases, and these are as many of these as there are real numbers, contradictions occur to some degree. In this sense contradictions in generalized set theory and logic represent the rule and not the exception. Fuzzy cubes box Aristotelian sets into corners.

Completing the fuzzy square illustrates this fundamental proposition. Consider again the two-dimensional fuzzy set A defined by the fit vector $(\frac{1}{3} \ \frac{3}{4})$. We find the corresponding overlap and underlap sets by first finding the complement set A^c and then combining the fit vectors pairwise with minimum and with maximum:

$$\begin{aligned} A &= (\tfrac{1}{3} \ \tfrac{3}{4}) \\ A^c &= (\tfrac{2}{3} \ \tfrac{1}{4}) \\ A \cap A^c &= (\tfrac{1}{3} \ \tfrac{1}{4}) \\ A \cup A^c &= (\tfrac{2}{3} \ \tfrac{3}{4}) \quad . \end{aligned}$$

The sets-as-points view shows that these four points in the unit square hang together,

and move together, in a very natural way. Consider the geometry of Figure 16.3.

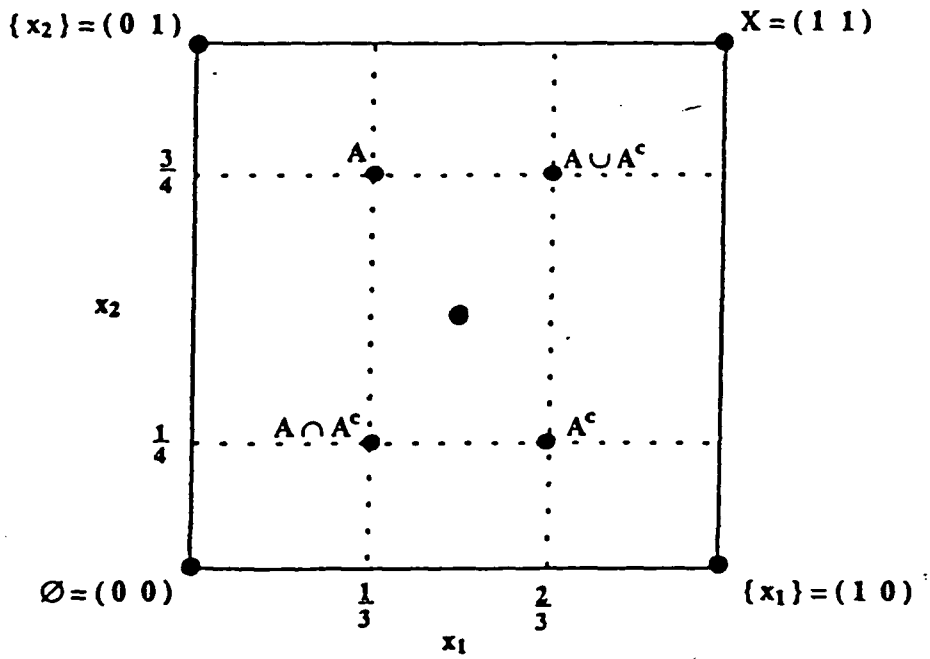


Figure 16.3 Completing the fuzzy square. The fuzzier A is, the closer A is to the midpoint of the fuzzy cube. As A approaches the midpoint, all four points—A, A^c, $A \cap A^c$, and $A \cup A^c$—contract to the midpoint. The less fuzzy A is, the closer A is to the nearest vertex. As A approaches the vertex, all four points spread out to the four vertices and the bivalent power set 2^X is recovered. In an n-dimensional fuzzy cube, the 2^n fuzzy sets with elements a_i or $1 - a_i$ similarly contract to the midpoint or expand to the 2^n vertices as A approaches total fuzziness or total bivalence.

In Figure 16.3 the four fuzzy sets involved in the fuzziness of set A—the sets A, A^c, $A \cap A^c$, and $A \cup A^c$—contract to the midpoint as A becomes maximally fuzzy and expand out to the Boolean corners of the cube as A becomes minimally fuzzy. The same contraction and expansion occurs in n dimensions for the 2^n fuzzy sets defined by all combinations

of $m_A(x_1)$ and $m_{A^c}(x_1), \ldots, m_A(x_n)$ and $m_{A^c}(x_n)$. The same contraction and expansion occurs in n dimensions for the 2^n fuzzy sets defined by all combinations of $m_A(x_1)$ and $m_{A^c}(x_1), \ldots, m_A(x_n)$ and $m_{A^c}(x_n)$.

At the midpoint nothing is distinguishable. At the vertices everything is distinguishable. These extremes represent the two ends of the spectrum of logic and set theory. In this sense the midpoint represents the black hole of set theory.

Paradox at the Midpoint

The midpoint is full of paradox. Classical logic and set theory forbid the midpoint by the same axioms, noncontradiction and excluded middle, that generate the of "paradoxes" or antinomies of bivalent systems. Where midpoint phenomena appear in Western thought, thoerists have invariably labeled them "paradoxes" or denied them altogether. Midpoint phenomena include the half-empty and half-full cup, the Taoist Yin-Yang, the liar from Crete who said that all Cretans are liars, Bertrand Russell's set of all sets that are not members of themselves, and Russell's barber.

Russell's barber is a bewhiskered man who lives in a town and who shaves. His barber shop sign says that he shaves a man if and only if he does not shave himself. So who shaves the barber? If he shaves himself, then by definition he does not. But if he does not shave himself, then by definition he does. So he does and he does not—contradiction ("paradox"). Gaines [1983] observed that we can numerically interpret this paradoxical circumstance as follows.

Let S be the proposition that the barber shaves himself and not-S that he does not. Then since S implies not-S and not-S implies S, the two propositions are logically equivalent: S = not-S. Equivalent propositions have the same truth values:

$$t(S) = t(\text{ not-}S) \tag{6}$$

$$= 1 - t(S) \quad . \tag{7}$$

Solving for $t(S)$ gives the midpoint point of the truth interval (the one-dimensional cube $[0, 1]$): $t(S) = \frac{1}{2}$. The midpoint is equidistant to the vertices 0 and 1. In the bivalent (two-valued) case, roundoff is impossible and paradox occurs. (6) and (7) describe the logical *form* of the many paradoxes, though different paradoxes involve different descriptions [Quine, 1987].

In bivalent logic both statements S and not-S must have truth value zero or unity. The fuzzy resolution of the paradox uses only the fact that the truth values are equal. It does not constrain their range. The midpoint value $\frac{1}{2}$ emerges from the structure of the problem and the order-reversing effect of negation.

The paradoxes of classical set theory and logic illustrate the price we pay for an arbitrary insistence on bivalence [Quine, 1981]. Scientists often insist on bivalence in the name of science. But in the end this insistence reduces to a mere cultural preference, a reflection of an educational predilection that goes back at least to Aristotle. Fuzziness shows that there are limits to logical certainty. We can no longer assert the laws of noncontradiction and excluded middle *for sure*—and *for free*.

Fuzziness caries with it intellectual responsibility. We must explain how fuzziness fits in bivalent systems, or vice versa. The fuzzy theorist must explain why so many people have been in error for so long. We now have the machinery to offer an explanation: We round off. Rounding off, quantizing, simplifies life and often costs little. We agree to call empty the near empty cup, and present the large pulse and absent the small pulse. We round off points inside the fuzzy cube to the nearest vertex. This roundoff heuristic works fine as a first approximation to describing the universe until we get near the midpoint of the cube. We find these phenomena harder to roundoff. In the logically extreme case, at the midpoint of the cube, the procedure breaks down completely because every vertex is equally close. If we still insist on bivalence, we can only give up and declare paradox.

Faced with midpoint phenomena, the fuzzy skeptic resembles the flat-earther, who denies that the earth's surface is curved, when she stands at the north pole, looks at her compass, and wants to go south.

Counting with Fuzzy Sets

How big is a fuzzy set? The size or cardinality of A, $M(A)$, equals the sum of the fit values of A:

$$M(A) = \sum_{i=1}^{n} m_A(x_i) \quad . \tag{8}$$

The count of $A = (\frac{1}{3}\ \frac{3}{4})$ equals $M(A) = \frac{1}{3} + \frac{3}{4} = \frac{13}{12}$. Some fuzzy theorists [Zadeh, 1983] call the cardinality measure M the *sigma-count*. The measure M generalizes [Kosko, 1986a] the classical counting measure of combinatorics and measure theory. (So (X, I^n, M) defines the fundamental measure space of fuzzy theory.) In general the measure M does not yield integer values.

The measure M has a natural geometric interpretation in the sets-as-points framework. $M(A)$ equals the magnitude of the vector drawn from the origin to the fuzzy set A, as Figure 16.4 illustrates.

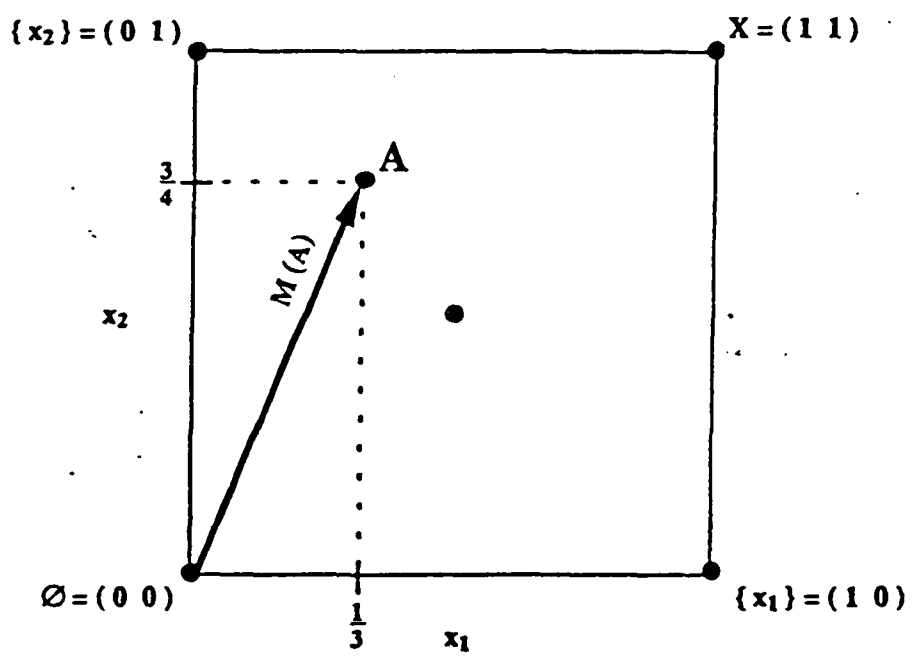


Figure 16.4 The count $M(A)$ of A equals the fuzzy Hamming norm (l^1 norm) of the vector drawn from the origin to A.

Consider the l^p distance between fuzzy sets A and B in I^n:

$$l^p(A, B) = \sqrt[p]{\sum_{i=1}^{n} |m_A(x_i) - m_B(x_i)|^p} , \tag{9}$$

where $1 \leq p \leq \infty$. The l^2 distance is the physical Euclidean distance actually illustrated in the figures. The simplest distance is the l^1 or fuzzy Hamming distance, the sum of the absolute fit differences. We shall use fuzzy Hamming distance throughout, though all results admit a general l^p formulation. Using the fuzzy Hamming distance we can rewrite the count M as the desired l^1 norm:

$$M(A) = \sum_{i}^{n} m_A(x_i) \tag{10}$$

$$= \sum_{i} |m_A(x_i) - 0| \tag{11}$$

$$= \sum_{i} |m_A(x_i) - m_{\emptyset}(x_i)| \tag{12}$$

$$= l^1(A, \emptyset) \quad . \tag{13}$$

THE FUZZY ENTROPY THEOREM

How fuzzy is a fuzzy set? We measure fuzziness with by a *fuzzy entropy* measure. Entropy is a generic notion. It need not be probabilistic. Entropy measures the uncertainty of a system or message. A fuzzy set describes a type of system or message. Its uncertainty equals its fuzziness.

The fuzzy entropy of A, $E(A)$, varies from 0 to 1 on the unit hypercube I^n. Only the

cube vertices have zero entropy, since nonfuzzy sets are unambiguous. The cube midpoint uniquely has unity or maximum entropy. Fuzzy entropy smoothly increases as a set point moves from any vertex to the midpoint. Klir [1988] discusses the algebraic requirements for fuzzy entropy measures.

Simple geometric considerations lead to a ratio form for the fuzzy entropy [Kosko, 1986b]. The closer the fuzzy set A is to the nearest vertex A_{near}, the farther A is from the farthest vertex A_{far}. The farthest vertex A_{far} resides opposite the long diagonal from the nearest vertex A_{near}. Let a denote the distance $l^1(A, A_{near})$ to the nearest vertex, and let b denote the distance $l^1(A, A_{far})$ to the farthest vertex. Then the fuzzy entropy equals the ratio of a to b:

$$E(A) = \frac{a}{b} = \frac{l^1(A, A_{near})}{l^1(A, A_{far})} \quad . \tag{14}$$

Figure 16.5 shows the sets-as-points interpretation of the fuzzy entropy, where $A = (\frac{1}{3}\ \frac{3}{4})$, $A_{near} = (0\ 1)$, and $A_{far} = (1\ 0)$. So $a = \frac{1}{3} + \frac{1}{4} = \frac{7}{12}$ and $b = \frac{2}{3} + \frac{3}{4} = \frac{17}{12}$. So $E(A) = \frac{7}{17}$.

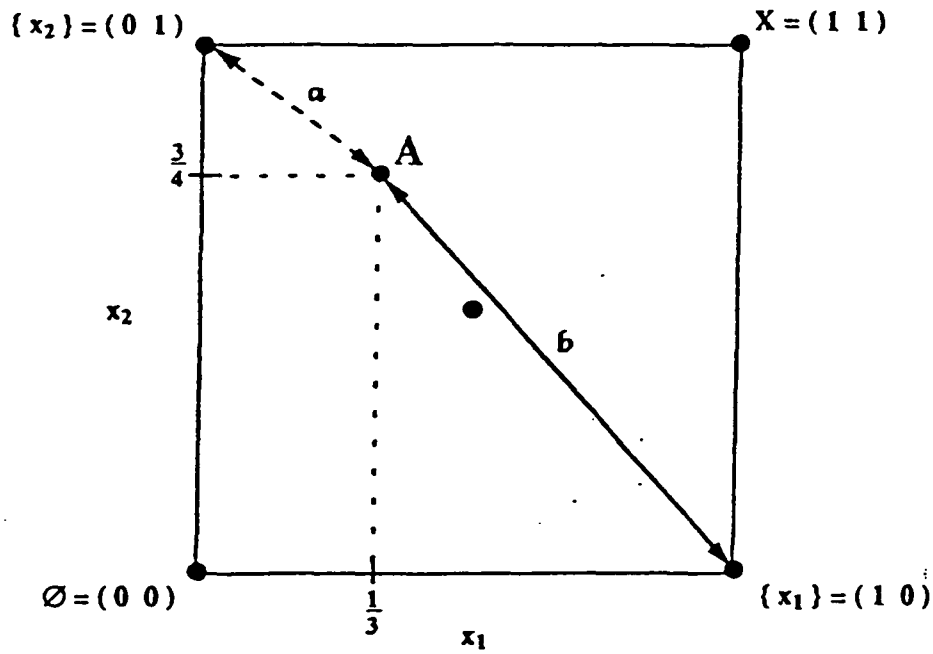


Figure 16.5 Fuzzy entropy, $E(A) = \frac{a}{b}$, balances distance to nearest vertex with distance to farthest vertex.

Alternatively, if you read this in a room, you can imagine the room as the unit cube I^3 and your head as a fuzzy set in it. Once you locate the nearest corner of the room, the farthest corner resides opposite the long diagonal emanating from the nearest corner. If you put your head in a corner, then $a = 0$, and so $E(A) = 0$. If you put your head in the metrical center of the room, every corner is nearest and farthest. So $a = b$, and $E(A) = 1$.

Overlap and underlap characterize set fuzziness. So we can expect them to affect the measure of fuzziness. Figure 16.3 shows the connection. By symmetry, each of the four points A, A^c, $A \cap A^c$, and $A \cup A^c$ is equally close to its nearest vertex. The common distance equals a. Similarly, each point is equally far from its farthest vertex. The common

distance equals b. One of the first four distances is the count $M(A \cap A^c)$. One of the second four distances is the count $M(A \cup A^c)$. This gives a geometric proof of the Fuzzy Entropy Theorem [Kosko, 1986b-87], which states that fuzziness consists of counted violations of the law of noncontradiction balanced with counted violations of the law of excluded middle.

$$\textbf{Fuzzy Entropy Theorem:} \qquad E(A) = \frac{M(A \cap A^c)}{M(A \cup A^c)} \tag{15}$$

An algebraic proof is straightforward. The completed fuzzy square in Figure 16.6, contains a geometric proof (in this special case).

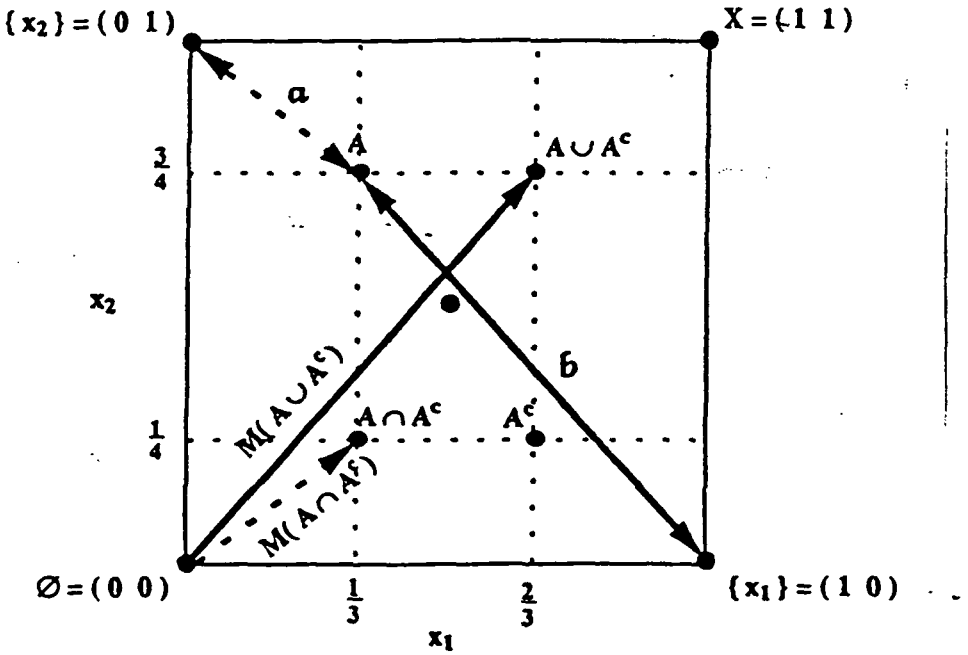


Figure 16.6 Geometry of the Fuzzy Entropy Theorem. By symmetry each of the four points on the completed fuzzy square is equally close to its nearest vertex and equally far from its farthest vertex.

The Fuzzy Entropy Theorem explains why set fuzziness begins where Western logic ends. When sets (or propositions) obey the laws of noncontradiction and excluded middle,

overlap is empty and underlap is exhaustive So $M(A \cap A^c) = 0$ and $M(A \cup A^c) = n$, and thus $E(A) = 0$.

The Fuzzy Entropy Theorem also provides a first-principles derivation of the basic fuzzy set operations of minimum (intersection), maximum (union), and order reversal (complementation) proposed in 1965 by Zadeh at the inception of fuzzy theory. (Lukasiewicz first proposed these operations for continuous or fuzzy logics in the 1920s [Rescher, 1969].)

For the fuzzy theorist, this result also shows that triangular norms or T-norms [Klir, 1988], which generalize conjunction or intersection, and the dual triangular co-norms C, which generalize disjunction or union, do not have the first-principles status of min and max. For, the triangular-norm inequalities,

$$T(x, y) \leq \min(x, y) \leq \max(x, y) \leq C(x, y) \quad , \tag{16}$$

show that replacing min with any T in the numerator term $M(A \cap A^c)$ can only make the numerator smaller. Replacing max with any C in the term $M(A \cup A^c)$ can only make the denominator larger. So any T or C not identically min or max makes the ratio smaller, strictly smaller if A is fuzzy. Then the entropy theorem does not hold, and the resulting pseudo-entropy measure does not equal unity at the midpoint, though it continues to be maximized there. We can see this with the product T-norm [Prade, 1985] $T(x, y) = xy$ and its DeMorgan dual co-norm $C(x, y) = 1 - T(1 - x, 1 - y) = x + y - xy$, or with the bounded sum T-norm $T(x, y) = \max(0, x + y - 1)$ and DeMorgan dual $C(x, y) = \min(1, x+y)$. The Entropy Theorem similarly fails in general if the negation or complementation operator $N(x) = 1 - x$ with a parameterized operator $N_a(x) = \frac{1 - x}{1 + ax}$ for nonzero $a > -1$.

All probability distributions, all sets A with $M(A) = 1$, in I^n form a $n-1$ dimensional simplex S^n. In the unit square the probability simplex equals the negatively sloped diagonal line. In the unit 3-cube it equals a solid triangle. In the unit 4-cube it equals a tetrahedron, and so on up.

If no probabilistic fit value p_i satisfies $p_i > \frac{1}{2}$, then the Fuzzy Entropy Theorem implies [Kosko, 1987] that the the distribution P has fuzzy entropy $E(P) = \frac{1}{n-1}$. Else

$E(P) < \frac{1}{n-1}$. So the probability simplex S^n is entropically degenerate for large dimensions n. This result also shows that the uniform distribution $(\frac{1}{n}, \ldots, \frac{1}{n})$ maximizes fuzzy entropy on S^n but not uniquely. This in turn shows that fuzzy entropy differs from the average-information measure of probabilistic entropy, which the uniform distribution maximizes uniquely.

The Fuzzy Entropy Theorem implies that, analogous to $\log \frac{1}{p}$, a unit of fuzzy information equals $\frac{f}{1-f}$ or $\frac{1-f}{f}$, depending on whether the fit value f obeys $f \leq \frac{1}{2}$ or $f \geq \frac{1}{2}$.

The event x can be ambiguous or clear. It is ambiguous if f equals approximately $\frac{1}{2}$ and clear if f equals approximately 1 or 0. If an ambiguous event occurs, is observed, is disambiguated, etc., then it is maximally informative: $E(f) = E(\frac{1}{2}) = 1$. If a clear event occurs, is observed, etc., it is minimally informative: $E(f) = E(0) = E(1) = 0$. This agrees with the information interpretation of the probabilistic entropy measure $\log \frac{1}{p}$, where the occurrence of a sure event ($p = 1$) is minimally informative (zero entropy) and the occurrence of an impossible event ($p = 0$) is maximally informative (infinite entropy).

THE SUBSETHOOD THEOREM

Sets contain subsets. A is a *subset* of B, denoted $A \subset B$, if and only if every element in A is an element of B. The power set 2^B contains all of B's subsets. So, alternatively [Bandler-Kohout, 1980], A is a subset of B iff A belongs to B's power set:

$$A \subset B \quad \text{if and only if} \quad A \in 2^B . \qquad (17)$$

The subset relation corresponds to the implication relation in logic. In classical logic *truth* maps the set of statements $\{S\}$ to two truth values: $t : \{S\} \longrightarrow \{0, 1\}$. Consider the truth-tabular definition of implication for bivalent propositions P and Q:

P	Q	P ⟶ Q
0	0	1
0	1	1
1	0	0
1	1	1

The implication is false if and only if the antecedent P is true and the consequent Q is false—when "truth implies falsehood."

The same holds for subsets. Representing sets as bivalent functions or "indicator" functions $m_A : X \longrightarrow \{0, 1\}$, A is a subset of B iff there is no element x that belongs to A but not to B, or $m_A(x) = 1$ but $m_B(x) = 0$. We can rewrite this membership-function definition as

$$A \subset B \quad \text{if and only if} \quad m_A(x) \leq m_B(x) \text{ for all } x \ . \tag{18}$$

Zadeh [1965] proposed the same relation for fuzzy set containment. We refer to this as the *dominated membership function relationship*. If $A = (.3\ 0\ .7)$ and $B = (.4\ .7\ .9)$, then A is a fuzzy subset of B, but B is not a fuzzy subset of A. Either fuzzy set A is, or is not, a fuzzy subset of B. So the relation of fuzzy subsethood is *not* fuzzy. It is either black or white.

The sets-as-points view asks a geometric question: What do all fuzzy subsets of B look like? What does the *fuzzy power set* of B—$F(2^B)$, the set of all fuzzy subsets of B—look like? The dominated membership function relationship implies that $F(2^B)$ defines the hyper-rectangle snug against the origin in a unit hypercube with side lengths equal to the fit values $m_A(x_i)$. Figure 7 displays the fuzzy power set of the set $B = (\frac{1}{3}\ \frac{2}{3})$. $F(2^B)$ has infinite count if B is not empty. For finite-dimensional sets, we can measure the size of $F(2^B)$ [Kosko, 1987] as the Lebesgue measure or volume $V(B)$, the product of the fit values:

$$V(B) = \prod_{i=1}^{n} m_B(x_i) \ . \tag{19}$$

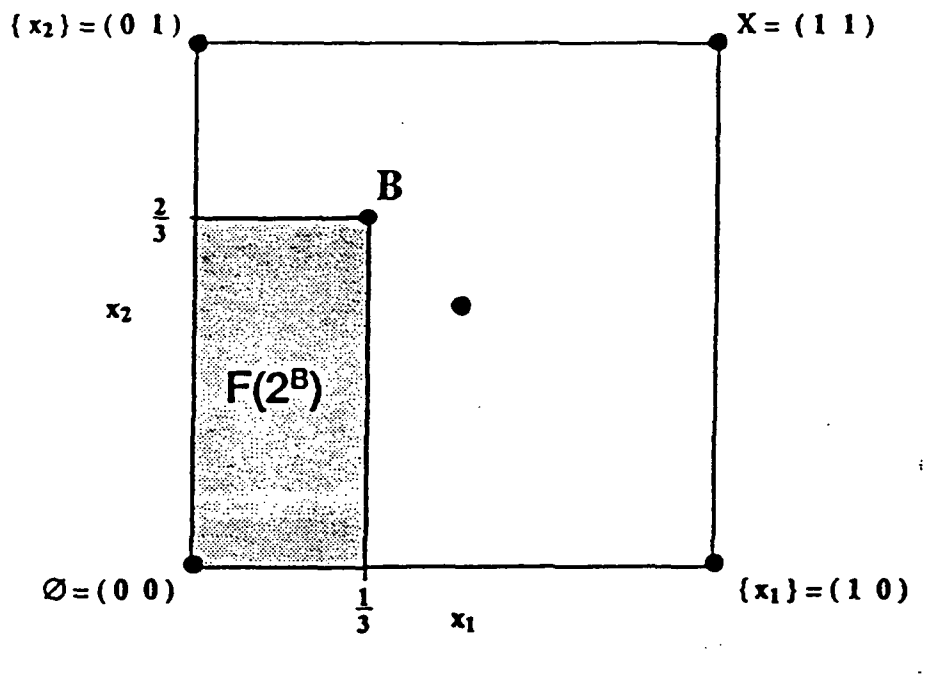


Figure 16.7 Fuzzy power set $F(2^B)$ as a hyper-rectangle in the fuzzy cube. Side lengths equal the fit values $m_B(x_i)$. The size or volume of $F(2^B)$ equals the product of the fit values.

Figure 16.7 illustrates that $F(2^B)$ is not a fuzzy set. Either cube point A is or is not in the hyper-rectangle $F(2^B)$. Different points A outside the hyper-rectangle $F(2^B)$ resemble subsets of B to different degrees. The bivalent definition of subsethood ignores this.

The natural generalization defines fuzzy subsets on $F(2^B)$: Some sets A belong to $F(2^B)$ to different degrees. Then the abstract membership function $m_{F(2^B)}(A)$ can equal any number in [0, 1]. This defines degrees of subsethood.

Let $S(A, B)$ denote the degree to which A is a subset of B:

$$S(A, B) = \text{Degree}(A \subset B) \tag{20}$$

$$= m_{F(2^B)}(A) \quad . \tag{21}$$

$S(.,\ .)$ denotes the *subsethood measure*. $S(.,\ .)$ takes values in [0,1]. We will see that it provides the fundamental, unifying structure in fuzzy theory.

We want to measure $S(A, B)$. We will first present an earlier [Kosko, 1986b-87] algebraic derivation of the subsethood measure $S(A, B)$. We will then present a new, more fundamental, geometric derivation.

We call the algebraic derivation the *fit-violation strategy*. Intuitively we study a law by breaking it. Consider again the dominated membership function relationship: $A \subset B$ if and only if $m_A(x) \leq m_B(x)$ for all x in X.

Suppose element x_v violates the dominated membership function relationship: $m_A(x_v) > m_B(x_v)$. Then A is not a subset of B, at least not totally. Suppose further that the dominated membership inequality holds for all other elements x. Only element x_v violates the relationship. For instance, X may consist of one hundred values: $X = \{x_1, \ldots, x_{100}\}$. The violation might occur, say, with the first element: $x_1 = x_v$. Then intuitively A is largely a subset of B. Suppose that X contains a thousand elements, or a trillion elements, and only the first element violates (18). Then it seems A is overwhelmingly a subset of B; perhaps $S(A, B) = .999999999999$.

This example suggests we should count fit violations in magnitude and frequency. The greater the violations in magnitude, $m_A(x_v) - m_B(x_v)$, and the greater the number of violations relative to the size $M(A)$ of A, the less A is a subset of B or, equivalently, the more A is a *superset* of B. For, both intuitively and by (18), supersethood and subsethood relate as additive opposites:

$$\text{SUPERSETHOOD}(A, B) = 1 - S(A, B) \quad . \tag{22}$$

We count violations by adding them. If we sum over all x, the summand should equal

$m_A(x_v) - m_B(x_v)$ when this difference is positive, and equal zero when it is nonpositive. So the summand equals $\max(0, m_A(x) - m_B(x))$. So the unnormalized count equals the sum of these maxima:

$$\sum_{x \in X} \max(0, m_A(x) - m_B(x)). \tag{23}$$

The count $M(A)$ provides a simple, and appropriate, normalization factor. Below we formally arrive at $M(A)$ by examining boundary cases in the geometric approach to subsethood. We can assume $M(A) > 0$, since $M(A) = 0$ if and only if A is empty. The empty set trivially satisfies the dominated membership function relationship (18). So it is a subset of every set. Normalization gives the minimal measure of nonsubsethood, of supersethood:

$$\text{SUPERSETHOOD}(A, B) = \frac{\sum_x \max(0, m_A(x) - m_B(x))}{M(A)} . \tag{24}$$

Then subsethood is the negation of this ratio. This gives the minimal fit-violation measure of subsethood:

$$S(A, B) = 1 - \frac{\sum_x \max(0, m_A(x) - m_B(x))}{M(A)} . \tag{25}$$

The subsethood measure may appear ungraceful at first, but it behaves as it should. $S(A, B) = 1$ if and only if (18) holds. For if (18) holds, (23) sums zero violations. Then $S(A, B) = 1 - 0 = 1$. If $S(A, B) = 1$, every numerator summand equals zero. So no violation occurs. At the other extreme, $S(A, B) = 0$ if and only if B is the empty set. The empty set uniquely contains no proper subsets, fuzzy or nonfuzzy. Degrees of subsethood occur between these extremes: $0 < S(A, B) < 1$.

The subsethood measure relates to logical implication. Viewed at the 1-dimensional level of fuzzy logic, and so ignoring the normalizing count ($M(A) = 1$), the subsethood measure reduces to the Lukasiewicz implication operator:

$$S(A, B) = 1 - \max(0, m_A - m_B) \tag{26}$$

$$= 1 - [1 - \min(1 - 0, 1 - (m_A - m_B))] \tag{27}$$

$$= \min(1, 1 - m_A + m_B) \tag{28}$$

$$= t_L(A \longrightarrow B) \quad . \tag{29}$$

The min(.) operator in (28) clearly generalizes the above truth-tabular definition of bivalent implication.

Consider the fit vectors $A = (.2\ 0\ .4\ .5)$ and $B = (.7\ .6\ .3\ .7)$. Neither set is a proper subset of the other. A is almost a subset of B but not quite since $m_A(x_3) - m_B(x_3) = .4 - .3 = .1 > 0$. Hence $S(A, B) = 1 - \frac{.1}{1.1} = \frac{10}{11}$. Similarly $S(B, A) = 1 - \frac{1.3}{2.3} = \frac{10}{23}$.

Subsethood applies to nonfuzzy sets. Consider the sets $C = \{x_1, x_2, x_3, x_5, x_7, x_9, x_{10}, x_{12}, x_{14}\}$ and $D = \{x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{12}, x_{13}, x_{14}\}$ with corresponding bit vectors

$$C = (1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 1)$$

$$D = (0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1) \quad .$$

C and D are not subsets of each other. But C should very nearly be a subset of D since only x_1 violates (18). We find $S(C, D) = 1 - \frac{1}{9} = \frac{8}{9}$ while $S(D, C) = 1 - \frac{4}{12} = \frac{2}{3}$. So D is more a subset of C than it is not. This holds because the two sets are largely equivalent. They have much overlap: $M(C \cap D) = 8$. This observation anticipates the Fuzzy Subsethood Theorem presented below.

We now turn to a new and purely geometric derivation of the subsethood operator $S(A, B)$. Consider the sets-as-points geometry of subsethood in Figure 16.7. Set A is either in the hyper-rectangle $F(2^B)$ or not in it. Intuitively $S(A, B)$ should approach unity as A approaches the fuzzy power set $F(2^B)$. $S(A, B)$ should decrease, and the supersethood measure $1 - S(A, B)$ should increase, as A moves from $F(2^B)$.

So the key idea is metrical: *How close is A to* $F(2^B)$? Let $d(A, F(2^B))$ denote this l^p distance defined in (9). $d(A, B')$ denotes the distance between A and point B' in the hyper-rectangle, and $B' \subset B$. Distance $d(A, F(2^B))$ equals the smallest such distance. Since the hyper-rectangle $F(2^B)$ is closed and bounded (compact) and convex, some subset B^* of B achieves this minimum distance. So the infimum, the greatest lower bound, equals the distance $d(A, B^*)$:

$$d(A, F(2^B)) = \inf \{d(A, B') : B' \in F(2^B)\} \tag{30}$$

$$= d(A, B^*) \quad . \tag{31}$$

We can easily locate the closest set B^* in the hypercube geometry. If A is a subset of B—if A is in the hyper-rectangle $F(2^B)$—then A equals the closest subset: $A = B^*$. So suppose A is not a proper subset of B. Then A lies outside the hyper-rectangle $F(2^B)$.

We cal slice the unit cube I^n into 2^n hyper-rectangles by extending the sides of $F(2^B)$ to hyperplanes. The hyperplanes intersect perpendicularly (orthogonally), at least in the Euclidean case. $F(2^B)$ defines one of the hyper-rectangles. The hyper-rectangle interiors correspond to the 2^n cases whether $m_A(x_i) < m_B(x_i)$ or $m_A(x_i) > m_B(x_i)$ for fixed B and arbitrary A. The edges correspond to the loci of points where some $m_A(x_i) = m_B(x_i)$.

The 2^n hyper-rectangles classify as *mixed* or *pure* membership domination. In the pure case either $m_A < m_B$, or $m_A > m_B$, holds in the hyper-rectangle interior for all x and all interior points A. In the mixed case $m_A(x_i) < m_B(x_i)$ holds for some of the coordinates

x_i, and $m_A(x_j) > m_B(x_j)$ holds for the remaining coordinates x_j in the interior for all interior A. So there are only two pure membership-domination hyper-rectangles, the set of proper subsets $F(2^B)$ and the set of proper supersets, which includes X.

Figure 16.8 illustrates how the fuzzy power set $F(2^B)$ of $B = (\frac{1}{3} \quad \frac{2}{3})$ linearly extends to partition the unit square into 2^2 rectangles. The non-subsets A_1, A_2, and A_3 reside in distinct quadrants. The northwest and southeast quadrants define the mixed membership-domination rectangles. The southwest and the northeast quadrants define the pure rectangles.

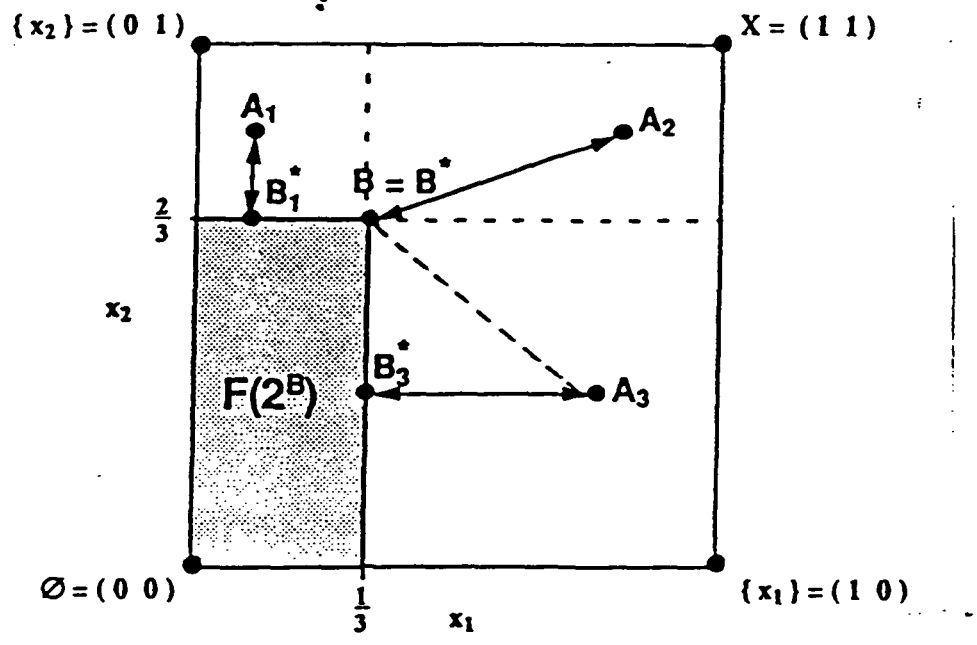


Figure 16.8 Partition of hypercube I^n into 2^n hyper-rectangles by linearly extending the edges of $F(2^B)$. We find the nearest points B_1^* and B_3^* to points A_1 and A_3 in the northwest and southeast quadrants by the normals from $F(2^B)$ to A_1 and A_3. The nearest point B^* to point A_2 in the northeast quadrant is B itself. This "orthogonal" optimality condition allows $d(A, B)$ by the general Pythagorean Theorem as the hypotenuse in an l^p "right" triangle.

B is the nearest set B^* to A in the pure superset hyper-rectangle. To find the nearest set B^* in the mixed case we draw a perpendicular (orthogonal) line segment from A to $F(2^B)$. Convexity of $F(2^B)$ is ultimately responsible. In Figure 16.8 the perpendicular lines from A_1 and A_3 intersect line edges (1-dimensional linear subspaces) of the rectangle $F(2^B)$. The line from A_2 to B, the corner of $F(2^B)$, is degenerately perpendicular since B is a zero-dimensional linear subspace.

These "orthogonality" conditions also hold in three dimensions. Let your room again be the unit 3-cube. Consider a large dictionary fit snugly against the floor corner corresponding to the origin. Point B equals the dictionary corner farthest from the origin. Extending the three exposed faces of the dictionary partitions the room into 8 octants. The dictionary occupies one octant. We connect points in the other 7 octants to the nearest points on the dictionary by drawing lines, or tying strings, that perpendicularly intersect one the three exposed faces, or one of the three exposed edges, or the corner B.

The "orthogonality" condition invokes the l^p-version of the Pythagorean Theorem. For our l^1 purposes:

$$d(A,\ B) \ = \ d(A,\ B^*) \ + \ d(B,\ B^*) \quad . \tag{32}$$

The more familiar l^2-version, actually pictured in Figure 16.8, requires squaring these distances. For the general ℓ^p case:

$$||A - B||^p \ = \ ||A - B^*||^p \ + \ ||B^* - B||^p \ , \tag{33}$$

or equivalently,

$$\sum_{i=1}^{n} |a_i - b_i|^p \ = \ \sum_{i=1}^{n} |a_i - b_i^*|^p \ + \ \sum_{i=1}^{n} |b_i^* - b_i|^p \ . \tag{34}$$

Equality holds for all $p \geq 1$ since, as is clear from Figure 16.8 or 16.10 and, in general, from the algebraic argument below, either $b_i^* = a_i$ or $b_i^* = b_i$.

This Pythagorean equality is surprising. We have come to think of the Pythagorean

Theorem (and orthogonality) as an ℓ^2 or Hilbert-space property. Yet here it holds in *every* ℓ^p space—if B^* is the set in $F(2^B)$ closest to A in ℓ^p distance. Of course for other sets strict inequality holds in general if $p \neq 2$. This suggests a special status for the closest set B^*. We shall see below that the Subsethood Theorem confirms this suggestion. We shall use the term "orthogonality" loosely to refer to this ℓ^p Pythagorean relationship, while remembering its customary restriction to ℓ^2 spaces and inner products.

A natural interpretation defines supersethood as the distance $d(A, F(2^B)) = d(A, B^*)$. Supersethood increases with this distance; subsethood decreases with it. To keep supersethood, and thus subsethood, unit-interval valued, we must suitably normalize the distance.

A constant provides the simplest normalization term for $d(A, B^*)$. That constant equals the maximum unit-cube distance, $n^{\frac{1}{p}}$ in the general l^p case and n in our ℓ^1 case. This gives the candidate subsethood measure

$$S(A, B) = 1 - \frac{d(A, B^*)}{n} \quad . \tag{35}$$

This candidate subsethood measure fails in the boundary case when B is the empty set. For then $d(A, B^*) = d(A, B) = M(A)$. So the measure in (35) gives $S(A, \emptyset) = 1 - \frac{M(A)}{n} > 0$. Equality holds exactly when $A = X$. But the empty set has no subsets. Only normalization factor $M(A)$ satisfies this boundary condition. Of course $M(A) = n$ when $A = X$. Explicitly we require $S(A, \emptyset) = 0$, as well as $S(\emptyset, A) = 1$.

Normalizing by n also treats all equidistant points the same. Consider points A_1 and A_2 in Figure 16.9. Both points are equidistant to their nearest $F(2^B)$ point: $d(A_1, B_1^*) = d(A_2, B_2^*)$. But A_1 is closer to B than A_2 is. In particular A_1 is closer to the horizontal line defined by the fit value $m_B(x_2) = \frac{2}{3}$. The count $M(A)$ reflects this: $M(A_1) > M(A_2)$. The count gap $M(A_1) - M(A_2)$ arises from the fit gap involving x_1, and reflects $d(A_1, B) < d(A_2, B)$. In general the count $M(A)$ relates to this distance, as we can see by checking extreme cases of closeness of A to B (and drawing some diamond-shaped l^1 spheres centered at B). Indeed if $m_A > m_B$ everywhere, $d(A, B) = M(A) - M(B)$.

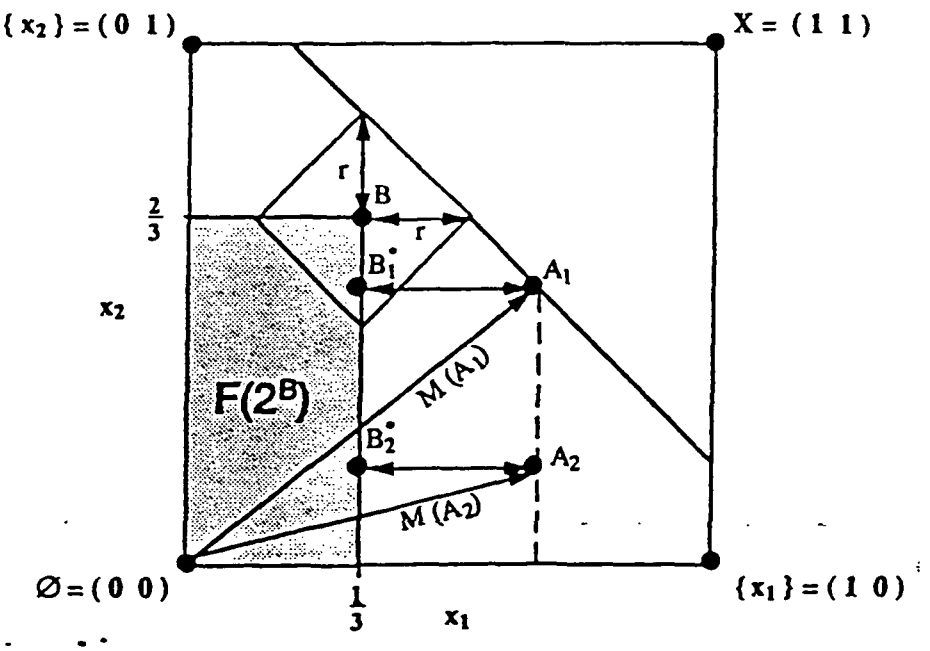


Figure 16.9 Dependence of subsethood on the count $M(A)$. A_1 and A_2 are equidistant to $F(2^B)$ but A_1 is closer to B than A_2 is; correspondingly, $M(A_1) > M(A_2)$. Loci of points A of constant count $M(A)$ define line segments parallel to the negatively sloping long diagonal. l^1 spheres centered at B are diamond shaped.

Since $F(2^B)$ fits snugly against the origin, the count $M(A)$ in any of the other $2^n - 1$ hyper-rectangles can be only larger than the count $M(B^*)$ of the nearest $F(2^B)$ points. The normalization choice of n leaves the candidate subsethood measure indifferent to which of the $2^n - 1$ hyper-rectangles contains A and to where A resides is in the hyper-rectangle. Each point in each hyper-rectangle involves a different combination of fit-violations and satisfactions. The normalization choice of $M(A)$ reflects this fit-violation structure and behaves appropriately in boundary cases.

The normalization choice $M(A)$ leads to the subsethood measure

$$S(A, B) = 1 - \frac{d(A, B^*)}{M(A)} \quad . \tag{36}$$

We now show that this measure equals the subsethood measure (25) derived algebraically above.

Let B' be any subset of B. Then by definition the nearest subset B^* obeys the inequality:

$$\sqrt[p]{\sum_{i=1}^{n} |a_i - b_i^*|^p} \leq \sqrt[p]{\sum_{i=1}^{n} |a_i - b_i'|^p} \quad , \tag{37}$$

where for convenience $a_i = m_A(x_i)$, and $b_i = m_B(x_i)$. We will assume $p = 1$ but the following characterization of b_i^* holds for any $p > 1$.

"Orthogonality" implies $a_i \geq b_i^*$. So first suppose $a_i = b_i^*$. This equality holds if and only if no violation occurs: $a_i \leq b_i$. (If this condition holds for all i, then $A = B^*$.) So $\max(0, a_i - b_i) = 0$. Next suppose $a_i > b_i^*$. This inequality holds if and only if a violation occurs: $a_i > b_i$. (If this holds for all i, then $B = B^*$.) So $b_i^* = b_i$ since B^* is the subset of B nearest to A. Equivalently, $a_i > b_i$ holds if and only if $\max(0, a_i - b_i) = a_i - b_i$. The two cases together prove that $\max(0, a_i - b_i) = |a_i - b_i^*|$. Summing over all x_i gives

$$d(A, B^*) = \sum_{i=1}^{n} \max(0, m_A(x_i) - m_B(x_i)) \quad . \tag{38}$$

So the two subsethood measures (25) and (36) are equivalent.

This proof also proves a deeper characterization of the optimal subset B^*:

$$B^* = A \cap B \quad . \tag{39}$$

For if a violation occurs, then $a_i > b_i$, and $b_i = b_i^*$. So $\min(a_i, b_i) = b_i^*$. Otherwise $a_i = b_i^*$, and so $\min(a_i, b_i) = b_i^*$. So $B^* = A \cap B$.

This in turn proves that B^* is a point of double optimality. B^* is both the subset of B nearest A, and A^*, the subset of A nearest to B:

$$d(B,\ F(2^A)) \ = \ d(B,\ A^*) \ = \ d(B,\ B^*) \quad . \tag{10}$$

Figure 16.10 illustrates that $B^* \ = \ A \cap B \ = \ A^*$ identifies the set within both the hyper-rectangle $F(2^A)$ and the hyper-rectangle $F(2^B)$ that has maximal count $M(A \cap B)$.

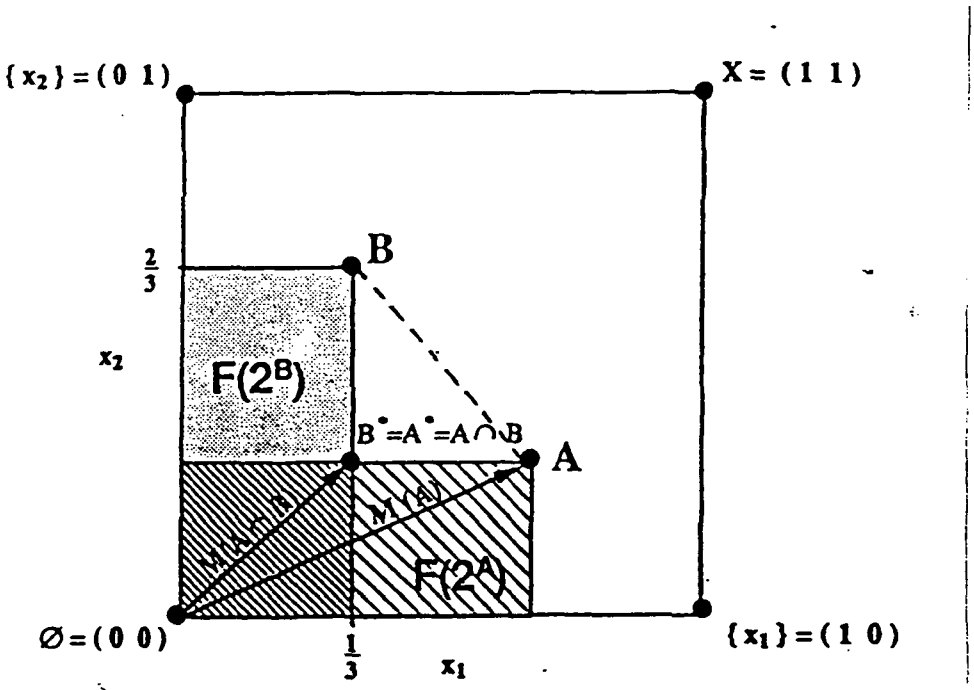


Figure 16.10. B^* as both the subset of B nearest A and the subset A^* of A nearest B : $B^* \ = \ A^* \ = \ A \cap B$. The distance $d(A,\ B^*) \ = \ M(A) - M(A \cap B)$ illustrates the Subsethood Theorem.

Figure 16.10 also shows that the distance $d(A,\ B^*)$ equals a vector magnitude difference: $d(A,\ B^*) \ = \ M(A) \ - \ M(A \cap B)$. Dividing both sides of this equality by $M(A)$ and rearranging proves a still deeper structural characterization of subsethood, the Subsethood Theorem.

$$\textbf{SubsethoodTheorem.} \qquad S(A,\ B) \ = \ \frac{M(A \cap B)}{M(A)} \quad . \tag{41}$$

The Subsethood Theorem immediately implies a Bayes Theorem:

Subsethood Theorem.

$$S(A,\ B) \ = \ \frac{M(B)\ S(B,A)}{M(A)} \quad , \tag{42}$$

since (41) implies $M(A \cap B) = M(B)\ S(B,A)$.

The ratio form of the subsethood measure $S(A,\ B)$ has the same ratio form as the conditional probability $P(B|A)$ has in (1). We *derived* the ratio form for the subsethood measure $S(A,\ B)$ but *assumed* it for the conditional probability $P(B|A)$. Since every probability is a conditional probability, $P(A) = P(A|X)$, this suggests we can reduce probability to subsethood. We shall argue that this reduction holds both frequentist or "objective" probability and axiomatic or Bayesian or "subjective" probability.

Consider the physical interpretation of randomness as the relative frequency n_A/n. n_A denotes the number of successes that occur in n trials. Historically probabilists have called the success ratio (or its limit) n_A/n the "probability of success" or $P(A)$. We can now derive the relative-frequency definition of probability as $S(X,\ A)$, the degree to which a bivalent superset X, the sample space, is a subset of its own subset A. The concept of "randomness" never enters the deterministic set-theoretic framework. This holds equally for flipping coins, drawing balls from urns, or computing Einstein-Bose statistics.

Suppose A is a nonfuzzy subset of X. Then

$$S(X,A) \ = \ \frac{M(A \cap X)}{M(X)} \tag{43}$$

$$= \frac{M(A)}{M(X)} \quad , \tag{44}$$

$$= \frac{n_A}{n} \quad . \tag{45}$$

The n elements of X constitute the *de facto* universe of discourse of the "experiment." (We can take the limit of the ratio $S(X,\ A)$ if it mathematically makes sense to do so [Kac, 1959].) The "probability" $\frac{n_A}{n}$ has reduced to a degree of subsethood, a purely fuzzy set-theoretical relationship. Perhaps if, centuries ago, scientists had developed set theory before they formalized gambling, the undefined notion of "randomness" might never have culturally prevailed, if even survived, in the age of modern science.

The measure of overlap $M(A \cap X)$ provides the key component of relative frequency. This count does not involve "randomness". $M(A \cap X)$ counts which elements are identical or similar. The phenomena themselves are deterministic and black or white. The same situation gives the same number. We may use the number to place bets or to switch a phone line, but it remains part of the description of a specific state of affairs. We need not invoke an undefined "randomness" to further describe the situation.

Subsethood subsumes elementhood. We can interpret the membership degree $m_A(x_i)$ as the subsethood degree $S(\{x_i\},\ A)$, where $\{x_i\}$ denotes a singleton subset or "element" x_i of X. $\{x_i\}$ corresponds to a bit vector with a 1 in the ith slot and 0s elsewhere: $\{x_i\} = (0,\ldots,0,1,0,\ldots,0)$. If we view A as the fit vector $(a_1,\ldots,a_i,\ldots,a_n)$, then $\{x_i\} \cap A = (0,\ldots,0,a_i,0,\ldots,0)$, the ith coordinate projection. Since the count $M(\{x_i\})$ equals one, the Subsethood Theorem gives

$$S(\{x_i\},\ A) = \frac{M(\{x_i\} \cap A)}{M(\{x_i\})} \tag{46}$$

$$= M(\ (0,\ldots,a_i,\ldots,0)\ ) \tag{47}$$

$$= a_i \tag{48}$$

$$= m_A(x_i) \tag{49}$$

$$= \text{Degree}(x_i \in A) \quad . \tag{50}$$

So subsethood reduces to elementhood if antecedent sets are bivalent singleton sets.

The subsethood orthogonality conditions project A onto the facing side of the hyper-rectangle $F(2^B)$. This projection gives the "normal equations" of least-squares parameter estimation [Sorenson, 1980], a version of which we saw in Chapter 5. In general for two R^n vectors x and y, we project x onto y to give the projection vector $\mathbf{p} = c\,\mathbf{y}$. The difference $\mathbf{x} - \mathbf{p}$ is orthogonal to $\mathbf{y}$: $(\mathbf{x} - \mathbf{p}) \perp \mathbf{y}$. So

$$0 = (\mathbf{x} - \mathbf{p})\,\mathbf{y}^T \tag{51}$$

$$= (\mathbf{x} - c\mathbf{y})\,\mathbf{y}^T \tag{52}$$

$$= \mathbf{x}\mathbf{y}^T - c\mathbf{y}\mathbf{y}^T \quad , \tag{53}$$

where column vector $\mathbf{y}^T$ denotes the transpose of row vector $\mathbf{y}$. (53) gives the projection coefficient c as the familiar normal equations:

$$c = \frac{\mathbf{x}\,\mathbf{y}^T}{\mathbf{y}\mathbf{y}^T} \tag{54}$$

$$= \frac{\sum_{i=1}^{n} x_i\, y_i}{\sum_{i=1}^{n} x_i^2} \quad . \tag{55}$$

Consider the unit square in Figure 16.10 with the same A, say $A = (\frac{3}{4}\ \frac{1}{2})$. But suppose we shift B directly to the left to $B = (0\ \frac{2}{3})$. This contracts the rectangle $F(2^B)$ to the line segment $[0\ \frac{2}{3}]$ along the vertical axis. These assumptions simplify the correlation

mathematics yet still preserve the least-squares structure. We expect that $B^* = cB$, or $cB = A \cap B$, when we project A onto $F(2^B)$ or, equivalently in this special case, when we project A onto B. The intersection $A \cap B$ equals the minimum fit vector $(0\ \frac{1}{2})$, $AB^T = 0 + \frac{2}{6} = \frac{1}{3}$, and $BB^T = 0 + (\frac{2}{3})^2 = \frac{4}{9}$. Then

$$c = \frac{AB^T}{BB^T}$$

$$= \frac{\frac{1}{3}}{\frac{4}{9}} = \frac{3}{4} \quad ,$$

and

$$B^* = c\,B$$

$$= \frac{3}{4}(0 \quad \frac{2}{3})$$

$$= (0 \quad \frac{1}{2})$$

$$= A \cap B \quad ,$$

as expected. More generally if $B = (b_1\ b_2)$, $b_1 = 0$, $b_2 > 0$, and $a_2 \leq b_2$, then

$$c = \frac{A\,B^T}{B\,B^T} = \frac{a_1 b_1 + a_2 b_2}{b_1^2 + b_2^2} \tag{56}$$

$$= \frac{a_2\,b_2}{b_2^2} \tag{57}$$

$$= \frac{a_2}{b_2} \quad . \tag{58}$$

Then $c\ B = (0\ \frac{a_2 b_2}{b_2}) = (0\ a_2) = A \cap B$ since $a_2 \leq b_2$.

Subsethood has extended the Pythagorean Theorem, relative frequency, and elementhood, and involves the normal equations of least-square estimation. We shall now see how subsethood relates to axiomatic or Bayesian probability and to fuzzy entropy.

Bayesian Polemics

Bayesian probabilists interpret probability as a subjective state of knowledge. In practice they use relative frequencies (subsethood degrees) but only to approximate these "states of knowledge."

Bayesianism is often a polemical doctrine. Some Bayesians claim that they, and only they, use all and only the available uncertainty information in the description of uncertain phenomena. This stems from the Bayes Theorem expansion of the "a posteriori" conditional probability $P(H_i|E)$, the probability that H_i, the ith of k-many disjoint hypotheses $\{H_j\}$, is true given observed evidence E:

$$P(H_i|E) = \frac{P(E \cap H_i)}{P(E)} \tag{59}$$

$$= \frac{P(E|H_i)\ P(H_i)}{P(E)} \tag{60}$$

$$= \frac{P(E|H_i)\ P(H_i)}{\sum_{j=1}^{k} P(E|H_j)\ P(H_j)} \quad , \tag{61}$$

since the hypotheses partition the sample space X : $H_1 \cup H_2 \cup \ldots \cup H_k = X$ and $H_i \cap H_j = \emptyset$ if $i \neq j$.

The Bayesian approach uses all available information in computing the posterior distribution $P(H_i|E)$ by using the "a priori" or prior distribution $P(H_i)$ of the hypotheses. The Bayesian approach stems from the ratio form of the conditional probability measure.

The Subsethood Theorem trivially implies Bayes Theorem when the hypotheses $\{H_i\}$ and evidence E are nonfuzzy subsets. More important, the Subsethood Theorem implies the Fuzzy Bayes Theorem in the more interesting case when the observed data E is fuzzy:

$$S(E, H_i) = \frac{S(H_i, E)\, M(H_i)}{\sum_{j=1}^{k} S(H_j, E)\, M(H_j)} \tag{62}$$

$$= \frac{S(H_i, E)\, f_i}{\sum_{j=1}^{k} S(H_j, E)\, f_j} \quad , \tag{63}$$

where $f_i = \frac{M(H_i)}{M(X)} = \frac{M(H_i)}{n} = S(X, H_i)$ gives the "relative frequency" of H_i, the degree to which all the hypotheses are H_i.

The Subsethood Theorem implies inequality when the partitioning hypotheses are fuzzy. For instance, if $k = 2$, H^c is the complement of an arbitrary fuzzy set H, and evidence E is fuzzy, then [Kosko, 1986b] the occurrence of nondegenerate hypothesis overlap and underlap gives a lower bound on the posterior subsethood:

$$S(E, H) \geq \frac{S(H, E)\, f_H}{S(H, E)\, f_H + S(H^c, E) f_{H^c}} \quad , \tag{64}$$

where $f_H = S(X, H)$. The lower bound increases with $M(H)$ and decreases with $M(H^c)$. Since a like lower bound holds for $S(E, H^c)$, adding the two posterior subsethoods gives the additive inequality

$$S(E, H) + S(E, H^c) \geq 1 \quad , \tag{65}$$

an inequality Zadeh [1983] arrived at independently by directly defining a "relative sigma-count" as the subsethood measure given by the Subsethood Theorem. If H is nonfuzzy,

equality holds as in the additive law of conditional probability:

$$P(H|E) + P(H^c|E) = 1 \quad . \tag{66}$$

The Subsethood Theorem implies a deeper Bayes theorem for arbitrary fuzzy sets, the **Odds-Form Fuzzy Bayes Theorem:**

$$\frac{S(A_1 \cap H, A_2)}{S(A_1 \cap H, A_2^c)} = \frac{S(A_2 \cap H, A_1)}{S(A_2^c \cap H, A_1)} \frac{S(H, A_2)}{S(H, A_2^c)} \quad . \tag{67}$$

We prove this theorem directly by replacing the subsethood terms on the righthand side with their equivalent ratios of counts, canceling like terms three times, multiplying by $\frac{M(A_1 \cap H)}{M(A_1 \cap H)}$, rearranging, and applying the Subsethood Theorem a second time.

We have now developed enough fuzzy theory to examine critically the recent anti-fuzzy polemics of Lindley [1987] and Jaynes [1979] (and thus Cheeseman [1985] who uses Jaynes' arguments). To begin we observe four more corollaries of the Subsethood Theorem:

$$(i) \quad 0 \leq S(H, A) \leq 1, \tag{68}$$

$$(ii) \quad S(H, A) = 1 \text{ if } H \subset A, \tag{69}$$

$$(iii) \quad S(H, A_1 \cup A_2) = S(H, A_1) + S(H, A_2) - S(H, A_1 \cap A_2) \quad , \tag{70}$$

$$(iv) \quad S(H, A_1 \cap A_2) = S(H, A_1)\, S(A_1 \cap H, A_2). \tag{71}$$

Each relationship follows from the ratio form of $S(A, B)$. The third relationship (70) uses the additivity of the count $M(A)$, which follows from $\min(x, y) + \max(x, y) = x + y$.

Suppose we make the notational identification $S(H, A) = P(A|H)$. We then obtain the defining relationships of conditional probability Lindley proposed:

$$Convexity: \quad 0 \leq P(A|H) \leq 1 \text{ and } P(A|H) = 1 \text{ if } H \text{ implies } A, \tag{72}$$

$$Addition: \quad P(A_1 \cup A_2|H) = P(A_1|H) + P(A_2|H) - P(A_1 \cap A_2|H) \tag{73}$$

$$Multiplication: \quad P(A_1 \cap A_2|H) = P(A_1|H)\, P(A_2|A_1 \cap H). \tag{74}$$

"From these three rules," Lindley tells us, "all of the many, rich and wonderful results of the probability calculus follow. They may be described as the axioms of probability." Lindley takes these as "unassailable" axioms: "We really have no choice about the rules governing our measurement of uncertainty: they are dictated to us by the inexorable laws of logic." Lindley proceeds to build a "coherence" argument around the Odds-Form Bayes Theorem, which he correctly deduces from the axioms as the equality

$$\frac{P(A_2|A_1 \cap H)}{P(A_2^c|A_1 \cap H)} = \frac{P(A_1|A_2 \cap H)}{P(A_1|A_2^c \cap H)} \frac{P(A_2|H)}{P(A_2^c|H)}, \tag{75}$$

where here we interpret A^c as not-A. "Any other procedure," Lindley claims, "is incoherent." This polemic evaporates in the face of the above four subsethood corollaries and the Odds-Form Fuzzy Bayes Theorem. Ironically, rather than establish the primacy of axiomatic probability, Lindley seems to argue that it is fuzziness in disguise.

Maximum-entropy estimation provides another source of Bayesian probability polemic [Cheeseman, 1985]. Here the axiomatic argument rests on the so-called Cox's Theorem [1946].

According to physicist E.T. Jaynes [1979]: "Cox proved that any method of inference in which we represent degrees of plausibility by real numbers, is necessarily either equivalent to Laplace's, or inconsistent," where Jaynes cites Laplace as an early Bayesian probabilist. In fact Cox used *bivalent* logic (Boolean algebra) and other assumptions to show that, again according to Jaynes, the "conditions of consistency can be be stated in the form of functional equations," namely the probabilistic product and sum rules:

$$P(A \cap B|C) = P(A|B \cap C)\, P(B|C) \ , \tag{76}$$

$$P(B|A) + P(B^c|A) = 1 \ . \tag{77}$$

The Subsethood Theorem implies

$$S(C, A \cap B) = S(B \cap C, A)\, S(C, B) \ , \tag{78}$$

$$S(A, B) + S(A, B^c) \geq 1 \ , \tag{79}$$

with, as we have seen, equality holding for the second subsethood relationship when B is nonfuzzy, which holds in the Cox-Jaynes setting.

In the probabilistic case overlap and underlap are degenerate. So $P(A \cap A^c|B) = P(\emptyset|B) = \frac{P(\emptyset)}{P(B)} = 0$, and $P(B|A \cap A^c) = P(B|\emptyset)$ is undefined. Yet in general $S(B, A \cap A^c) > 0$, and we can define $S(A \cap A^c, B)$ when A and B are fuzzy or nonfuzzy.

Jaynes' claim is either false or concedes that probability is a special case of fuzziness. For strictly speaking, since the subsethood measure $S(A, B)$ satisfies the multiplicative and additive laws specified by Cox and yet differs from the conditional probability $P(B|A)$, Jaynes' claim is false.

Presumably Jaynes was unaware of fuzzy sets. He suggests that the frequency theory of probability provides the only alternative uncertainty theory, and we have reduced relative frequency to the subsethood measure $S(X, A)$. So if we restrict consideration to nonfuzzy sets A and B, equality holds in the above subsethood relations, and Jaynes argues correctly: probability and fuzziness coincide. But fuzziness exists, indeed abounds, outside this restriction and classical probability theory does not. So fuzzy theory extends probability theory. Equivalently, probability represents a special case of fuzziness.

When we examine Cox's actual arguments, we find that Cox assumes that the uncertainty combination operators are continuously *twice differentiable*. Min and max are not twice differentiable. Technically, Cox's theorem does not apply.

THE ENTROPY-SUBSETHOOD THEOREM

We independently derived the Fuzzy Entropy Theorem and the Subsethood Theorem from first principles, from sets-as-points unit-cube geometry. Both theorems involve ratios of cardinalities. So we can suspect a connection.

The Entropy-Subsethood Theorem shows that the connection involves overlap $A \cap A^c$ and underlap $A \cup A^c$. The theorem eliminates fuzzy entropy in favor of subsethood. So subsethood emerges as the fundamental, characterizing quantity of fuzziness—and, arguably, of probability as well.

Entropy-Subsethood Theorem: $$E(A) = S(A \cup A^c, A \cap A^c) \; . \qquad (80)$$

The theorem follows if we replace B and A in the Subsethood Theorem with respectively overlap $A \cap A^c$ and underlap $A \cup A^c$. Since overlap is a subset of underlap, since $S(A \cap A^c, A \cup A^c) = 1$, the intersection of the two sets equals the overlap.

The Entropy-Subsethood Theorem describes a peculiar relationship. It gives fuzziness or ambiguity as the degree to which the superset $A \cup A^c$ is a subset of its own subset $A \cap A^c$, the extent to which the whole is a part of one of its own parts, a relationship Western logic forbids.

This relationship violates our ingrained Venn-diagram intuitions of unambiguous set inclusion. Only the midpoint of I^n yields total containment of underlap in overlap. The

cube vertices yield zero containment. This parallels in the extreme the relative frequency relationship $S(X, A) = \frac{n_A}{N}$, where nonfuzzy subset A contains to some degree its nonfuzzy superset X.

Figure 16.11 illustrates the Entropy-Subsethood Theorem. It shows that d^*, the shortest distance from underlap $A \cup A^c$ to the hyper-rectangle that defines the fuzzy power set of overlap $A \cap A^c$, equals the distance $d(A \cup A^c, A \cap A^c) = d(A, A^c)$ and equals a difference of vector magnitudes: $d^* = M(A \cup A^c) - M(A \cap A^c)$.

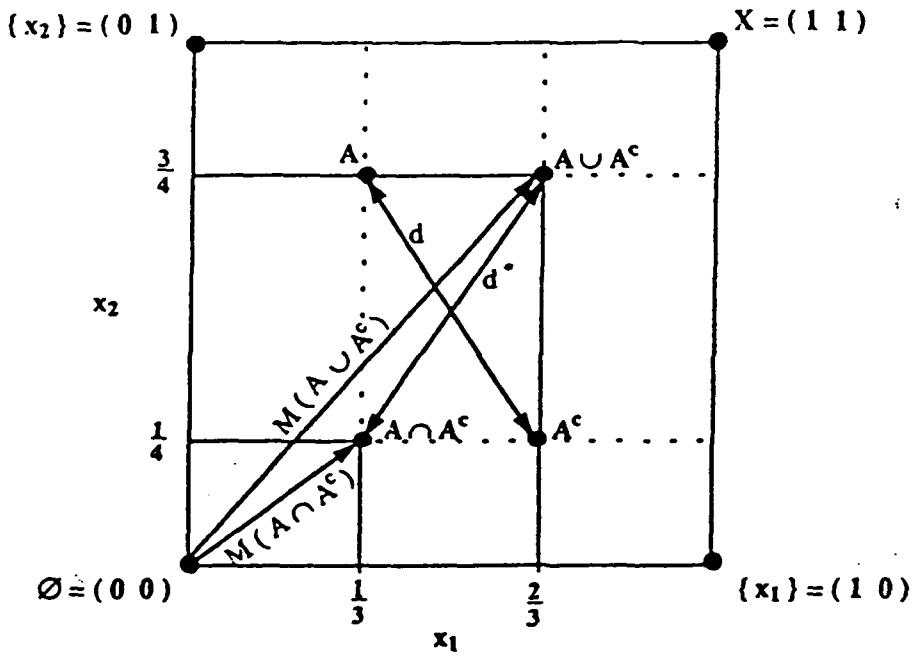


Figure 16.11 Entropy-Subsethood Theorem in two dimensions. Just as the long diagonals have equal length, $d(A, A^c) = d(A \cup A^c, A \cap A^c) = d^* = M(A \cup A^c) - M(A \cap A^c)$, the shortest distance from $A \cup A^c$ to the fuzzy power set of $A \cap A^c$.

The Entropy-Subsethood Theorem implies that no probability measure measures fuzziness. For the moment, suppose not. Suppose fuzzy entropy measures nothing new; fuzziness is simply disguised probability. Suppose, as Lindley [1987] claims, that probability

theory "is adequate for all problems involving uncertainty." Then there exists some probability measure P such that $P = E$. P cannot equal zero everywhere because $P(X) = 1$. Then there is some A such that $P(A) = E(A) > 0$. But in a probability space overlap or underlap are degenerate: $A \cap A^c = \emptyset$, and $A \cup A^c = X$.

The Entropy-Subsethood Theorem then implies that $0 < P(A) = E(A) = S(A \cup A^c, A \cap A^c) = S(X, \emptyset)$. X can be a subset to nonzero degree of the empty set only if X itself is empty, and hence only if A is empty: $X = A = \emptyset$. Then the sure event X is impossible: $P(X) = P(\emptyset) = 0$. Or the impossible event is sure: $P(\emptyset) = 1$. Either outcome gives a bivalent contradiction, impervious to normalization. So there exists no probability measure P that measures fuzziness. Fuzziness exists.

REFERENCES

Bandler, W., Kohout, L., "Fuzzy Power Sets and Fuzzy Implication Operators," *Fuzzy Sets and Systems*, vol. 4, 13 - 30, 1980.

Black, M., "Vagueness: An Exercise in Logical Analysis," *Philosophy of Science*, vol. 4, 427-455, 1937.

Cheeseman, P., "In Defense of Probability," *Proc. of the IJCAI-85*, 1002 - 1009, Aug. 1985.

Chung, K.L., *A Course in Probability Theory*, Academic Press, 1974.

Cox, R. T., "Probability, Frequency, and Reasonable Expectations," *American Journal of Physics*, vol. 14, no. 1, 1 - 13, Jan./Feb. 1946.

Gaines, B. R., "Precise Past, Fuzzy Future," *International Journal of Man-Machine Studies*, vol. 19, 117 - 134, 1983.

Hume, D., *An Inquiry Concerning Human Understanding*, 1748.

Jaynes, E.T., "Where do we Stand on Maximum Entropy?" in *The Maximum Entropy Formalism*, Levine & Tribus, Eds., MIT Press, 1979.

Kac, M., *Probability and Related Topics in Physical Sciences, Lectures in Applied Mathematics*, vol. I, Interscience: New York, 1959.

Kant, I., *Critique of Pure Reason*, Second Edition, 1787.

Kline, M., *Mathematics: The Loss of Certainty*, Oxford University Press, 1980.

Klir, G.J., Folger, T.A., *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall: New Jersey, 1988.

Kosko, B., "Counting With Fuzzy Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, 556 - 557, July 1986.

Kosko, B., "Fuzzy Entropy and Conditioning," *Information Sciences*, vol. 40, 165 - 174, 1986.

Kosko, B., *Foundations of Fuzzy Estimation Theory*, Ph.D. dissertation, June 1987, Department of Electrical Engineering, University of California at Irvine, Order Number 8801936, University Microfilms International, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

Kosko, B., "Fuzzy Quantum States," in preparation, 1991.

Lindley, D.V., "The Probability Approach to the Treatment of Uncertainty in Artificial Intelligence and Expert Systems," *Statistical Science*, vol. 2, no. 1, 17 - 24, Feb. 1987.

Mill, J.S., *A System of Logic*, 1843.

Prade, H. "A Computational Approach to Approximate and Plausible Reasoning with Applications to Expert Systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, 260 - 283, 1985.

Quine, W.V.O., "What Price Bivalence?" *Journal of Philosophy*, vol. 78, no. 2, 90-95, February 1981.

Quine, W.V.O., *Quiddities*, Harvard University Press, 1987.

Graw-Hill, 1969.

s:

on, New York: Marrel Dekker, 1980.

Philosophicus, Routledge & Kegan Paul Ltd., London,

tion and Control, vol. 8, 338 - 353, 1965.

pproach to Fuzzy Quantifiers in Natural Languages",
), no. 1, 149 - 184, 1983.

7. Prove the fuzzy De Morgan Laws:

   (a) $A \cap B = (A^c \cup B^c)^c$

   (b) $A \cup B = (A^c \cap B^c)^c$

8. Prove:

$$0 \leq \ell^p(A, A_{near}) \leq \frac{n^{1/p}}{2} \leq \ell^p(A, A_{far}) \leq n^{1/p}, \qquad p \geq 1.$$

9. Prove the ℓ^1-version of the Fuzzy Entropy Theorem:

$$E(A) = \frac{\ell^1(A, A_{near})}{\ell^1(A, A_{far})} = \frac{M(A \cap A^c)}{M(A \cup A^c)}$$

10. Prove: $M(A) + M(B) = M(A \cap B) + M(A \cup B)$.

11. Prove: $\frac{1}{n} M(A \cap A^c) + \frac{1}{n} M(A \cup A^c) = 1$.

12. Prove:

   (a) $E(P) = \frac{1}{n-1}$ if $M(P) = 1$ and all $p_i \leq 1/2$,

   (b) $E(P) < \frac{1}{n-1}$ if $M(P) = 1$ and some $p_j > 1/2$.

13. Prove the fit-violation version of the Subsethood Thoerem:

$$S(A.B) = 1 - \frac{\sum_x \max(0, m_A - m_B(x))}{M(A)} = \frac{M(A \cap B)}{M(A)}.$$

14. Prove:

$$S(E, H_i) = \frac{S(H_i, E)\, f_i}{\sum_{j=1}^{K} S(H_j, E)\, f_j},$$

   where $f_i = S(X, H_i)$, the nonfuzzy sets $H_1, \ldots, H_K$ partition X, and E is fuzzy.

15. Prove:

$$S(E, H) \geq \frac{S(H, E) f_H}{S(H, E) f_H + S(H^c, E) f_{H^c}},$$

where $f_H = S(X, H)$ and E and H are arbitrary fuzzy sets.

16. Prove the Odds-Form Bayes Theorem:

$$\frac{S(A_1 \cap H, A_2)}{S(A_1 \cap H, A_2^c)} = \frac{S(A_2 \cap H, A_1)}{S(A_2^c \cap H, A_1)} \frac{S(H, A_2)}{S(H, A_2^c)},$$

for arbitrary fuzzy sets A_1, A_2, and H.

17. Prove directly the additive inequality: $S(A, B) + S(A, B^c) \geq 1$.

18. Prove:

(a) $0 \leq S(H, A) \leq 1$,

(b) $S(H, A) = 1$ if $H \subset A$,

(c) $S(H, A_1 \cup A_2) = S(H, A_1) + S(H, A_2) - S(H, A_1 \cap A_2)$,

(d) $S(H, A_1 \cap A_2) = S(H, A_1)\, S(A_1 \cap H, A_2)$.

19. Show that $N_a(N_a(x)) = x$ for the generalized negation operator

$$N_a(x) = \frac{1 - x}{1 + ax}, \qquad a > -1, \ 0 \leq x \leq 1,$$

20. If we define intersection $\cap_T$ pointwise by

$$T(x, y) = 1 - \min(1, [(1 - x)^p + (1 - y)^p]^{1/p}), \qquad p > 0,$$

how should we define the corresponding De Morgan dual union $\cup_S$?

21. What De Morgan dual union operator corresponds to the intersection operator

$$\max(0, x + y - 1)\,?$$

22. Zadeh's consequent conjunction syllogism schematizes as

Q_1 As are Bs

Q_2 As are Cs

Therefore: Q As are Bs and Cs

Show that if $Q_1 = S(A, B)$ and $Q_2 = S(A, C)$, then the fuzzy quantifier Q obeys

$$\max(0, Q_1 + Q_2 - 1) \leq Q \leq \min(Q_1, Q_2).$$

23. Define the *volume subsethood* measure $V(A, B)$ as

$$V(A, B) = \frac{v(A \cap B)}{v(A)} \quad ,$$

for fit vectors $A = (a_1, \ldots, a_n)$ and $B = (b_1, \ldots, b_n)$ such that $a_i > 0$. $v(A)$ is the Lebesgue or volume measure of A:

$$v(A) = \prod_{i=1}^{n} a_i \quad .$$

The volume subsethood measure $V(A, B)$ measures the ratio of the volume of the overlap hyper-rectangle $F(2^{A \cap B})$ to the volume of A's fuzzy power set $F(2^A)$. Prove that the volume subsethood measure $V(A, B)$ underestimates the subsethood measure $S(A, B)$:

$$V(A, B) \leq S(A, B) \quad .$$

In Review: Oct. 1990

ADAPTIVE FUZZY SYSTEMS FOR BACKING UP A TRUCK-AND-TRAILER

Seong-Gon Kong and Bart Kosko
Department of Electrical Engineering
Signal and Image Processing Institute
University of Southern California
Los Angeles, California 90089-0272

Abstract

We developed fuzzy and neural-network control systems to back up a simulated truck, and truck-and-trailer, to a loading dock in a planar parking lot. The fuzzy systems performed well until we randomly removed over 50 % of their fuzzy-associative-memory (FAM) rules. They also performed well when we replaced key FAM equilibration rules with destructive or "sabotage" rules. We trained the neural network systems with the supervised backpropagation learning algorithm and tested their robustness by removing random subsets of training data in learning sequences. The neural systems performed well but required extensive computation for training. We used unsupervised differential competitive learning (DCL), and product-space clustering, to adaptively generate FAM rules from training data. The original fuzzy and neural control systems generated trajectory data. The DCL system rapidly recovered the underlying FAM rules. Product-space clustering converted the neural truck systems into structured sets of FAM rules that approximated the neural system's behavior.

Fuzzy and Neural Control Systems

We construct fuzzy and neural control systems directly from control data, but from different types of control data. Fuzzy systems use a small number of structured *linguistic* input-output samples from an expert or from some other adaptive estimator. Neural systems use a large number of *numeric* input-output samples from the control process or from some other database. Adaptive fuzzy systems also use numeric control data.

Figure 1 illustrates this difference. The neural system estimates function $f : X \longrightarrow Y$ from several numerical *point* samples (x_i, y_i). The fuzzy system estimates f from a few fuzzy *set* samples or fuzzy associations (A_i, B_i).

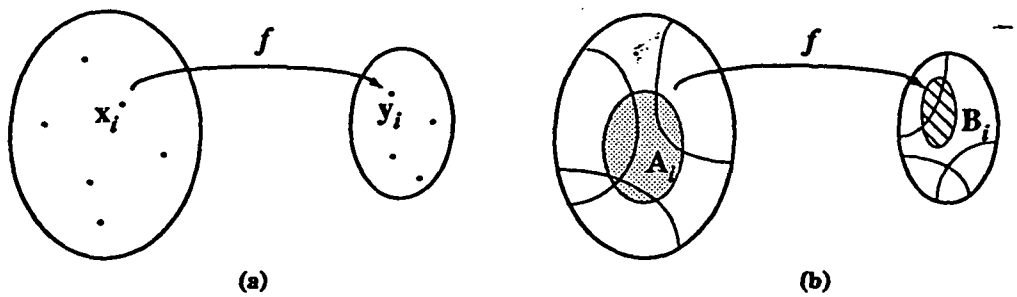


FIGURE 1 Geometry of neural and fuzzy function estimation. The neural approach (a) uses several numerical point samples. The fuzzy approach (b) uses a few fuzzy set samples.

Fuzzy and neural systems offer a key advantage over traditional control approaches. They offer *model-free estimation* of the control system. The user need not specify how the controller's output mathematically depends on its input. Instead the user provides a few common-sense associations of how the control variables behave. Or the user provides a statistically representative set of numerical training samples. Even if a math-model controller is available, fuzzy or neural controllers may prove more robust and easier to modify.

Which system, fuzzy or neural, performs better for which type of control problem de-

pends on the type and availability of sample data. If experts provide structured knowledge of the control process, or if sufficient numerical training samples are unavailable, the fuzzy approach may be preferable. We can construct a fuzzy control system with comparative ease when experts or fuzzy engineers provide accurate structured knowledge. A fuzzy control system seems a reasonable benchmark in such cases, even if we can develop a neural controller or math-model controller.

If we have representative numerical data but not structured expertise, the neural approach may be preferable. Or a statistical regression approach may be more appropriate. The data simply tell their own story—if there is a story to tell. Yet even here we can use a hybrid fuzzy-neural system, an adaptive fuzzy system. We can use the numerical data to generate *fuzzy associative memory* (FAM) rules. The FAM rules can then form the skeleton of a fuzzy control *architecture*. In short, if structured knowledge is unavailable, estimate it. This may be more practical than it would appear because of the small number of control FAM rules needed to reliably control many realworld processes.

How can we compare fuzzy and neural controllers? Abstract comparison proves difficult because both approaches build a control black box in different ways. That they build black boxes distinguishes them from math-model controllers. It also suggests we can compare them, at least approximately, by their black-box control performance.

Each control system generated an output *control surface* as it ranged over the common input space of parameter values. Figure 5 below shows three-dimensional control surfaces for the fuzzy and neural controllers. For control systems with few input parameters with moderately quantized ranges, we can store both fuzzy and neural controllers—or rather their quantized control surfaces—as decision look-up tables. Then once we specify a system performance criterion, we can in principle quantitatively compare the controllers.

Comparing system trajectories proved more complicated. In the case at hand, we wanted to back up a truck, and truck-and-trailer, to a loading dock. We can measure and compare the quality and quantity of the truck trajectory, perhaps with mean-squared error criteria. Intuitively, we preferred smooth short trajectories to jagged long trajectories. Reaching the loading-dock goal was also important. In practice it is the most important performance requirement. We must balance the trajectory type with the trajectory

destination, and this reduces to the pragmatic issue of balancing means and ends.

Below we develop a simple fuzzy control system and a simple neural control system for backing up a truck, and truck-and-trailer, in an open parking lot. The recent neural network truck backer-upper simulation of Nguyen and Widrow [1989] motivated our choice of control problem.

The fuzzy control system compared favorably with the neural controller in terms of black-box development effort, black-box computational load, smoothness of truck trajectories, and robustness.

We studied robustness of the fuzzy control systems in two ways. We deliberately added confusing FAM rules—"sabotage" rules—to the system, and we randomly removed different subsets of FAM rules. We studied robustness of the neural controller by randomly removing different portions of the training data in learning sequences. We also converted the neural control systems to structured FAM-bank systems.

Backing up a truck

Figure 2 shows the simulated truck and loading zone. The truck corresponds to the cab part of the neural truck in the Nguyen-Widrow neural truck backer-upper system. The three state variables ϕ, x, and y exactly determine the truck position. ϕ specifies the angle of the truck with the horizontal. The coordinate pair (x, y) specifies the position of the rear center of the truck in the plane.

The goal was to make the truck arrive at the loading dock at a right angle ($\phi_f = 90°$) and to align the position (x, y) of the truck with the desired loading dock (x_f, y_f). We considered only backing up. The truck moved backward by some fixed distance at every stage. The loading zone corresponded to the plane $[0, 100] \times [0, 100]$, and (x_f, y_f) equaled $(50, 100)$.

At every stage the fuzzy and neural controllers should produce the steering angle θ that backs up the truck to the loading dock from any initial position and from any angle in the loading zone.

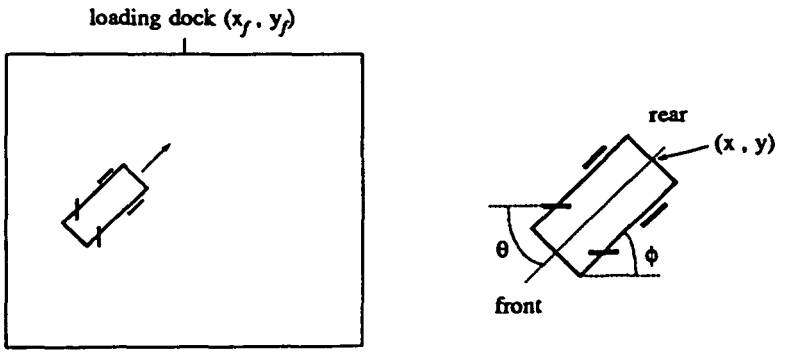


FIGURE 2 Diagram of simulated truck and loading zone.

Fuzzy Truck Backer-Upper System

We first specified each controller's input and output variables. The input variables were the truck angle ϕ and the x-position coordinate x. The output variable was the steering-angle signal θ. We assumed enough clearance between the truck and the loading dock so we could ignore the y-position coordinate. The variable ranges were as follows:

$$0 \leq x \leq 100 \ ,$$
$$-90 \leq \phi \leq 270 \ ,$$
$$-30 \leq \theta \leq 30 \ .$$

Positive values of θ represented clockwise rotations of the steering wheel. Negative values represented counterclockwise rotations. We discretized all values to reduce computation. The resolution of ϕ and θ was one degree each. The resolution of x was 0.1.

Next we specified the fuzzy-set values of the input and output fuzzy variables. The fuzzy sets numerically represented linguistic terms, the sort of linguistic terms an expert might use to describe the control system's behavior. We chose the fuzzy-set values of the fuzzy variables as follows:

Angle ϕ		x-position x		Steering-angle signal θ	
RB:	Right Below	LE:	Left	NB:	Negative Big
RU:	Right Upper	LC:	Left Center	NM:	Negative Medium
RV:	Right Vertical	CE:	Center	NS:	Negative Small
VE:	Vertical	RC:	Right Center	ZE:	Zero
LV:	Left Vertical	RI:	Right	PS:	Positive Small
LU:	Left Upper			PM:	Positive Medium
LB:	Left Below			PB:	Positive Big

Fuzzy subsets contain elements with degrees of membership. A fuzzy membership function $m_A : Z \longrightarrow [0,1]$ assigns a real number between 0 and 1 to every element z in the universe of discourse Z. This number $m_A(z)$ indicates the degree to which the object or data z belongs to the fuzzy set A. Equivalently, $m_A(z)$ defines the *fit* (fuzzy unit) value [Kosko, 1986] of element z in A.

Fuzzy membership functions can have different shapes depending on the designer's preference or experience. In practice fuzzy engineers have found triangular and trapezoidal shapes help capture the modeler's sense of fuzzy numbers and simplify computation. Figure 3 shows membership-function graphs of the fuzzy subsets above. In the third graph, for example, $\theta = 20°$ is Positive Medium to degree 0.5, but only Positive Big to degree 0.3.

In Figure 3 the fuzzy sets CE, VE, and ZE are narrower than the other fuzzy sets. These narrow fuzzy sets permit fine control near the loading dock. We used wider fuzzy sets to describe the endpoints of the range of the fuzzy variables ϕ, x, and θ. The wider fuzzy sets permitted rough control far from the loading dock.

Next we specified the fuzzy "rulebase" or bank of *fuzzy associative memory* (FAM) rules. Fuzzy associations or "rules" (A, B) associate output fuzzy sets B of control values with input fuzzy sets A of input-variable values. We can write fuzzy associations as antecedent-consequent pairs or IF-THEN statements.

In the truck backer-upper case, the FAM bank contained the 35 FAM rules in Figure 4. For example, the FAM rule of the left upper block (FAM rule 1) corresponds to the following

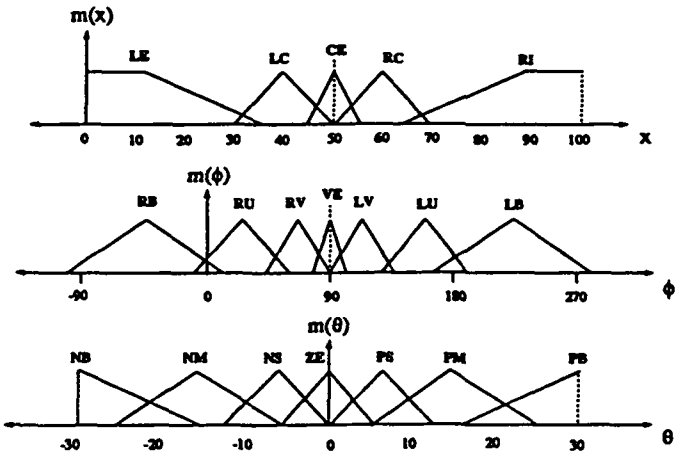


FIGURE 3 Fuzzy membership functions for each linguistic fuzzy-set value. To allow finer control, the fuzzy sets that correspond to near the loading dock are narrower than the fuzzy sets that correspond to far from the loading dock.

fuzzy association:

$$IF\ x = LE\ AND\ \phi = RB, \qquad THEN\ \theta = PS.$$

FAM rule 18 indicates that if the truck is in near the equilibrium position, then the controller should not produce a positive or negative steering-angle signal. The FAM rules in the FAM-bank matrix reflect the symmetry of the controlled system.

For the initial condition $x = 50$ and $\phi = 270$, the fuzzy truck did not perform well. The symmetry of the FAM rules and the fuzzy sets cancelled the fuzzy controller output in a rare saddle point. For this initial condition, the neural controller (and truck-and-trailer below) also performed poorly. Any perturbation breaks the symmetry. For example, the rule (If $x = 50$ and $\phi = 270$, then $\theta = 5$) corrected the problem.

The three-dimensional control surfaces in Figure 5 show steering-angle signal outputs θ that correspond to all combinations of values of the two input state variables ϕ and x. The control surface defines the fuzzy controller. In this simulation the correlation-minimum FAM inference procedure, discussed in [Kosko, 1990a], determined the fuzzy control surface. If the control surface changes with sampled variable values, the system

		x				
		LE	LC	CE	RC	RI
ϕ	RB	[1] PS	[2] PM	[3] PM	[4] PB	[5] PB
	RU	[6] NS	[7] PS	PM	PB	PB
	RV	NM	NS	PS	PM	PB
	VE	NM	NM	[18] ZE	PM	PM
	LV	NB	NM	NS	PS	PM
	LU	NB	NB	NM	NS	PS
	LB	NB	NB	NM	NM	[35] NS

FIGURE 4 FAM-bank matrix for the fuzzy truck backer-upper controller.

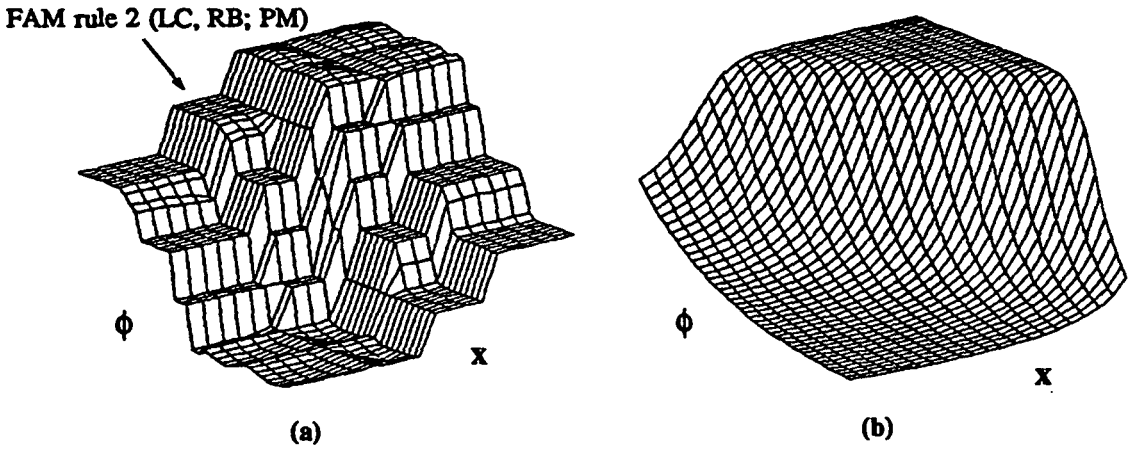


FIGURE 5 (a) Control surface of the fuzzy controller. Fuzzy-set values determined the input and output combination corresponding to FAM rule 2 (IF x=LC AND ϕ=RB, THEN θ=PM). (b) Corresponding control surface of the neural controller for constant value y=20.

behaves as an *adaptive* fuzzy controller. Below we demonstrate unsupervised adaptive control of the truck and the truck-and-trailer systems.

Finally, we determined the output action given the input conditions. We used the correlation-minimum inference method illustrated in Figure 6. Each FAM rule produced the output fuzzy set clipped at the degree of membership determined by the input conditions and the FAM rule. Alternatively, correlation-product inference [Kosko, 1990a] would combine FAM rules multiplicatively. Each FAM rule emitted a fit-weighted output fuzzy set O_i at each iteration. The total output O added these weighted outputs:

$$O = \sum_i O_i \tag{1}$$

$$= \sum_i \min(f_i, S_i) \quad , \tag{2}$$

where f_i denotes the antecedent fit value and S_i represents the consequent fuzzy set of steering-angle values in the ith FAM rule. Earlier fuzzy systems combined the output sets O_i with pairwise maxima. But this tends to produce a uniform output set O as the number of FAM rules increases. Adding the output sets O_i invokes the fuzzy version of the Central Limit Theorem. This tends to produce a symmetric, unimodal output fuzzy set O of steering-angle values.

Fuzzy systems map fuzzy sets to fuzzy sets. The fuzzy control system's output defines the fuzzy set O of steering-angle values at each iteration. We must "defuzzify" the fuzzy set O to produce a numerical (point-estimate) steering-angle output value θ.

As discussed in [Kosko, 1990a], the simplest defuzzification scheme selects the value corresponding to the *maximum fit* value in the fuzzy set. This mode-selection approach ignores most of the information in the output fuzzy set and requires an additional decision algorithm when multiple modes occur.

Centroid defuzzification provides a more effective procedure. This method uses the *fuzzy centroid* $\bar{\theta}$ as output:

$$\bar{\theta} = \frac{\sum_{j=1}^{p} \theta_j \, m_O(\theta_j)}{\sum_{j=1}^{p} m_O(\theta_j)} \quad , \tag{3}$$

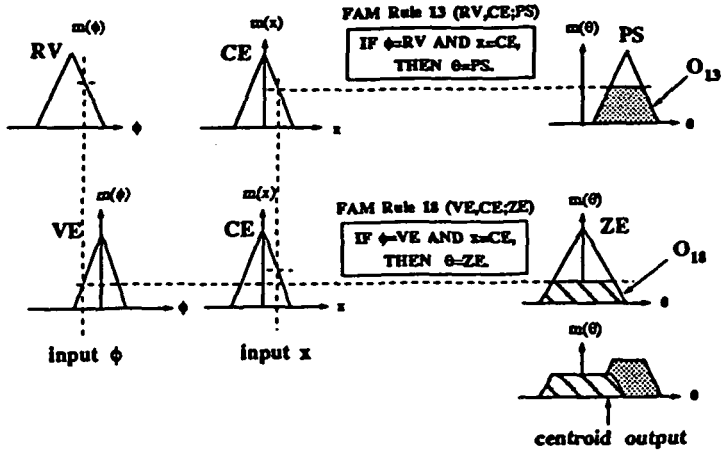


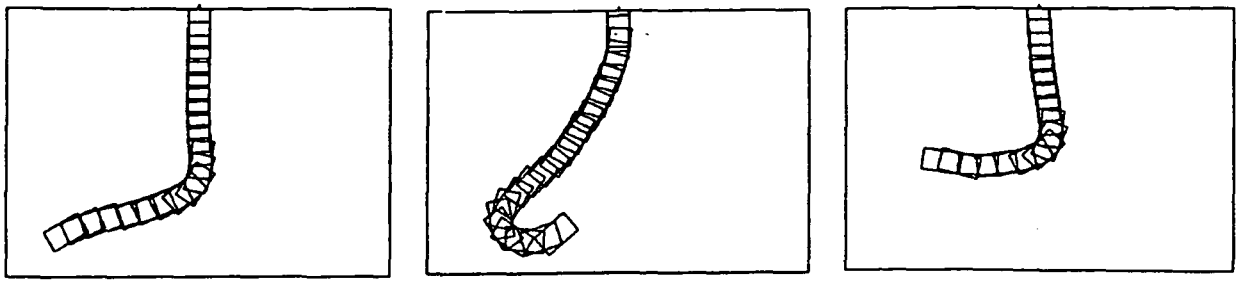
FIGURE 6 Correlation-minimum inference with centroid defuzzification method. Then FAM-rule antecedents combined with AND use the *minimum* fit value to activate consequents. Those combined with OR would use the *maximum* fit value.

where O defines a fuzzy subset of the steering-angle universe of discourse $\Theta = \{\theta_1, \ldots, \theta_p\}$. The central-limit-theorem effect produced by adding output fuzzy set O_i benefits both max-mode and centroid defuzzification. Figure 6 shows the correlation-minimum inference and centroid defuzzification applied to FAM rules 13 and 18. We used centroid defuzzification in all simulations.

With 35 FAM rules, the fuzzy truck controller produced successful truck backing-up trajectories starting from any initial position. Figure 7 shows typical examples of the fuzzy-controlled truck trajectories from different initial positions. The fuzzy control system did not use ("fire") all FAM rules at each iteration. Equivalently most output consequent sets are empty. In most cases the system used only one or two FAM rules at each iteration. The system used at most 4 FAM rules at once.

Neural Truck Backer-Upper System

The neural truck backer-upper of Nguyen and Widrow [1989] consisted of multilayer



(a) (b) (c)

FIGURE 7 Sample truck trajectories of the fuzzy controller for initial positions (x, y, ϕ): (a) (20,20,30), (b) (30,10,220), and (c) (30,40,−10).

feedforward neural networks trained with the backpropagation gradient-descent (stochastic-approximation) algorithm. The *neural control system* consisted of two neural networks: the controller network and the truck emulator network. The *controller network* produced an appropriate steering-angle signal output given any parking-lot coordinates (x, y), and the angle ϕ. The *emulator network* computed the next position of the truck. The emulator network took as input the previous truck position and the current steering-angle output computed by the controller network.

We did not train the emulator network since we could not obtain "universal" synaptic connection weights for the truck emulator network. The backpropagation learning algorithm did not converge for some sets of training samples. The number of training samples for the emulator network might exceed 3000. For example, the combinations of training samples of a given angle ϕ, x-position, y-position, and steering angle signal θ might correspond to 3150 ($18 \times 5 \times 5 \times 7$) samples depending on the division of the input-output product space. Moreover, the training samples were numerically similar since the neuronal signals assumed scaled values in $[0, 1]$ or $[-1, 1]$. For example, we treated close values, such as 0.40 and 0.41, as distinct sample values.

Simple kinematic equations replaced the truck emulator network. If the truck moved

backward from (x, y) to (x', y') at an iteration, then

$$x' = x + r\cos(\phi') \ , \tag{4}$$

$$y' = y + r\sin(\phi') \ , \tag{5}$$

$$\phi' = \phi + \theta \ . \tag{6}$$

r denotes the fixed driving distance of the truck for all backing movements. We used equations (4)–(6) instead of the emulator network. This did not affect the post-training performance of the neural truck backer-upper since the truck emulator network back-propagated only errors.

We trained only the controller network with backpropagation. The controller network used 24 "hidden" neurons with logistic sigmoid functions. In the training of the truck-controller, we estimated the ideal steering-angle signal at each stage before we trained the controller network. In the simulation, we used the arc-shaped truck trajectory produced by the fuzzy controller as the ideal trajectory. The fuzzy controller generated each training sample (x, y, ϕ, θ) at each iteration of the backing-up process. We used 35 training sample vectors and needed more than 100,000 iterations to train the controller network.

Figure 5b shows the resulting neural control surface for $y = 20$. The neural control surface shows less structure than the corresponding fuzzy control surface. This reflects the unstructured nature of black-box supervised learning. Figure 8 shows the network connection topology for our neural truck backer-upper control system.

Figure 9 shows typical examples of the neural-controlled truck trajectories from several initial positions. Even though we trained the neural network to follow the smooth arc-shaped path, some learned truck trajectories were non-optimal.

Comparison of Fuzzy and Neural Systems

As shown in Figure 7 and 9, the fuzzy controller always smoothly backed up the truck but the neural controller did not. The neural-controlled truck sometimes followed an irregular path.

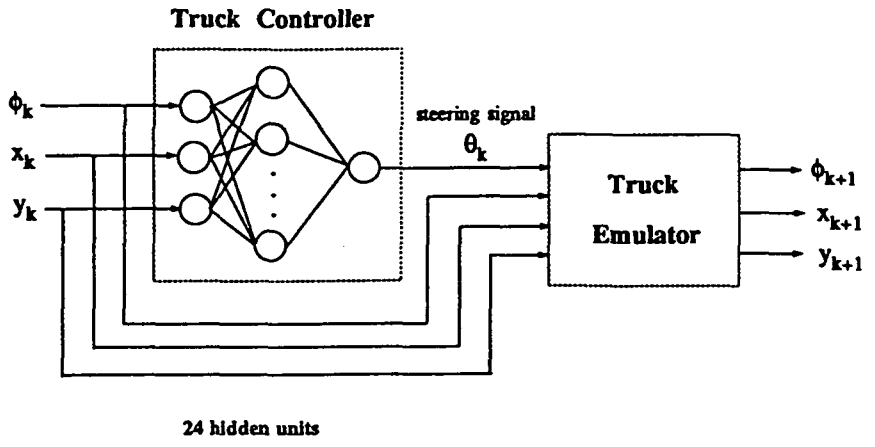


FIGURE 8 Topology of our neural control system.

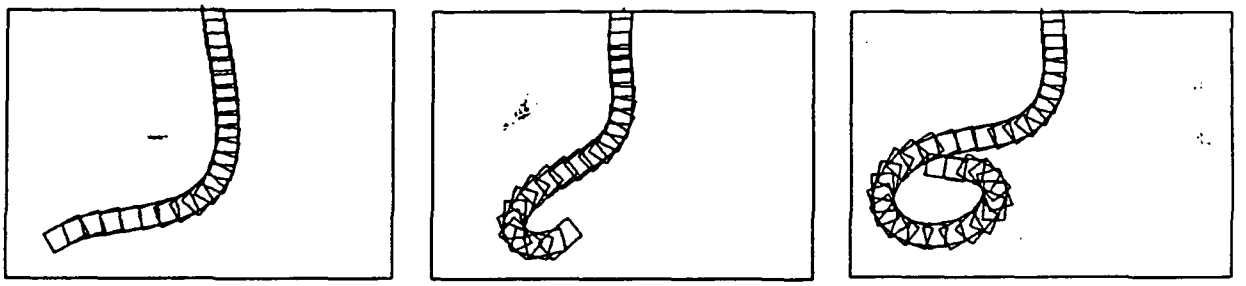
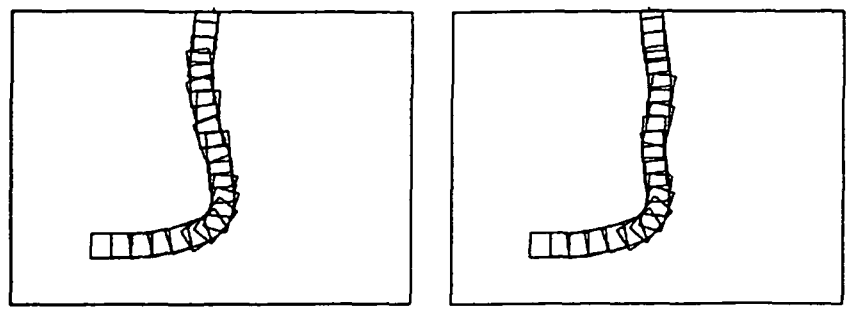


FIGURE 9 Sample truck trajectories of the neural controller for initial positions (x, y, ϕ): (a) (20,20,30), (b) (30,10,220), and (c) (30,40,-10).



(a) (b)

FIGURE 10 The fuzzy truck trajectory after we replaced the key steady-state FAM rule 18 by the two worst rules: (a) IF $x = CE$ AND $\phi = VE$, THEN $\theta = PB$, and (b) IF $x = CE$ AND $\phi = VE$, THEN $\theta = NB$.

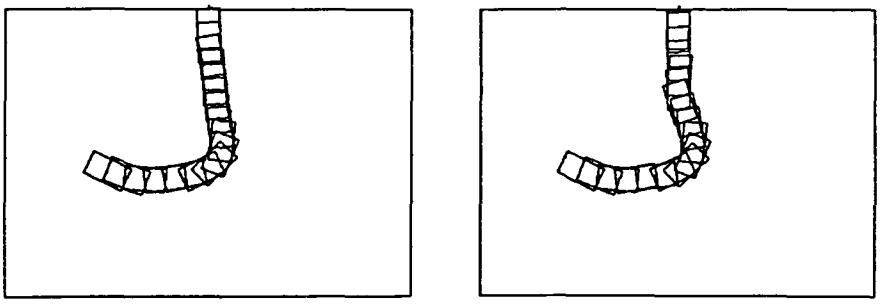
Training the neural control system was time-consuming. The backpropagation algorithm required thousands of back-ups to train the controller network. In some cases, the learning algorithm did not converge.

We "trained" the fuzzy controller by encoding our own common sense FAM rules. Once we develop the FAM-rule bank, we can compute control outputs from the resulting FAM-bank matrix or control surface. The fuzzy controller did not need a truck emulator and did not require a math model of how outputs depended on inputs.

The fuzzy controller was computationally lighter than the neural controller. Most computation operations in the neural controller involved the multiplication, addition, or logarithm of two real numbers. In the fuzzy controller, most computational operations involved comparing and adding two real numbers.

Sensitivity Analysis

We studied the sensitivity of the fuzzy controller in two ways. We replaced the FAM rules with destructive or "sabotage" FAM rules, and we randomly removed FAM rules.



(a) (b)

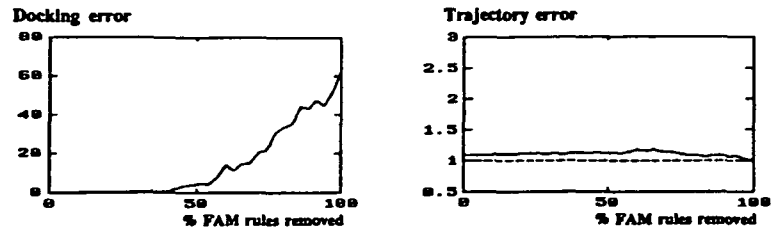
FIGURE 11 Fuzzy truck trajectory when (a) no FAM rules are removed and (b) FAM rules 7, 13, 18 and 23 are removed.

We deliberately chose sabotage FAM rules to confound the system. Figure 10 shows the trajectory when two sabotage FAM rules replaced the important steady-state FAM rule—FAM rule 18: the fuzzy controller should produce zero output when the truck is nearly in the correct parking position. Figure 11 shows the truck trajectory after we removed four randomly chosen FAM rules (7, 13, 18, and 23). These perturbations did not significantly affect the fuzzy controller's performance.

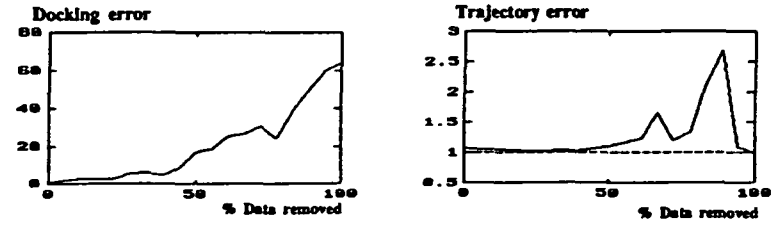
We studied robustness of each controller by examining failure rates. For the fuzzy controller we removed fixed percentages of randomly selected FAM rules from the system. For the neural controller we removed training data. Figure 12 shows performance errors averaged over ten typical back-ups with missing FAM rules for the fuzzy controller and missing training data for the neural controller. The missing FAM rules and training data ranged from 0 % to 100 % of the total. In Figure 12a, the docking error equaled the Euclidean distance from the actual final position (ϕ, x, y) to the desired final position (ϕ_f, x_f, y_f):

$$\text{Docking Error} = \sqrt{(\phi_f - \phi)^2 + (x_f - x)^2 + (y_f - y)^2} \quad . \tag{7}$$

In Figure 12b, the trajectory error equaled the ratio of the actual trajectory length of the



(a) Fuzzy controller



(b) BP-Neural controller

FIGURE 12 Comparison of robustness of the controllers: (a) Docking and Trajectory error of the fuzzy controller, (b) Docking and Trajectory error of the neural controller.

truck divided by the straight line distance to the loading dock:

$$\text{Trajectory Error} = \frac{\text{length of truck trajectory}}{\text{distance(initial position, desired final position)}}. \tag{8}$$

Adaptive Fuzzy Truck Backer-Upper

Adaptive FAM (AFAM) systems generate FAM rules directly from training data. A one-dimensional FAM system, $S : I^n \longrightarrow I^p$, defines a FAM rule, a single association of the form (A_i, B_i). In this case the input-output product space equals $I^n \times I^p$. As discussed in [Kosko, 1990a], a FAM rule (A_i, B_i) defines a cluster or ball of points in the product-space cube $I^n \times I^p$ centered at the point (A_i, B_i). Adaptive clustering algorithms can estimate the

unknown FAM rule (A_i, B_i) from training samples in R^2. We used differential competitive learning (DCL) to recover the bank of FAM rules that generated the truck training data.

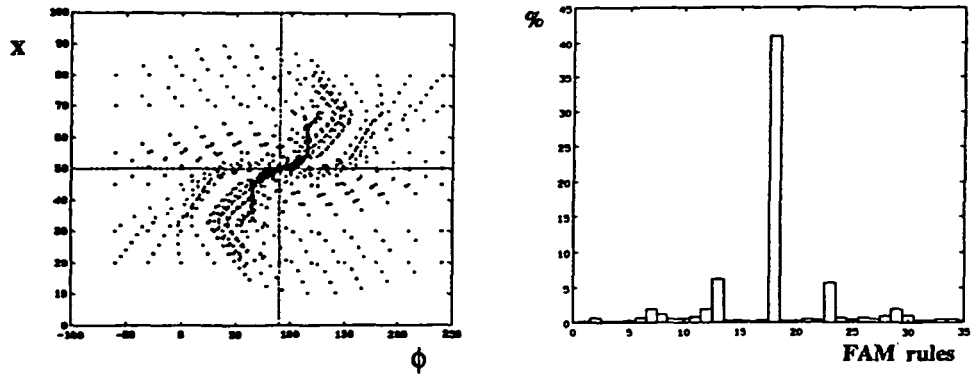
We generated 2230 truck samples from 7 different initial positions and varying angles. We chose the initial positions (20,20), (30,20), (45,20), (50,20), (55,20), (70,20), and (80,20). We changed the angle from $-60°$ to $240°$ at each initial position. At each step, the fuzzy controller produced output steering angle θ. The training vectors (x, ϕ, θ) defined points in a three-dimensional product-space. x had 5 fuzzy set values: LE, LC, CE, RC, and RI. ϕ had 7 fuzzy set values: RB, RU, RV, VE, LV, LU, and LB. θ had 7 fuzzy set values: NB, NM, NS, ZE, PS, PM, and PB. So there were 245 $(5 \times 7 \times 7)$ possible FAM cells.

We defined FAM cells by partitioning the effective product-space. FAM cells near the center were smaller than outer FAM cells because we chose narrow membership functions near the steady-state FAM cell. Uniform partitions of the product-space produced poor estimates of the original FAM rules. As in Figure 3, this reflected the need to judiciously define the fuzzy-set values of the system fuzzy variables.

We performed product-space clustering with the version of DCL discussed in [Kosko, 1990a]. If a FAM cell contained at least one of the 245 synaptic quantization vectors, we entered the corresponding FAM rule in the FAM matrix.

Figure 13a shows the input sample distribution of (x,ϕ). We did not include the variable θ in the figure. Training data clustered near the steady-state position ($x = 50$ and $\phi = 90°$). Figure 13b displays the synaptic-vector histogram after DCL classified 2230 training vectors for 35 FAM rules. Since successful FAM system generated the training samples, most training samples, and thus most synaptic vectors, clustered in the steady-state FAM cell.

DCL product-space clustering estimated 35 new FAM rules. Figure 14 shows the DCL-estimated FAM bank and the corresponding control surface. The DCL-estimated control surface visually resembles the underlying unknown control surface in Figure 5a. The two systems produce nearly equivalent truck-backing behavior. This suggests adaptive product-space clustering can estimate the FAM rules underlying expert behavior in many cases, even when the expert or fuzzy engineer cannot articulate the FAM rules.

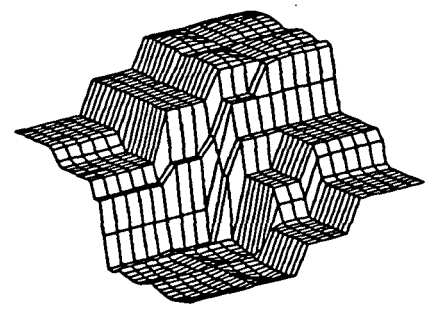


(a) Input data distribution (b) Synaptic-vector histogram

FIGURE 13 (a) Input data distribution, (b) Synaptic-vector histogram. Differential competitive learning allocated synaptic quantization vectors to FAM cells. The steady-state FAM cell (CE, VE; ZE) contained the most synaptic vectors.

$\dot{\phi}$ \ x	LE	LC	CE	RC	RI
RB	PS	PM	PM	PB	PB
RU	ZE	PM	PM	PB	PB
RV	NS	ZE	PS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NB	NS	PS	PS
LU	NB	NB	NB	NS	PS
LB	NB	NB	NM	NM	NS

(a)



(b)

FIGURE 14 (a) DCL-estimated FAM bank. (b) Corresponding control surface.

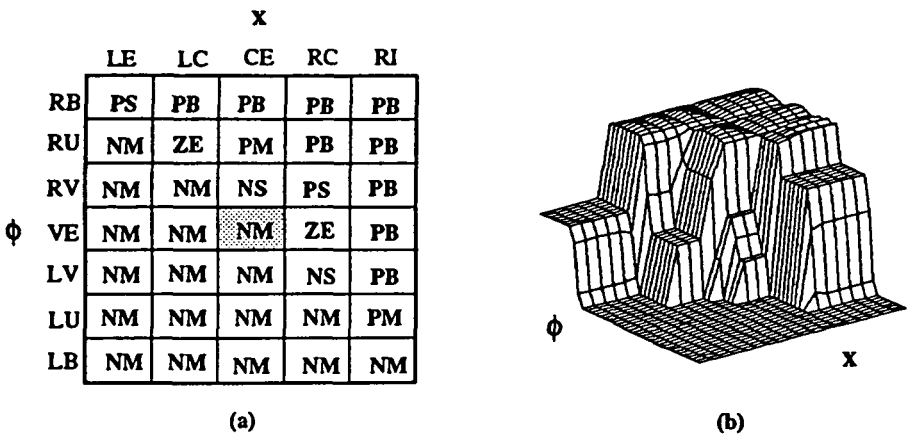


FIGURE 15 (a) FAM bank generated by the neural control surface in Figure 5b. (b) Control surface of the neural BP-AFAM system in (a).

We also used the neural control surface in Figure 5b to estimate FAM rules. We divided the input-output product-space into FAM cells as in the fuzzy control case. If the neural control surface intersected the FAM cell, we entered the corresponding FAM rule in a FAM bank. We averaged all neural control-surface values in a square region over the two input variables x and ϕ. We assigned the average value to one of 7 output fuzzy sets. Figure 15 shows the resulting FAM bank and corresponding control surface generated by the neural control surface in Figure 5b. This new control surface resembles the original fuzzy control surface in Figure 5a more than it resembles the neural control surface in Figure 5b. Note the absence of a steady-state FAM rule in the FAM matrix in Figure 5a.

Figure 16 compares the DCL-AFAM and BP-AFAM control surfaces with the fuzzy control surface in Figure 5a. Figure 16 shows the absolute difference of the control surfaces. As expected, the DCL-AFAM system produced less absolute error than the BP-AFAM system produced.

Figure 17 shows the docking and trajectory errors of the two AFAM control systems. The DCL-AFAM system produced less docking error than the BP-AFAM system produced for 100 arbitrary backing-up trials. The two AFAM systems generated similar backing-up trajectories. This suggests that black-box neural estimators can define the front-end of

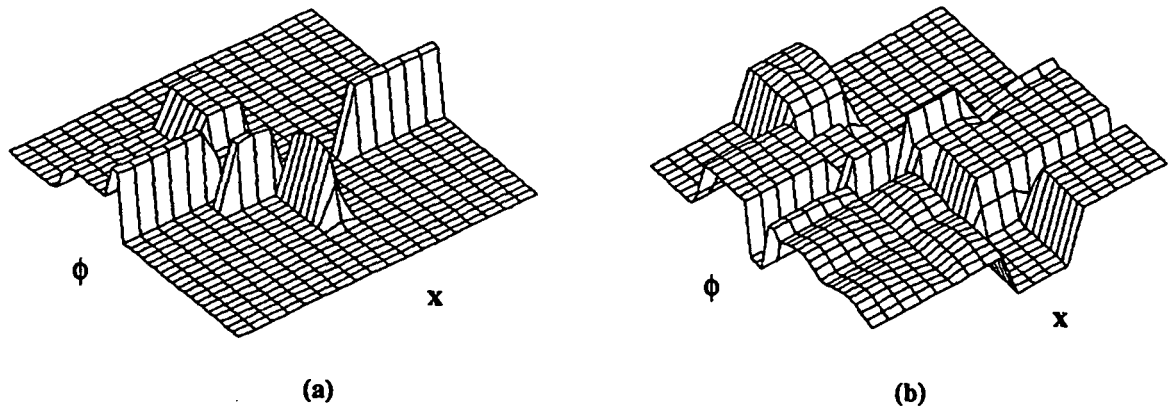


FIGURE 16 (a) Absolute difference of the FAM surface in Figure 5a and the DCL-estimated FAM surface in Figure 14b. (b) Absolute difference of the FAM surface in Figure 5a and the neural-estimated FAM surface in Figure 15b.

FAM-structured systems. In principle we can use this technique to generate structured FAM rules for *any* neural application. We can then inspect and refine these rules and perhaps replace the original neural system with the tuned FAM system.

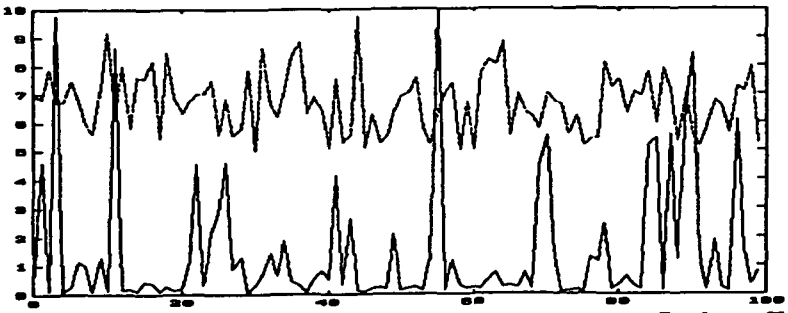
Fuzzy Truck-and-Trailer Controller

We added a trailer to the truck system, as in the original Nguyen-Widrow model. Figure 18 shows the simulated truck-and-trailer system. We added one more variable (cab angle, ϕ_c) to the three state variables of the trailerless truck. In this case a FAM rule takes the form

$$IF\ x = LE\ AND\ \phi_t = RB\ AND\ \phi_c = PO, \qquad THEN\ \beta = NS.$$

The four state variables x, y, ϕ_t, and ϕ_c determined the position of the truck-and-trailer system in the plane. Fuzzy variable ϕ_t corresponded to ϕ for the trailerless truck. Fuzzy variable ϕ_c specified the relative cab angle with respect to the center line along the trailer. ϕ_c ranged from $-90°$ to $90°$. The extreme cab angles $90°$ and $-90°$ corresponded to two

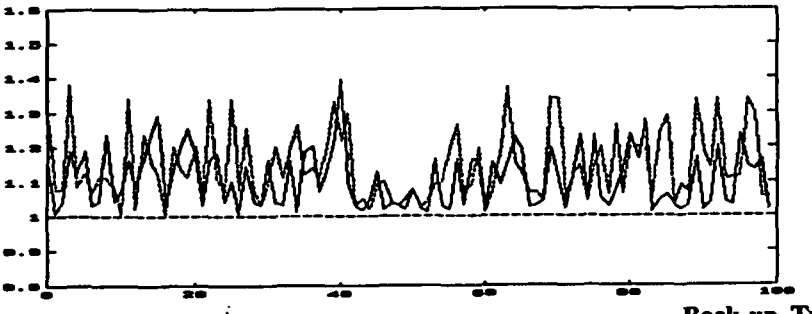
(a) Docking Error



Back-up Trial

- DCL-AFAM (solid) : mean = 1.4449, s.d. = 2.2594
- BP-AFAM (dashed) : mean = 6.6863, s.d. = 1.0665

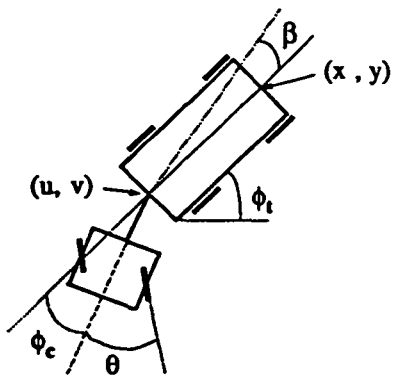
(b) Trajectory Error



Back-up Trial

- DCL-AFAM (solid) : mean = 1.1075, s.d. = 0.0839
- BP-AFAM (dashed) : mean = 1.1453, s.d. = 0.1016

FIGURE 17 (a) Docking errors and (b) Trajectory errors of the DCL-AFAM and BP-AFAM control systems.



(x , y) : Cartesian coordinate of the rear end, [0,100].

(u , v) : Cartesian coordinate of the joint.

ϕ_t : Angle of the trailer with horizontal, [0,360].

ϕ_c : Relative angle of the cab with trailer, [-90,90].

θ : Steering angle, [-30,30].

β : Angle of the trailer updated at each step, [-30,30].

FIGURE 18 Diagram of the simulated truck-and-trailer system.

"jackknife" positions of the cab with respect to the trailer. Positive ϕ_c value indicated that the cab resided on the left-hand side of the trailer. Negative value indicated that it resided on the right-hand side. Figure 18 shows a positive angle value of ϕ_c.

Fuzzy variables x, ϕ_t, and ϕ_c defined the input variables. Fuzzy variable β defined the output variable. β measured the angle that we needed to update the trailer at each iteration. We computed the steering-angle output θ with the following geometric relationship. With the output β value computed, the trailer position (x, y) moved to the new position (x', y'):

$$x' = x + r\cos(\phi_t + \beta), \tag{9}$$

$$y' = y + r\sin(\phi_t + \beta), \tag{10}$$

where r denotes a fixed backing distance. Then the joint of the cab and the trailer (u, v) moved to the new position (u', v'):

$$u' = x' - \ell\cos(\phi_t + \beta), \tag{11}$$

$$v' = y' - \ell\sin(\phi_t + \beta), \tag{12}$$

where ℓ denotes the trailer length. We updated the directional vector $(dirU, dirV)$, which defined the cab angle, by

$$dirU' = dirU + \Delta u, \tag{13}$$

$$dirV' = dirV + \Delta v, \tag{14}$$

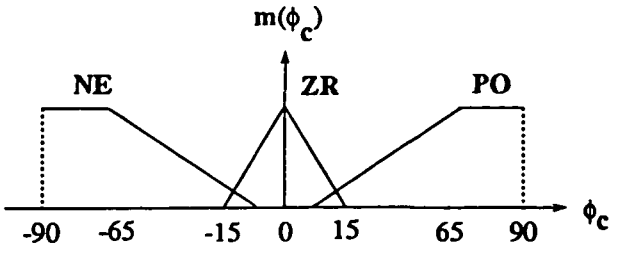


FIGURE 19 Membership graphs of the three fuzzy-set values of fuzzy variable ϕ_c.

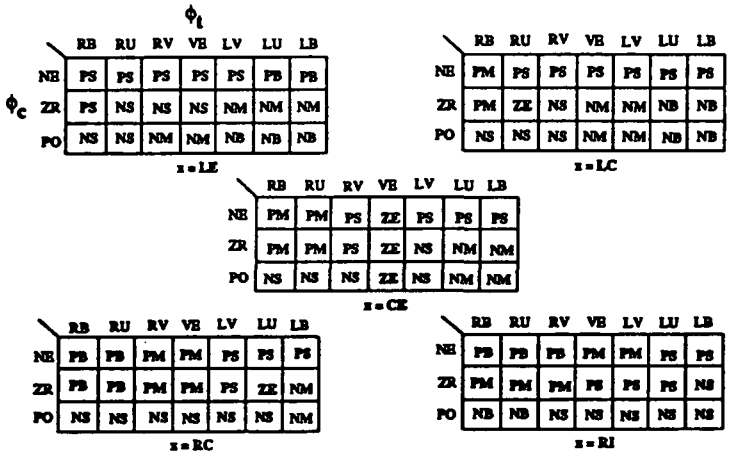


FIGURE 20 FAM bank of the fuzzy truck-and-trailer control system.

where $\Delta u = u' - u$, and $\Delta v = v' - v$. The new directional vector $(dirU', dirV')$ defines the new cab angle ϕ_c'. Then we obtain the steering angle value as $\theta = \phi_{c,h}' - \phi_{c,h}$, where $\phi_{c,h}$ denotes the cab angle with the horizontal. We chose the same fuzzy-set values and membership functions for β as we chose for θ. β ranged from $-30°$ to $30°$. We chose the fuzzy-set values of ϕ_c as NE, ZR and PO as in Figure 19.

Figure 20 displays the 5 FAM-rule matrices in the FAM bank of the fuzzy truck-and-trailer system. In Figure 20 we fixed the fuzzy variable x as LE, LC, CE, RC, and RI. There were 735 ($7 \times 5 \times 7 \times 3$) possible FAM rules and only 105 actual FAM rules.

Figure 21 shows typical backing-up trajectories of the fuzzy truck-and-trailer control

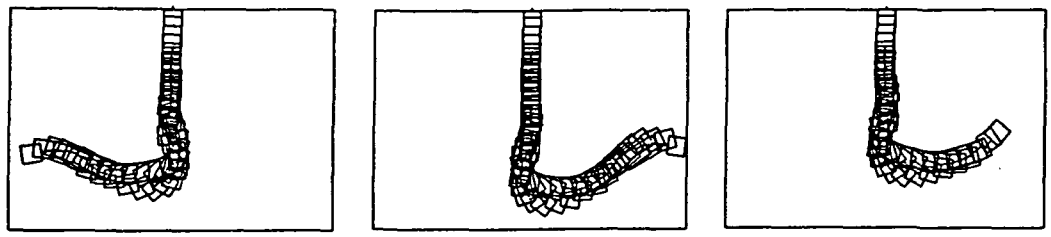


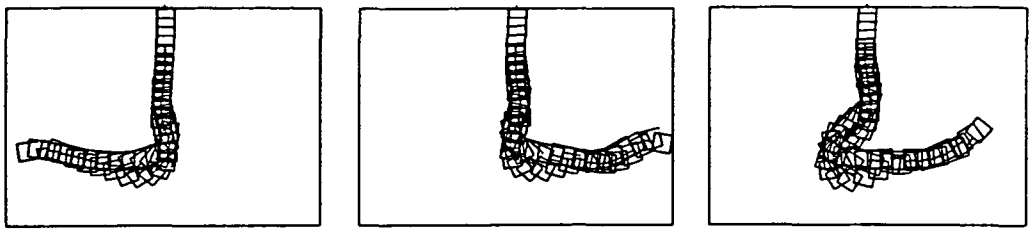
FIGURE 21 Sample truck-and-trailer trajectories from the fuzzy controller for initial positions (x, y, ϕ_t, ϕ_t): (a) (25, 30, −20, 30), (b) (80, 30, 210, −40), and (c) (70, 30, 200, 30).

system from different initial positions. The truck-and-trailer backed up in different directions depending on the relative position of the cab with respect to the trailer. The fuzzy control systems successfully controlled the truck-and-trailer in jackknife positions.

BP Truck-and-Trailer Control Systems

We added the cab-angle variable ϕ_c as to the backpropagation-trained neural truck controller as an input. The controller network contained 24 hidden neurons with output variable β. The training samples consisted of 5-dimensional space of the form $(x, y, \phi_t, \phi_c, \beta)$. We trained the controller network with 52 training samples from the fuzzy controller: 26 samples for the left half of the plane, 26 samples for the right half of the plane. We used equations (9)–(14) instead of the emulator network. Training required more than 200,000 iterations. Some training sequences did not converge. The BP-trained controller performed well except in a few cases. Figure 22 shows typical backing-up trajectories of the BP truck-and-trailer control system from the same initial positions used in Figure 21.

We performed the same robustness tests for the fuzzy and BP-trained truck-and-trailer controllers as in the trailerless truck case. Figure 23 shows performance errors averaged



(a) (b) (c)

FIGURE 22 Sample truck-and-trailer trajectories of the BP-trained controller for initial positions (x, y, ϕ_t, ϕ_t): (a) (25, 30, −20, 30), (b) (80, 30, 210, −40), and (c) (70, 30, 200, 30).

over ten typical back-ups from ten different initial positions. These performance graphs resemble closely the performance graphs for the trailerless truck systems in Figure 12.

AFAM Truck-and-Trailer Control Systems

We generated 6250 truck-and-trailer data using the original FAM system in Figure 20. We backed up the truck-and-trailer from the same initial positions as in the trailerless truck case. The trailer angle ϕ_t ranged from $-60°$ to $240°$, and the cab angle ϕ_c assumed only the three values $-45°$, $0°$, and $45°$. The training vectors $(x, \phi_t, \phi_c, \beta)$ defined points in the four-dimensional input-output product-space. We nonuniformly partitioned the product space into FAM cells to allow narrower fuzzy-set values near the steady-state FAM cell.

We used DCL to train the AFAM truck-and-trailer controller. The total number of FAM cells equaled 735 ($7 \times 5 \times 7 \times 3$). We used 735 synaptic quantization vectors. The DCL algorithm classified the 6250 data into 105 FAM cells. Figure 24 shows the synaptic-vector histogram corresponding to the 105 FAM rules. Figure 25 shows the estimated FAM bank by the DCL algorithm. Figure 26 shows the original and DCL-estimated control surfaces for the fuzzy truck-and-trailer systems.

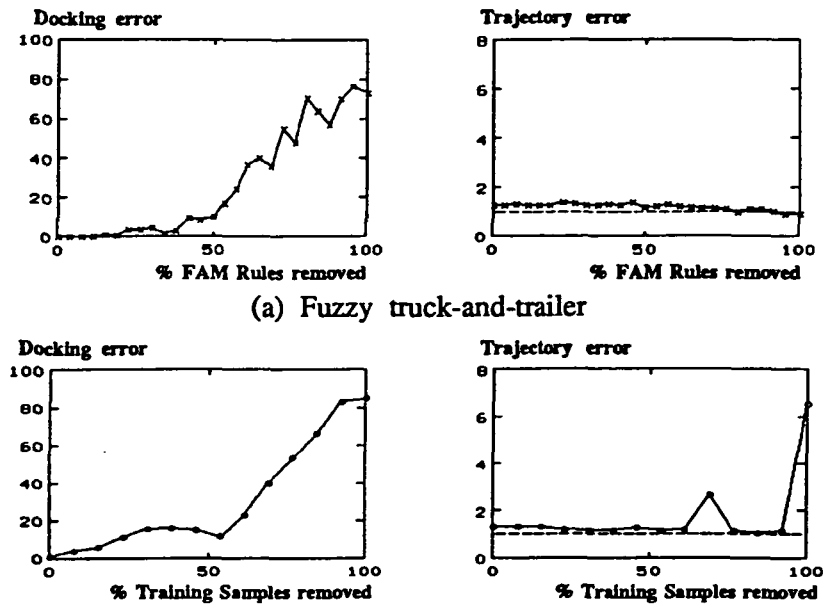


FIGURE 23 Comparison of robustness of the two truck-and-trailer controllers: (a) Docking and trajectory error of the fuzzy controller, (b) Docking and trajectory error of the BP controller.

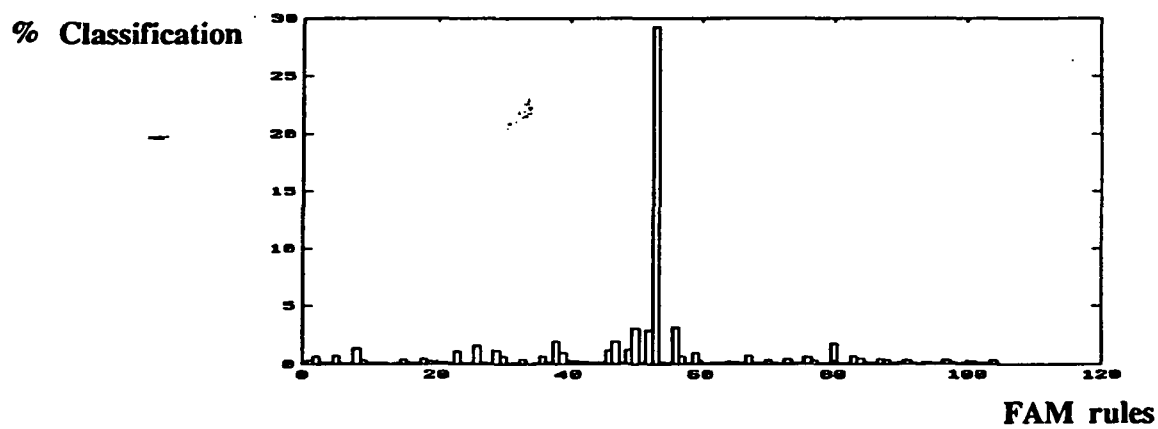


FIGURE 24 Synaptic-vector histogram for the AFAM truck-and-trailer system.

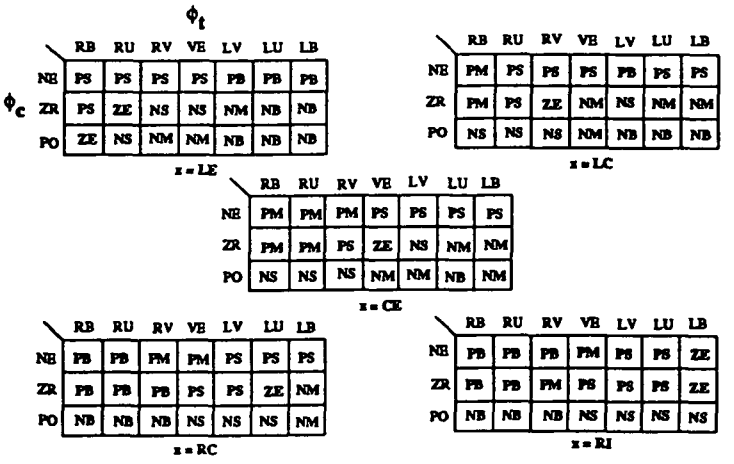
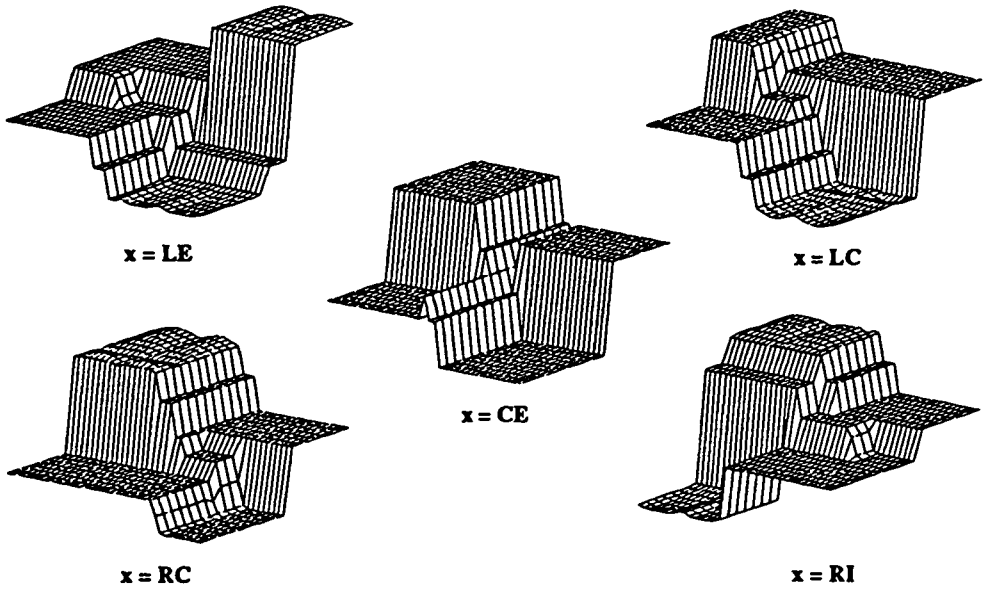
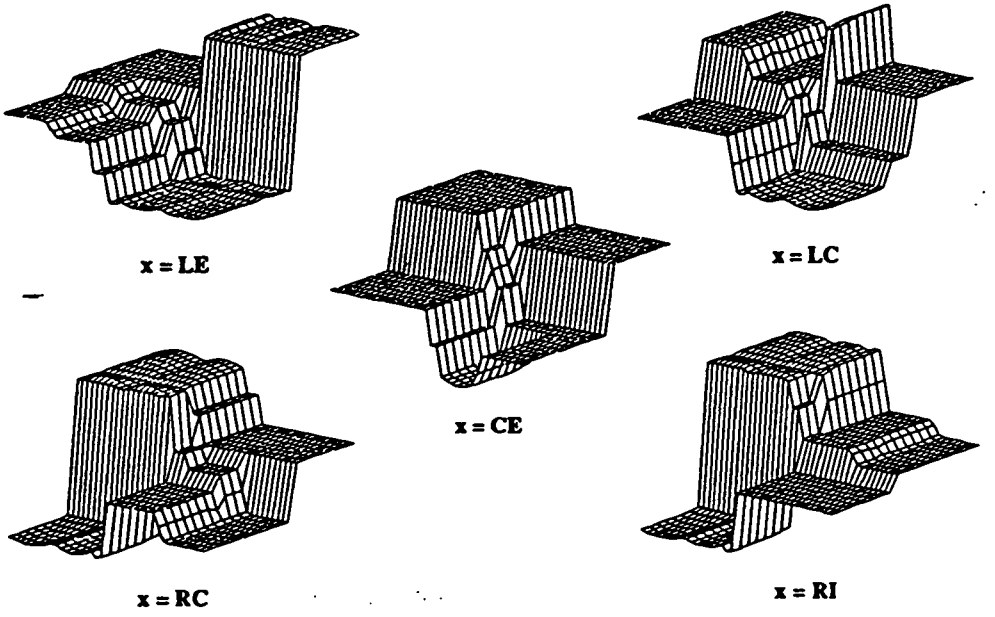


FIGURE 25 DCL-estimated FAM bank for the AFAM truck-and-trailer system.

Figure 27 shows the trajectories of the original FAM and the DCL-estimated AFAM truck-and-trailer controllers. Figure 27a and 27b show the two trajectories from the initial position $(x, y, \phi_t, \phi_c) = (30,30,10,45)$. Figure 27c and 27d show the trajectories from initial position (60,30,210,−60). The original FAM and DCL-estimated AFAM systems exhibited comparable truck-and-trailer control performance except in a few cases, where the DCL-estimated AFAM trajectories were irregular.

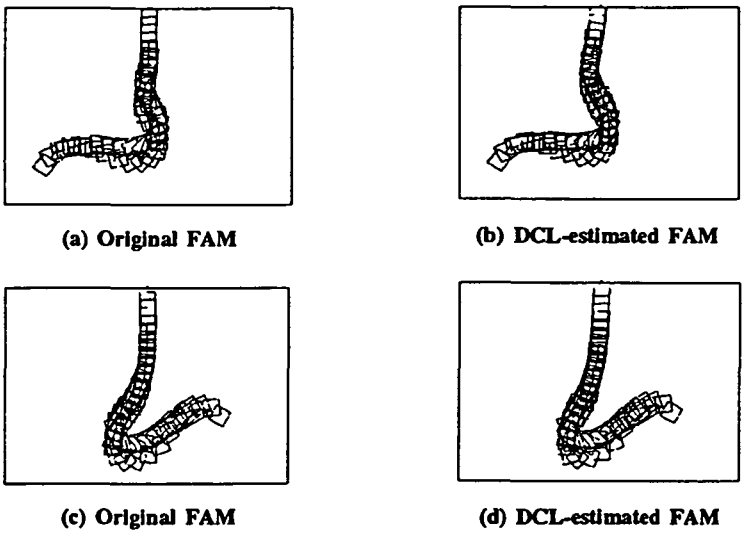


(a) Original control surfaces for the truck-and-trailer system



(b) DCL-estimated control surfaces for the truck-and-trailer system

FIGURE 26 (a) Original control surface (b) DCL-estimated control surface



(a) Original FAM (b) DCL-estimated FAM

(c) Original FAM (d) DCL-estimated FAM

FIGURE 27 Sample truck-and-trailer trajectories from the original and the DCL-estimated FAM systems starting at initial positions $(x, y, \phi_t, \phi_c) =$ (30,30,10,45) and (60,30,210,−60).

Conclusion

We quickly engineered fuzzy systems to successfully back up a truck and truck-and-trailer system in a parking lot. We used only common sense and error-nulling intuitions to generate sufficient banks of FAM rules. These systems performed well until we removed over 50 % of the FAM rules. This extreme robustness suggests that, for many estimation and control problems, different fuzzy engineers can rapidly develop prototype fuzzy systems that perform similarly and well.

The speed with which the DCL clustering technique recovers the underlying FAM bank further suggests that we can likewise construct fuzzy systems for more complex, higher-dimensional problems. For these problems we may have access to only incomplete numerical input-output data. Pure neural-network or statistical-process-control approaches may generate systems with comparable performance. But these systems will involve far greater computational effort, will be more difficult to modify, and will not provide a structured

representation of the system's throughput.

Our neural experiments suggests that whenever we model a system with a neural network, for little extra computational cost we can generate a set of structured FAM rules that approximate the neural system's behavior. We can then tune the fuzzy system by refining the FAM-rule bank with fuzzy-engineering rules of thumb and with further training data.

Acknowledgment

This research was supported by the Air Force Office of Scientific Research (AFOSR-88-0236) and by a grant from the Rockwell Science Center.

References

Huber, P.J., *Robust Statistics*, Wiley, 1981.

Kong, S.G. and Kosko, B., "Differential Competitive Learning for Centroid Estimation and Phoneme Recognition," *IEEE Transactions on Neural Networks*, to appear, January 1991.

Kosko, B., "Fuzzy Entropy and Conditioning," *Information Sciences*, vol.40, 165-174, 1986.

Kosko, B., *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, 1990.

Kosko, B., "Stochastic Competitive Learning," *Proceedings of the Summer 1990 International Joint Conference on Neural Networks (IJCNN-90)*, vol.II, 215-226, June 1990.

Kosko, B., "Unsupervised Learning in Noise," *IEEE Transactions on Neural Networks*, vol.1, no.1, 44-57, March 1990.

Nguyen, D. and Widrow, B., "The Truck Backer-Upper: An Example of Self-Learning in Neural Networks," *Proceedings of International Joint Conference on Neural Networks (IJCNN-89)*, vol.II, 357-363, June 1989.

APPENDIX: Product-space Clustering with Differential Competitive Learning

Product-space clustering [Kosko, 1990a] is a form of stochastic adaptive vector quantization. Adaptive vector quantization (AVQ) systems adaptively quantize pattern clusters in R^n. Stochastic competitive learning systems are neural AVQ systems. Neurons compete for the activation induced by randomly sampled patterns. The corresponding synaptic fan-in vectors adaptively quantize the pattern space R^n. The p synaptic vectors $\mathbf{m}_j$ define the p columns of the synaptic connection matrix M. M interconnects the n input or linear neurons in the input neuronal field F_X to the p competing nonlinear neurons in the output field F_Y. Figure 28 below illustrates the neural network topology.

Learning algorithms estimate the unknown probability density function $p(\mathbf{x})$, which describes the distribution of patterns in R^n. More synaptic vectors arrive at more probable regions. Where sample vectors $\mathbf{x}$ are dense or sparse, synaptic vectors $\mathbf{m}_j$ should be dense or sparse. The local count of synaptic vectors then gives a nonparametric estimate of the volume probability $P(V)$ for volume $V \subset R^n$:

$$P(V) = \int_V p(\mathbf{x})\, d\mathbf{x} \tag{15}$$

$$\approx \frac{\text{Number of } \mathbf{m}_j \in V}{p} \quad . \tag{16}$$

In the extreme case that $V = R^n$, this approximation gives $P(V) = p/p = 1$. For improbable subsets V, $P(V) = 0/p = 0$.

Stochastic Competitive Learning Algorithms

The metaphor of competing neurons reduces to nearest-neighbor classification. The AVQ system compares the current vector random sample $\mathbf{x}(t)$ in Euclidean distance to the p columns of the synaptic connection matrix M, to the p synaptic vectors $\mathbf{m}_1(t), \ldots, \mathbf{m}_p(t)$. If the jth synaptic vector $\mathbf{m}_j(t)$ is closest to $\mathbf{x}(t)$, then the jth output neuron "wins" the competition for activation at time t. In practice we sometimes define the nearest N synaptic vectors as winners. Some scaled form of $\mathbf{x}(t) - \mathbf{m}_j(t)$ updates the nearest or "winning" synaptic vectors. "Losers" remain unchanged: $\mathbf{m}_i(t+1) = \mathbf{m}_i(t)$. Competitive synaptic vectors converge to pattern-class centroids exponentially fast [Kosko, 1990b].

The following three-step process describes the competitive AVQ algorithm, where the third step depends on which learning algorithm updates the winning synaptic vectors.

Competitive AVQ Algorithm

1. Initialize synaptic vectors: $\mathbf{m}_i(0) = \mathbf{x}(i), \quad i = 1, \ldots, p$.
   Sample-dependent initialization avoids many pathologies that can distort nearest-neighbor learning.

2. For random sample $\mathbf{x}(t)$, find the closest or "winning" synaptic vector $\mathbf{m}_j(t)$:

$$\|\mathbf{m}_j(t) - \mathbf{x}(t)\| = \min_i \|\mathbf{m}_i(t) - \mathbf{x}(t)\| \quad , \tag{17}$$

   where $\|\mathbf{x}\|^2 = x_1^2 + \cdots + x_n^2$ defines the squared Euclidean vector norm of $\mathbf{x}$. We can define the N synaptic vectors closest to $\mathbf{x}$ as "winners".

3. Update the winning synaptic vector(s) $\mathbf{m}_j(t)$ with an appropriate learning algorithm.

Differential competitive learning (DCL)

Differential competitive "synapses" learn only if the competing "neuron" changes its competitive status [Kosko, 1990c]:

$$\dot{m}_{ij} = \dot{S}_j(y_j)\,[\,S_i(x_i) - m_{ij}\,] \quad , \tag{18}$$

or in vector notation,

$$\dot{\mathbf{m}}_j = \dot{S}_j(y_j)\,[\,\mathbf{S}(\mathbf{x}) - \mathbf{m}_j\,] \quad , \tag{19}$$

where $\mathbf{S}(\mathbf{x}) = (S_1(x_1), \ldots, S_n(x_n))$ and $\mathbf{m}_j = (m_{1j}, \ldots, m_{nj})$. m_{ij} denotes the synaptic weight between the ith neuron in input field F_X and the jth neuron in competitive field F_Y. Nonnegative signal functions S_i and S_j transduce the real-valued activations x_i and y_j into bounded monotone nondecreasing signals $S_i(x_i)$ and $S_j(y_j)$. $\dot{m}_{ij}$ and $\dot{S}_j(y_j)$ denote the time derivatives of m_{ij} and $S_j(y_j)$, synaptic and signal velocities. $S_j(y_j)$ measures the competitive status of the jth competing neuron in F_Y. Usually S_j approximates a binary threshold function. For example, S_j may equal a steep binary logistic sigmoid,

$$S_j(y_j) = \frac{1}{1 + e^{-cy_j}} \quad , \tag{20}$$

for some constant $c > 0$. The jth neuron wins the laterally inhibitive competition if $S_j = 1$, loses if $S_j = 0$.

For discrete implementation, we use the DCL algorithm as a stochastic difference equation [Kong, 1991]:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + c_t\,\Delta S_j(y_j(t))\,[\,S(\mathbf{x}(t)) - \mathbf{m}_j(t)\,] \text{ if the } j\text{th neuron wins,} \tag{21}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) \qquad \text{if the } i\text{th neuron loses.} \tag{22}$$

$\Delta S_j(y_j(t))$ denotes the time change of the jth neuron's competition signal $S_j(y_j)$ in the competitive field F_Y:

$$\Delta S_j(y_j(t)) = \mathrm{sgn}[\,S_j(y_j(t+1)) - S_j(y_j(t))\,] \quad . \tag{23}$$

We define the signum operator $\mathrm{sgn}(x)$ as

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{if} \quad x = 0 \\ -1 & \text{if} \quad x < 0 \end{cases} \quad . \tag{24}$$

$\{c_t\}$ denotes a slowly decreasing sequence of learning coefficients, such as $c_t = .1\,(1 - t/2000)$ for 2000 training samples. Stochastic approximation [Huber, 1981] requires a decreasing gain sequence $\{c_t\}$ to suppress random disturbances and to guarantee convergence to local minima of mean-squared performance measures. The learning coefficients should decrease slowly,

$$\sum_{t=1}^{\infty} c_t = \infty \quad , \tag{25}$$

but not too slowly,

$$\sum_{t=1}^{\infty} c_t^2 < \infty \quad . \tag{26}$$

Harmonic-series coefficients, $c_t = 1/t$, satisfy these constraints.

We approximate the competitive signal difference ΔS_j as the activation difference Δy_j:

$$\Delta S_j(y_j(t)) = \mathrm{sgn}[\, y_j(t+1) - y_j(t) \,] \tag{27}$$

$$= \Delta y_j(t) \quad . \tag{28}$$

Input neurons in feedforward networks usually behave linearly: $S_i(x_i) = x_i$, or $S(\mathbf{x}(t)) = \mathbf{x}(t)$. Then we update the winning synaptic vector $\mathbf{m}_j(t)$ with

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + c_t\, \Delta y_j(t)\, [\, \mathbf{x}(t) - \mathbf{m}_j(t) \,] \quad . \tag{29}$$

We update the F_Y neuronal activations y_j with the additive model

$$y_j(t+1) = y_j(t) + \sum_{i}^{n} S_i(x_i(t))\, m_{ij}(t) + \sum_{k}^{p} S_k(y_k(t))\, w_{kj} \quad . \tag{30}$$

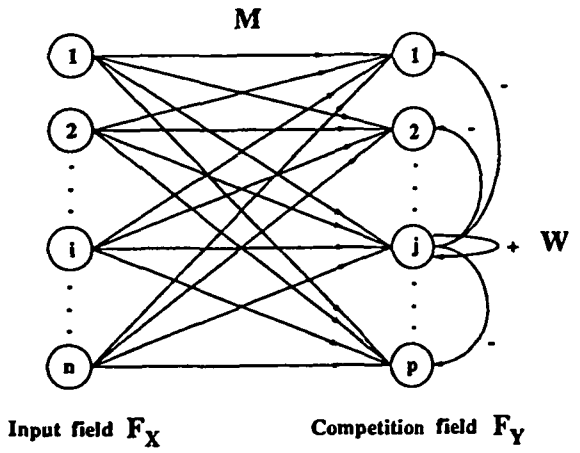


FIGURE 28 Topology of the laterally inhibitive DCL network.

For linear signal functions S_i, the first sum in (30) reduces to an inner product of sample and synaptic vectors:

$$\sum_{i}^{n} x_i(t)\, m_{ij}(t) \;=\; \mathbf{x}^T(t)\, \mathbf{m}_j(t) \quad . \tag{31}$$

Then positive learning tends to occur—$\Delta m_{ij} > 0$—when $\mathbf{x}$ is close to the jth synaptic vector $\mathbf{m}_j$.

Since a binary threshold function approximates the output signal function $S_k(y_k)$, the second sum in (30) sums over just the winning neurons: $\sum_k w_{kj}$ for all winning neurons y_k .

The $p \times p$ matrix W contains the F_Y within-field synaptic connection strengths. Diagonal elements w_{ii} are positive, off-diagonal elements negative. Winning neurons excite themselves and inhibit all other neurons. Figure 28 shows the connection topology of the laterally inhibitive DCL network.

Product-space clustering

We divided the space $0 \leq x \leq 100$ into five nonuniform intervals $[0, 32.5]$, $[32.5, 47.5]$, $[47.5, 52.5]$, $[52.5, 67.5]$, and $[67.5, 100]$. Each interval represented the five fuzzy-set values LE, LC, CE, RC, and RI. This choice corresponded to the nonoverlapping intervals of the fuzzy membership function graphs $m(x)$ in Figure 3. Similarly, we divided the space $-90 \leq \phi \leq 270$ into seven nonuniform intervals $[-90, 0]$, $[0, 66.5]$, $[66.5, 86]$, $[86, 94]$, $[94, 113.5]$, $[113.5, 182.5]$, and $[182.5, 270]$, which corresponded respectively to RB, RU, RV, VE, LV, LU, and LB. We divided the space $-30 \leq \theta \leq 30$ into seven nonuniform intervals $[-30, -20]$, $[-20, -7.5]$, $[-7.5, -2.5]$, $[-2.5, 2.5]$, $[2.5, 7.5]$, $[7.5, 20]$, and $[20, 30]$, which corresponded to NB, NM, NS, ZE, PS, PM, and PB.

DCL classified each input-output data vector into one of the FAM cells. We added a FAM rule to the FAM bank if the DCL-trained synaptic vector fell in the FAM cell. In case of ties we chose the FAM cell with the most densely clustered data.

For the BP-AFAM generated from the neural control surface in Figure 15, we divided the rectangle $[0, 100] \times [-90, 270]$ into 35 nonuniform squares with the same divisions defined above. Then we added and averaged the control surface values in the square. We added a FAM rule to the FAM bank if the averaged value corresponded to one of the seven FAM cells.

For the truck-and-trailer case, we divided the space $-90 \leq \phi_c \leq 90$ into three intervals $[-90, -12.\overline{5}]$, $[-12.5, 12.5]$, and $[12.5, 90]$, which corresponded to NE, ZR, and PO. There were 735 FAM cells, and 735 possible FAM rules, of the form $(x, \phi_t, \phi_c; \beta)$.

In Review: Oct. 1990

ADAPTIVE FUZZY SYSTEM FOR TARGET TRACKING

Peter J. Pacini and Bart Kosko
Department of Electrical Engineering
Signal and Image Processing Institute
University if Southern California
Los Angeles, CA 90089-0272

ABSTRACT

We compared fuzzy and Kalman-filter control systems for realtime target tracking. Both systems performed well, but in the presence of mild process (unmodeled effects) noise the fuzzy system exhibited finer control. We tested the robustness of the fuzzy controller by removing random subsets of fuzzy associations or "rules" and by adding destructive or "sabotage" fuzzy rules to the fuzzy system. We tested the robustness of the Kalman tracking system by increasing the variance of the unmodeled-effects noise process. The fuzzy controller performed well until we removed over 50% of the fuzzy rules. The Kalman controller's performance quickly degraded as the unmodeled-effects variance increased. We used unsupervised neural-network learning to adaptively generate the fuzzy controller's fuzzy-associative-memory structure. The fuzzy systems did not require a mathematical model of how system outputs depended on inputs.

Fuzzy and Math-Model Controllers

Fuzzy controllers differ from classical math-model controllers. Fuzzy controllers do not require a mathematical model of how control outputs functionally depend on control inputs. Fuzzy controllers also differ in the type of uncertainty they represent and how they represent it. The fuzzy approach represents ambiguous or fuzzy system behavior as partial implications or approximate "rules of thumb"—as fuzzy associations (A_i, B_i).

Fuzzy controllers are fuzzy systems. A finite *fuzzy set* A is a *point* [Kosko, 1987] in a unit hypercube $I^n = [0,1]^n$. A *fuzzy system* $F : I^n \rightarrow I^p$ is a *mapping* between unit hypercubes. I^n contains all fuzzy subsets of the domain space $X = \{x_1, \ldots, x_n\}$. I^n is the *fuzzy power set* $F(2^X)$ of X. I^p contains all the fuzzy subsets of the range space $Y = \{y_1, \ldots, y_p\}$. Element $x_i \in X$ belongs to fuzzy set A to degree $m_A(x_i)$. The 2^n nonfuzzy subsets of X correspond to the 2^n corners of the fuzzy cube I^n. The fuzzy system F maps fuzzy subsets of X to fuzzy subsets of Y. In general, X and Y are continuous not discrete sets.

Math-model controllers usually represent system uncertainty with probability distributions. Probability models describe system behavior with first-order and second-order statistics—with conditional means and covariances. They usually describe unmodeled effects and measurement imperfections with additive "noise" processes.

Mathematical models of the system state and measurement processes facilitate a mean-squared-error analysis of system behavior. In general we cannot accurately articulate such mathematical models. This greatly restricts the range of realworld applications. In practice we often use linear or quasi-linear (Markov) mathematical models.

Mathematical state and measurement models also make it difficult to add non-mathematical knowledge to the system. Experts may articulate such knowledge, or neural networks may adaptively infer it from sample data. In practice, once we have articulated the math model, we use human expertise only to estimate the initial state and covariance conditions.

Fuzzy controllers consist of a bank of *fuzzy associative memory* (FAM) "rules" or associations (A_i, B_i) operating in parallel, and operating to different degrees. Each FAM

rule is a set-level implication. It represents ambiguous expert knowledge or learned input-output transformations. A FAM rule can also summarize the behavior of a specific mathematical model. The system nonlinearly transforms exact or fuzzy state inputs to a fuzzy set output. This output fuzzy set is usually "defuzzified" with a centroid operation to generate an exact numerical output. In principle the system can use the entire fuzzy distribution as the output. We can easily construct, process, and modify the FAM bank of FAM rules in software or in digital VLSI circuitry.

Fuzzy controllers require that we articulate or estimate the FAM rules. The fuzzy-set framework provides more expressiveness than, say, traditional expert-system approaches, which encode bivalent propositional associations. But the fuzzy framework does not eliminate the burden of knowledge acquisition. We can use neural network systems to estimate the FAM rules. But neural systems also require an accurate (statistically representative) set of articulated input-output numerical samples. Below we use unsupervised competitive learning to adaptively generate target-tracking FAM rules.

Experts can hedge their system descriptions with fuzzy concepts. Although fuzzy controllers are numerical systems, experts can contribute their knowledge in natural language. This is especially important in complex problem domains, such as economics, medicine, and history, where we may not know how to mathematically model system behavior.

Below we compare a fuzzy controller with a Kalman-filter controller for realtime target tracking. This problem admits a simple and reasonably accurate mathematical description of its state and measurement processes. We chose the Kalman filter as a benchmark because of its many optimal linear-systems properties. We wanted to see whether this "optimal" controller remains optimal when compared with a computationally lighter fuzzy controller in different uncertainty environments.

We indirectly compared the sensitivity of the two controllers by varying their system uncertainties. We randomly removed FAM rules from the fuzzy controller. We also added "sabotage" FAM rules to the controller. Both techniques modeled less-stuctured control environments. For the Kalman filter, we varied the noise variance of the unmodeled-effects noise process.

Both systems performed well for mildly uncertain target environments. They degraded

differently as the system uncertainty increases. The fuzzy controller's performance degraded when we removed more than half the FAM rules. The Kalman-filter controller's performance quickly degraded when the additive state noise process increased in variance.

Realtime Target Tracking

A target tracking system maps azimuth-elevation inputs to motor control outputs. The nominal target moves through azimuth-elevation space. Two motors adjust the position of a platform to continuously point at the target.

The platform can be any directional device that accurately points at the target. The device may be a laser, video camera, or high-gain antenna. We assume we have available a radar or other device that can detect the direction from the platform to the target.

The radar sends azimuth and elevation coordinates to the tracking system at the end of each time interval. We calculate the current *error* e_k in platform position and *change in error* $\dot{e}_k$. Then a fuzzy or Kalman-filter controller determines the control outputs for the motors, one each for azimuth and elevation. The control outputs reposition the platform.

We can independently control movement along azimuth and elevation if we apply the same algorithm twice. This reduces the problem to matching the target's position and velocity in only one dimension.

Figure 1 shows a block diagram of the target tracking system. The controller's output v_k gives the estimated change in angle required during the next time interval. In principle a hardware system must transduce the angular velocity v_k into a voltage or current.

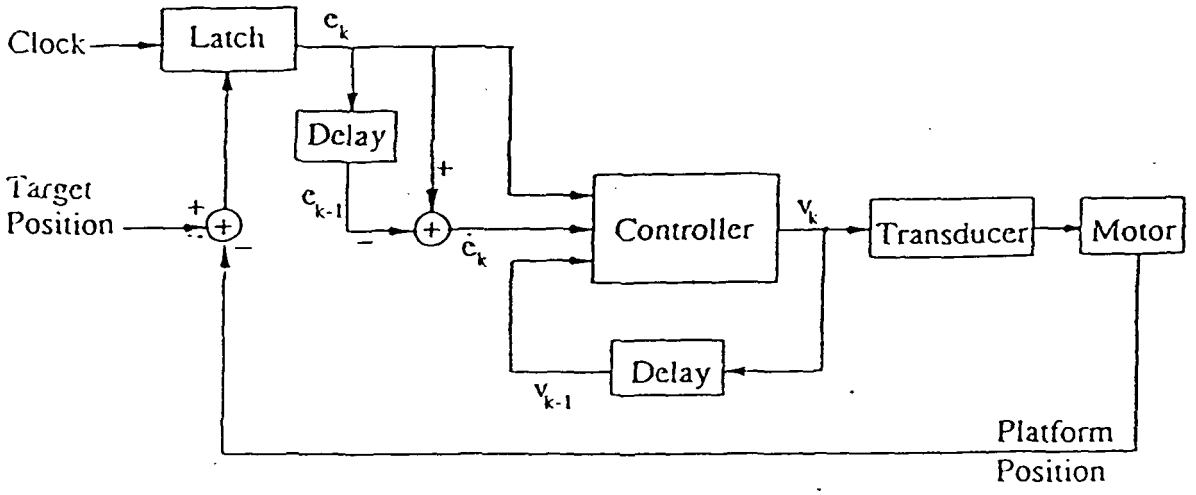


FIGURE 1 Target tracking system.

FUZZY CONTROLLER

We restrict the output angular velocity v_k of the fuzzy controller to the interval $[-6, 6]$. So we must insert a gain element before the voltage transduction. This gain must equal one-sixth the maximum angle through which the platform can turn in one time interval. Similarly, the position error e_k must be scaled so that 6 equals the maximum error. The product of this scale factor and the output gain provides a design parameter—the "gain" of the fuzzy controller.

The fuzzy controller uses heuristic control set-level "rules" or *fuzzy associative memory* (FAM) associations based on quantized values of e_k, $\dot{e}_k$, and v_{k-1}. We define seven fuzzy levels by the following library of fuzzy-set values of the fuzzy variables e_k, $\dot{e}_k$, and v_{k-1}:

LN : Large Negative

MN : Medium Negative

SN : Small Negative

ZE : Zero

SP : Small Positive

MP : Medium Positive

LP : Large Positive

We do not quantize inputs in the classical sense that we assign each input to exactly one output level. Instead, each linguistic value equals as a fuzzy set that overlaps with adjacent fuzzy sets. The fuzzy controller uses trapezoidal fuzzy-set values, as Figure 2 shows. The lengths of the upper and lower bases provide design parameters that we must calibrate for satisfactory performance. A good rule of thumb is *adjacent fuzzy-set values should overlap approximately 25 percent.* Below we discuss examples of calibrated and uncalibrated systems. The fuzzy controller attained its best performance with upper and lower bases of 1.2 and 3.9—26.2% overlap. Different target scenarios may require more or less overlap.

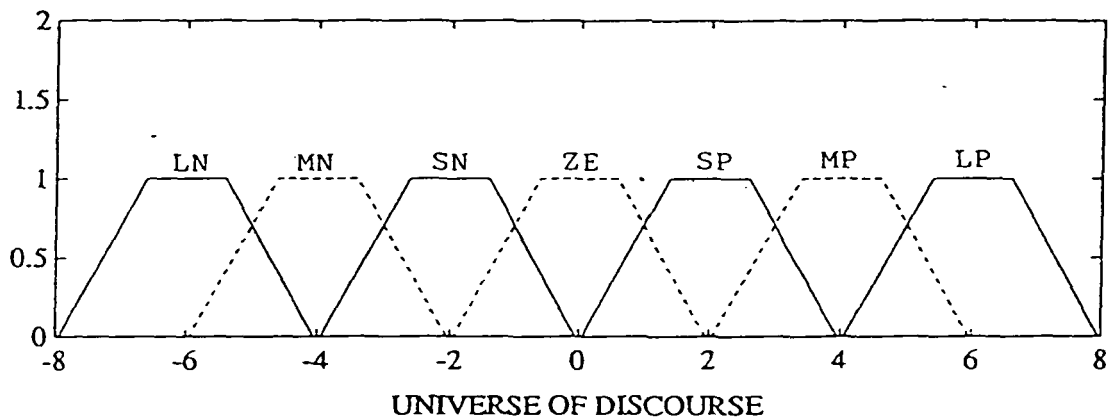


FIGURE 2 Library of overlapping fuzzy-set values defined on a universe

of discourse.

We assign each system input to a fit vector of length 7, where the ith *fit*, or *fuzzy unit* [Kosko, 1986], equals the value of the ith fuzzy set at the input value. In other words, the ith fit measures the degree to which the input belongs to the ith fuzzy-set value. For instance, we apply the input values $1, -4$, and 3.8 to the seven fuzzy sets in the library to obtain the fit vectors

$$\begin{aligned} 1 &\longrightarrow (0 \;\; 0 \;\; 0 \;\; .7 \;\; .7 \;\; 0 \;\; 0) \;\; , \\ -4 &\longrightarrow (0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0) \;\; , \\ 3.8 &\longrightarrow (0 \;\; 0 \;\; 0 \;\; 0 \;\; .1 \;\; 1 \;\; 0) \;\; . \end{aligned}$$

We determine these fit values above by convolving a Dirac delta function centered at the input value with each of the 7 fuzzy sets:

$$m_{SP}(3.8) \;=\; \delta(y - 3.8) * m_{SP}(y) \;=\; .1 \;\; . \qquad (1)$$

If we use a discretized universe of discourse, then we use a Kronecker delta function instead. Equivalently, for the discrete case n-dimensional universe of discourse $X = \{x_1, \ldots, x_n\}$, a control input corresponds to a *bit* (binary unit) vector B of length n. A single 1 element in the ith slot represents the "crisp" input value x_i. Similarly, we represent the kth library fuzzy set by an n-dimensional fit vector A_k that contains samples of the fuzzy set at the n discrete points within the universe of discourse X. The degree to which the crisp input x_i activates each fuzzy set equals the inner product $B \cdot A_k$ of the bit vector B and the corresponding fit vector A_k.

We formulate control FAM rules by associating output fuzzy sets with input fuzzy sets. The antecedent of each FAM rule conjoins e_k, $\dot{e}_k$, and v_{k-1} fuzzy-set values. For example,

$$\text{IF } e_k = MP \text{ AND } \dot{e}_k = SN \text{ AND } v_{k-1} = ZE, \text{ THEN } v_k = SP.$$

We abbreviate this as $(MP, SN, ZE; SP)$.

The scalar activation value w_i of the ith FAM rule's consequent equals the *minimum* of the three antecedent conjuncts' values. If alternatively we combine the antecedents disjunctively with OR, the activation degree of the consequent would equal the *maximum* of the three antecedent disjuncts' values. In the following example, $m_A(e_k)$ denotes the degree to which e_k belongs to the fuzzy set A:

			LN	MN	SN	ZE	SP	MP	LP
e_k	= 2.6	$\longrightarrow$	(0	0	0	0	1	.4	0)
$\dot{e}_k$	= -2.0	$\longrightarrow$	(0	0	1	0	0	0	0)
v_{k-1}	= 1.8	$\longrightarrow$	(0	0	0	.1	1	0	0)

$$m_{MP}(e_k) = .4$$

$$m_{SN}(\dot{e}_k) = 1$$

$$m_{ZE}(v_{k-1}) = .1$$

$$w_i = \min(.4,\ 1,\ .1) = .1 \ .$$

So the system activates the consequent fuzzy set SP to degree $w_i = .1$.

The output fuzzy set's shape depends on the FAM-rule encoding scheme used. With *correlation-minimum* encoding, we clip the consequent fuzzy set L_i in the library of fuzzy-set values to degree w_i with pointwise minimum:

$$m_{O_i}(y) = \min(w_i, m_{L_i}(y)) \ . \tag{2}$$

With *correlation-product* encoding, we multiply L_i by w_i:

$$m_{O_i}(y) = w_i \, m_{L_i}(y) \ , \tag{3}$$

or equivalently,

$$O_i = w_i \, L_i \ . \tag{4}$$

Figure 3 illustrates how both inference procedures transform L_i to scaled output O_i. For

the example above, correlation-product inference gives output fuzzy set $O_i = .1SP$, where $L_i = SP$ denotes the fuzzy set of small but positive angular velocity values.

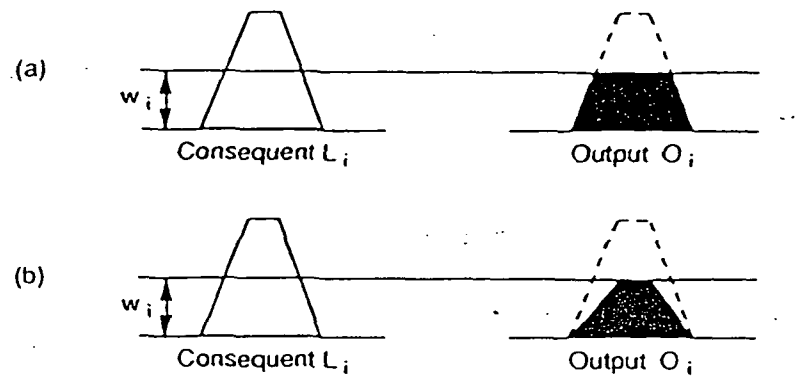


FIGURE 3 FAM inference procedure depends on FAM rule encoding procedure: (a) correlation-minimum encoding, (b) correlation-product encoding.

The fuzzy system activates each FAM rule consequent set to a different degree. For the ith FAM rule this yields the output fuzzy set O_i. The system then sums the O_i to form the combined output fuzzy set O:

$$O = \sum_{i=1}^{N} O_i \ , \tag{5}$$

or equivalently,

$$m_O(y) = \sum_{i=1}^{N} m_{O_i}(y) \ . \tag{6}$$

The control output v_k equals the *fuzzy centroid* of O:

$$v_k = \frac{\int y \, m_O(y) dy}{\int m_O(y) dy} \ , \tag{7}$$

where the limits of integration correspond to the entire universe of discourse Y of angular velocity values. Figure 4 shows an example of correlation-product inference for two FAM rules followed by centroid defuzzification of the combined output fuzzy set.

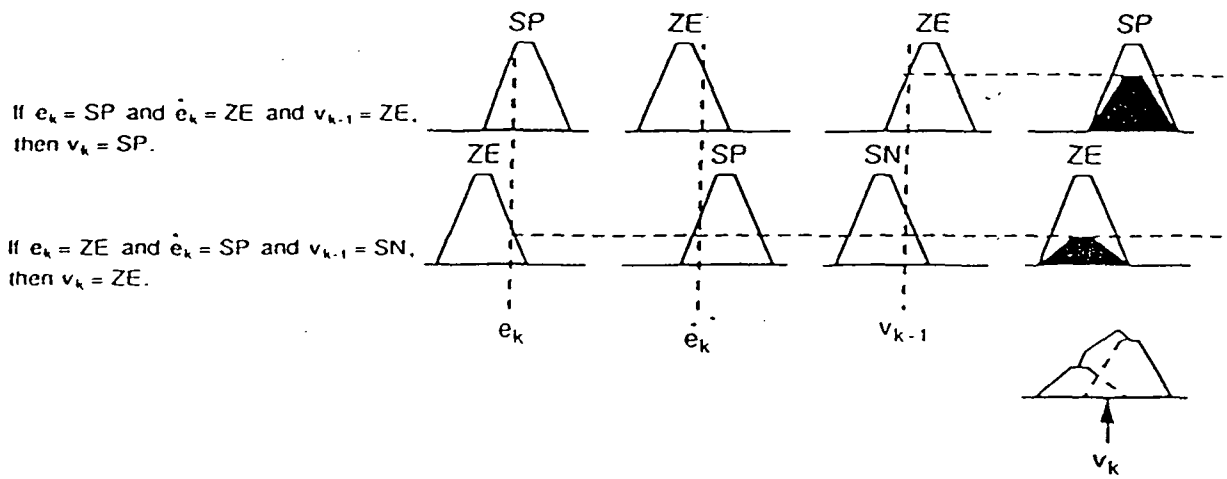


FIGURE 4 Correlation-product inferences followed by centroid defuzzification. FAM rule antecedents combined with AND use the *minimum* fit value to activate consequents. Those combined with OR use the *maximum* fit value.

To reduce computations, we can discretize the output universe of discourse Y to p values, $Y = \{y_1, \ldots, y_p\}$, which gives the discrete fuzzy centroid

$$v_k = \frac{\sum_{j=1}^{p} y_j \, m_O(y_j)}{\sum_{j=1}^{p} m_O(y_j)} \,. \tag{8}$$

Fuzzy Centroid Computation

We now develop two discrete methods for computing the fuzzy centroid (7). Theorem 1 states that we can compute the global centroid v_k from local FAM-rule centroids. Theorem 2 states that v_k can be computed from only 7 sample points if all the fuzzy sets

are symmetric and unimodal (in the broad sense of a trapezoid peak), though otherwise arbitrary. Both results reduce computation and favor digital implementation.

Theorem 1: If correlation-product inference determines the output fuzzy sets, then we can compute the global centroid v_k from local FAM-rule centroids:

$$v_k = \frac{\sum_{i=1}^{N} w_i c_i I_i}{\sum_{i=1}^{N} w_i I_i} . \tag{9}$$

Proof. The consequent fuzzy set of each FAM rule equals one of the fuzzy-set values shown in Figure 2. We assume each fuzzy set includes at least one unity value, $m_A(x) = 1$. Define I_i and c_i as the respective area and centroid of the ith FAM rule's consequent set L_i:

$$I_i = \int m_{L_i}(y)dy , \tag{10}$$

$$c_i = \frac{\int y\, m_{L_i}(y)dy}{\int m_{L_i}(y)dy}$$

$$= \frac{\int y\, m_{L_i}(y)dy}{I_i} ,$$

substituting from (10). Hence

$$\int y\, m_{L_i}(y)dy = c_i\, I_i . \tag{11}$$

Using (3), the result of correlation-product inference, we get

$$\int y\, m_{O_i}(y)dy = \int y\, w_i\, m_{L_i}(y)dy$$

$$= w_i \int y\, m_{L_i}(y)dy$$

$$= w_i c_i I_i \quad , \tag{12}$$

substituting from (11). Similarly,

$$\int m_{O_i}(y)\, dy = \int w_i\, m_{L_i}(y)dy$$

$$= w_i I_i \quad , \tag{13}$$

substituting from (10).

We can use (12) and (13) to derive a discrete expression equivalent to (7):

$$\int y\, m_O(y)dy = \int y[\sum_{i=1}^{N} m_{O_i}(y)]dy \quad \text{substituting from (6)} \quad ,$$

$$= \sum_i \int y\, m_{O_i}(y)\, dy$$

$$= \sum_i w_i c_i I_i \quad , \tag{14}$$

from (12). Similarly,

$$\int m_O(y)dy = \int \sum_{i=1}^{N} m_{O_i}(y)dy$$

$$= \sum_i \int m_{O_i}(y)dy$$

$$= \sum_i w_i I_i \quad , \tag{15}$$

from (13). Substituting (14) and (15) into (7), we derive a new form for the centroid:

$$v_k = \frac{\sum_{i=1}^{N} w_i c_i I_i}{\sum_{i=1}^{N} w_i I_i} \quad , \tag{16}$$

which is equivalent to (9). Each summand in each summation of (16) depends on only a single FAM rule. So we can compute the global output centroid from local FAM-rule centroids. Q.E.D.

Theorem 2: If the 7 library fuzzy sets are symmetric and unimodal (in the trapezoidal sense) and we use correlation-product inference, then we can compute the centroid v_k from only 7 samples of the combined output fuzzy set O:

$$v_k = \frac{\sum_{j=1}^{7} m_O(y_j)\ y_j\ J_j}{\sum_{j=1}^{7} m_O(y_j)\ J_j} \quad . \tag{17}$$

The 7 sample points are the centroids of the output fuzzy-set values.

Proof. Define $\bar{O}_i$ as a fit vector of length 7, where the fit value corresponding to the ith consequent set has the value w_i, and the other entries equal zero. If all the fuzzy sets are symmetric and unimodal, then the jth fit value of $\bar{O}_i$ is a sample of m_{O_i} at the centroid of the jth fuzzy set. The combined output fit vector is

$$\bar{O} = \sum_{i=1}^{N} \bar{O}_i \quad . \tag{18}$$

Since

$$m_O(y) = \sum_{i=1}^{N} m_{O_i}(y) \quad ,$$

the jth fit value of $\bar{O}$ is a sample of m_O at the centroid of the jth fuzzy set. Equivalently, the jth fit value of $\bar{O}$ equals the sum of the output activations w_i from the FAM rules with

consequent fuzzy sets equal to the jth library fuzzy-set value.

Define the reduced universe of discourse as $Y = \{y_1, \ldots, y_7\}$ such that y_j equals the centroid of the jth output fuzzy set. In vector form

$$\begin{aligned} Y &= (y_1, \ldots, y_7) \\ &= (-6, \ -4, \ -2, \ 0, \ 2, \ 4, \ 6) \end{aligned}$$

for the library of fuzzy sets in Figure 2. Also define the diagonal matrix

$$J = \text{diag}(J_1, \ldots, J_7) \quad , \tag{19}$$

where J_j denotes the area of the jth fuzzy-set value. If the ith FAM rule's consequent fuzzy set equals the jth fuzzy-set value, then the jth fit value of $\bar{O}$ increases by w_i, $c_i = y_j$, and $I_i = J_j$. So

$$\bar{O}JY^T = \sum_{j=1}^{7} m_O(y_j) y_j J_j = \sum_{i=1}^{N} w_i c_i I_i \quad . \tag{20}$$

Also,

$$\bar{O}J\mathbf{1}^T = \sum_{j=1}^{7} m_O(y_j) J_j = \sum_{i=1}^{N} w_i I_i \quad , \tag{21}$$

where $\mathbf{1} = (1, \ldots, 1)$. Substituting (20) and (21) into (16) gives

$$v_k = \frac{\sum_{j=1}^{7} m_O(y_j) \ y_j \ J_j}{\sum_{j=1}^{7} m_O(y_j) \ J_j} \quad , \tag{22}$$

which is equivalent to (17). Therefore, (22) gives a simpler, but equivalent form of the centroid (7) if all the fuzzy sets are symmetric and unimodal, and if we use correlation-product inference to form the output fuzzy sets O_i. Q.E.D.

Consider a fuzzy controller with the fuzzy sets defined in Figure 2, and 7 FAM rules with the following outputs:

i	w_i	Consequent
1	0.0	MP
2	0.2	SP
3	1.0	ZE
4	0.4	SN
5	0.1	SP
6	0.8	ZE
7	0.6	SN

Figure 5 shows the combined output fuzzy set O, with the SN, ZE, and SP components displayed with dotted lines. Using (7) we get a velocity output of -0.452. Alternatively, the combined output fit vector $\bar{O}$ equals (0, 0, 1.0, 1.8, 0.3, 0, 0). From (22) we get

$$v_k = \frac{-2 \times 1 + 0 \times 1.8 + 2 \times 0.3}{1 + 1.8 + 0.3} = -0.452 \quad .$$

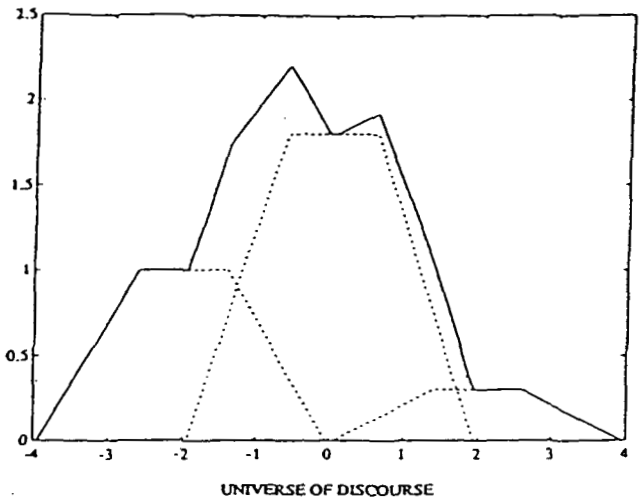


FIGURE 5 Output fuzzy set O.

Fuzzy Controller Implementation

A FAM bank or "rulebase" of FAM rules defines the fuzzy controller. Each FAM rule associates one consequent fuzzy set with three antecedent fuzzy-set conjuncts.

Suppose the ith FAM rule is $(MP, SN, ZE; SP)$. Suppose the inputs at time k are $e_k = 2.6$, $\dot{e}_k = -2.0$, and $v_{k-1} = 1.8$. Then

$$\begin{aligned} w_i &= \min(m_{MP}(e_k),\ m_{SN}(\dot{e}_k),\ m_{ZE}(v_{k-1})) \\ &= \min(.4,\ 1,\ .1) \\ &= .1\ . \end{aligned}$$

If all the fuzzy sets have the same shape, then they correspond to shifted versions of a

single fuzzy set ZE:

$$m_{SP}(y) \;=\; m_{ze}(y-2) \quad .$$

Define e^i, $\dot{e}^i$, and v^i as the centroids of the corresponding antecedent fuzzy sets in the example above. So $e^i = 4$, $\dot{e}^i = -2$, and $v^i = 0$. Then the output activation equals

$$\begin{aligned} w_i &= \min(m_{ZE}(e_k - e^i),\ m_{ZE}(\dot{e}_k - \dot{e}^i),\ m_{ZE}(v_{k-1} - v^i)) \\ &= \min(m_{ZE}(-1.4),\ m_{ZE}(0),\ m_{ZE}(1.8)) \\ &= \min(.4,\ 1,\ .1) \\ &= .1 \ , \end{aligned}$$

as computed above. Figure 6 schematizes such a FAM rule when presented with crisp inputs.

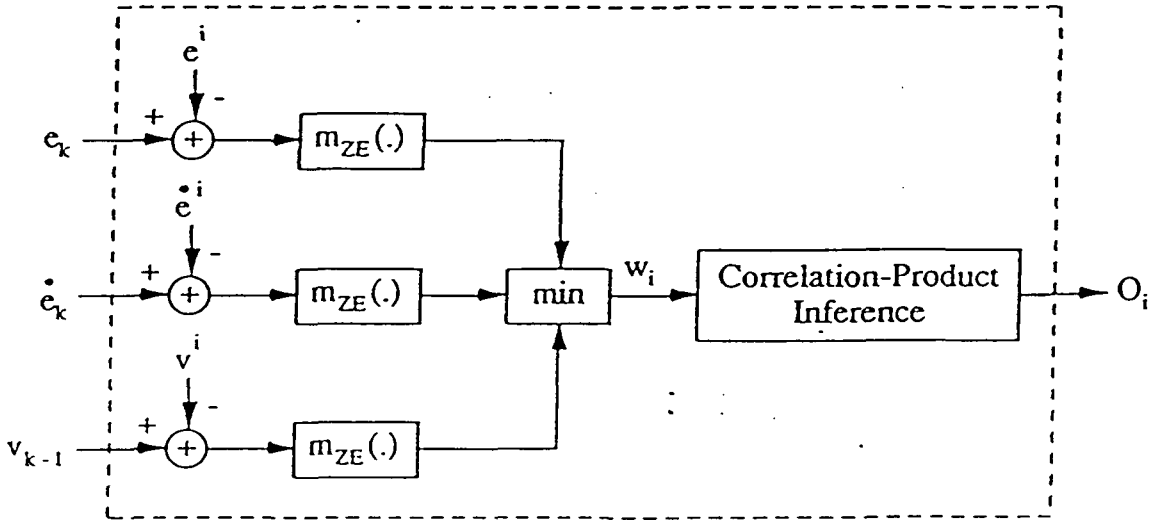


FIGURE 6 Algorithmic structure of a FAM rule for the special case of identically-shaped fuzzy sets and correlation-product inference.

The output fuzzy set O_i in Figure 6 equals the fuzzy set ZE scaled by w_i and shifted by c_i:

$$m_{O_i}(y) = w_i m_{ZE}(y - c_i) \quad . \tag{23}$$

Figure 7 illustrates O_i.

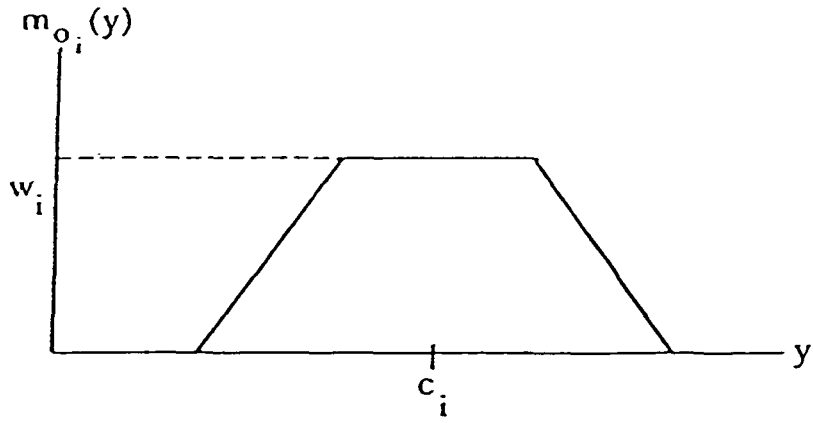


FIGURE 7 Trapezoidal output fuzzy set O_i.

The fuzzy control system activates a bank of FAM rules operated in parallel, as shown in Figure 8. The system sums the output fuzzy sets to form the total output set O, which the system converts to a "defuzzified" scalar output by computing its fuzzy centroid.

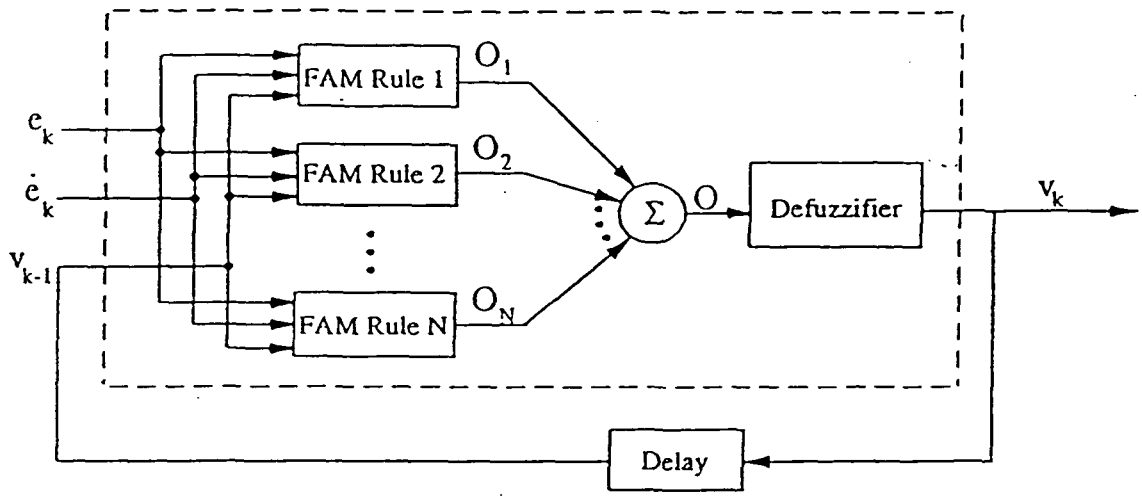


FIGURE 8 Fuzzy control system as a parallel FAM bank with centroidal output.

KALMAN FILTER CONTROLLER

We designed a one-dimensional **Kalman filter** to act as an alternative controller. The *state* and *measurement* equations take the general form

$$\begin{aligned} x_{k+1} &= \Phi_{k+1,k}\, x_k + \Gamma_{k+1,k}\, w_k + \Psi_{k+1,k}\, u_k \quad , \\ z_k &= H_k\, x_k + V_k \quad , \end{aligned} \tag{24}$$

where V_k denotes Gaussian white noise with covariance matrix R_k. If V_k is colored noise or if $R_k = 0$, then the filtering-error covariance matrix $P_{k|k}$ becomes singular. The state x_k and the measurements z_k are jointly Gaussian. Mendel [1987] gives details of this model.

Assume the following one-dimensional model:

$$\Phi_{k+1,k} = \Gamma_{k+1,k} = \Psi_{k+1,k} = H_k = 1 \text{ for all } k,$$

$$u_k = e_k + \dot{e}_k \quad . \tag{25}$$

Let x_{k+1} denote the output velocity required at time k to exactly lock onto the target at time $k+1$. So the controller output at time k equals the "predictive" estimate $\hat{x}_{k+1|k} = v_k$. Note that

$$\begin{aligned} e_k &= x_k - \hat{x}_{k|k-1} \\ &= \tilde{x}_{k|k-1} \ , \\ \dot{e}_k &= e_k - e_{k-1} \ . \end{aligned}$$

Substituting (25) into (24), we get the new state equation

$$x_{k+1} = x_k + e_k + \dot{e}_k + w_k \ , \tag{26}$$

where w_k denotes white noise that models target acceleration or other unmodeled effects. The new measurement equation is

$$\begin{aligned} z_k &= x_k + V_k \\ &= \hat{x}_{k|k-1} + \tilde{x}_{k|k-1} + V_k \\ &= \hat{x}_{k|k-1} + V_k' \ . \end{aligned} \tag{27}$$

Since we assume $\tilde{x}_{k|k-1}$ and V_k are uncorrelated, the variance of V_k' is

$$\begin{aligned} R_k' &= E[V_k'^2] \\ &= E[\tilde{x}_{k|k-1}^2] + E[V_k^2] \end{aligned} \tag{28}$$

$$= P_{k|k-1} + R_k \quad .$$

The general form of the recursive Kalman filter equations is

$$\begin{aligned}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k[z_k - H_k \hat{x}_{k|k-1}] \quad , \\
K_k &= P_{k|k-1} H_k^T [H_k P_{k|k-1} H_k^T + R_k]^{-1} \quad , \\
\hat{x}_{k+1|k} &= \Phi_{k+1,k} \hat{x}_{k|k} + \Psi_{k+1,k} u_k \quad , \\
P_{k|k-1} &= \Phi_{k,k-1} P_{k-1|k-1} \Phi_{k,k-1}^T + \Gamma_{k,k-1} Q_{k-1} \Gamma_{k,k-1}^T \quad , \\
P_{k|k} &= [I - K_k H_k] P_{k|k-1} \quad ,
\end{aligned} \tag{29}$$

where $Q_k = Var(w_k) = E[w_k w_k^T]$. Substituting (25), (27), (28) and the definition of v_k into (29), we get the following one-dimensional Kalman filter:

$$\begin{aligned}
\hat{x}_{k|k} &= v_{k-1} + K_k V_k' \quad , \\
K_k &= \frac{P_{k|k-1}}{R_k'} \quad , \\
v_k &= \hat{x}_{k|k} + e_k + \dot{e}_k \quad , \\
P_{k|k-1} &= P_{k-1|k-1} + Q_{k-1} \quad , \\
P_{k|k} &= [1 - K_k] P_{k|k-1} \quad .
\end{aligned} \tag{30}$$

Unlike the fuzzy controller, this Kalman filter does not automatically restrict the output v_k to a usable range. We must apply a threshold immediately after the controller. To remain consistent with the fuzzy controller, we set the following thresholds:

$$\begin{aligned}
|v_k| &\leq 9 \text{ degrees azimuth} \quad , \\
|v_k| &\leq 4.5 \text{ degrees elevation.}
\end{aligned}$$

Fuzzy and Kalman Filter Control Surfaces

Each control system maps inputs to outputs. Geometrically, these input-output transformations define *control surfaces*. The control surfaces are sheets in the input space (since the output velocity v_k is a scalar). Three inputs and one output give rise to a four-dimensional control surface, which we cannot plot. Instead, for each controller we can plot a family of three-dimensional control surfaces indexed by constant values of the fourth variable, the error e_k, say. Then each control surface corresponds to a different value of the error e_k.

The fuzzy control surface characterizes the fuzzy system's fuzzy-set value definitions and its bank of FAM rules. Different sets of FAM rules yield different fuzzy controllers, and hence different control surfaces. Figure 9 shows a cross section of the FAM bank when $e_k = ZE$. Each entry in this linguistic matrix represents one FAM rule with $e_k = ZE$ as the first antecedent term.

$\dot{e}_k$ \ v_{k-1}	LN	MN	SN	ZE	SP	MP	LP
LN	LN	LN	LN	LN	MN	SN	ZE
MN	LN	LN	LN	MN	SN	ZE	SP
SN	LN	LN	MN	SN	ZE	SP	MP
ZE	LN	MN	SN	ZE	SP	MP	LP
SP	MN	SN	ZE	SP	MP	LP	LP
MP	SN	ZE	SP	MP	LP	LP	LP
LP	ZE	SP	MP	LP	LP	LP	LP

FIGURE 9 $e_k = ZE$ cross section of the fuzzy control system's FAM bank. Each entry represents one FAM rule with $e_k = ZE$ as the first antecedent term.

The shaded FAM rule is "IF $e_k = ZE$ AND $\dot{e}_k = SP$ AND $v_{k-1} = SN$, THEN $v_k = ZE$," abbreviated as $(ZE, SP, SN; ZE)$. Note the ordinal anti-symmetry of this FAM-bank matrix. The six other cross-section FAM-bank matrices are similar. We can eliminate many FAM rule entries without greatly perturbing the fuzzy controller's behavior.

The entire FAM bank—including cross sections for e_k equal to each of the seven fuzzy-set values LN, MN, SN, ZE, SP, MP, and LP—determines how the system maps input fuzzy sets to output fuzzy sets. The fuzzy set membership functions shown in Figure 2 determine the degree to which each crisp input value belongs to each fuzzy-set value. So both the fuzzy-set value definitions and the FAM bank determine the defuzzified output v_k for any set of crisp input values e_k, $\dot{e}_k$, and v_{k-1}.

Figure 10 shows the control surface of the fuzzy controller for $e_k = 0$. We plotted the control output v_k against $\dot{e}_k$ and v_{k-1}. Since we use the same algorithm for tracking in azimuth and elevation, the control surfaces for the two dimensions differ in scale only by a factor of two.

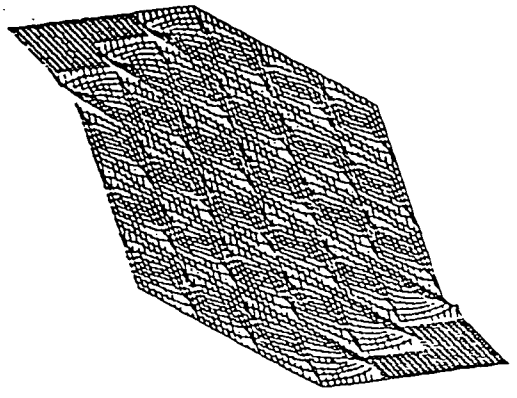


FIGURE 10 Control surface of the fuzzy controller for constant error $e_k = 0$. We plotted the control output v_k against $\dot{e}_k$ and v_{k-1} along the respective west and south borders.

The Kalman filter has a random control surface that depends on a time-varying pa-

rameter. From (30) we see that

$$v_k = \hat{x}_{k|k} + e_k + \dot{e}_k \ ,$$

$$\hat{x}_{k|k} = v_{k-1} + K_k V_k' \ ,$$

where V_k' denotes white noise with variance given by (28). Combining these two equations gives the equation for the random control surface:

$$v_k = v_{k-1} + e_k + \dot{e}_k + K_k V_k' \ . \tag{31}$$

At time k the noise term $K_k V_k'$ has variance

$$\sigma_k^2 = K_k^2 \, R_k' \tag{32}$$

$$= \frac{P_{k|k-1}^2}{R_k'} \text{ upon substituting from (30)} \ ,$$

$$= \frac{P_{k|k-1}^2}{P_{k|k-1} + R_k},$$

substituting from (28). Combining (31) and (32) gives a new control surface equation:

$$v_k = v_{k-1} + e_k + \dot{e}_k + \sigma_k V_k'' \ , \tag{33}$$

where V_k'' denotes unit-variance Gaussian noise. So the Kalman filter's control output equals the sum of the three input variables plus additive Gaussian noise with time-dependent variance σ_k^2. For constant error e_k, we can interpret (33) as a smooth control surface in R^3 defined by

$$v_k = v_{k-1} + e_k + \dot{e}_k \ ,$$

and perturbed at time k by Gaussian noise with variance σ_k^2.

In our simulations the standard deviation σ_k converged after only a few iterations. We used unity initial conditions: $P_{o|o} = R_k = 1$ for all k.

Table 1 lists the convergence rates and steady-state values of σ_k for three different values of the variance $Var(w)$ of the white-noise, unmodeled-effects process w_k. For $Var(w) = 0$, σ_k decreases rapidly at first—$\sigma_8 = .10$, $\sigma_{17} = .05$—but does not attain its steady-state value of zero within 100 iterations.

$Var(w)$	Steady-state value of σ_k	Number of iterations required for convergence
1.00	0.79	2
0.25	0.46	4
0.05	0.22	9

TABLE 1 Convergence rates and steady-state values of σ_k for different values of the variance $Var(w)$ of the white-noise, unmodeled-effects process w_k.

Figure 11 shows four realizations of the Kalman filter's random control surface for $e_k = 0$, each at a time k when σ_k has converged to its steady-state value. For each plot, we used output thresholds and initial variances for the azimuth case: $|v_k| \leq 9.0$, $R_k = P_{o|o} = 1.0$. As with the fuzzy controller, elevation control surfaces equal scaled versions of the corresponding azimuth control surfaces.

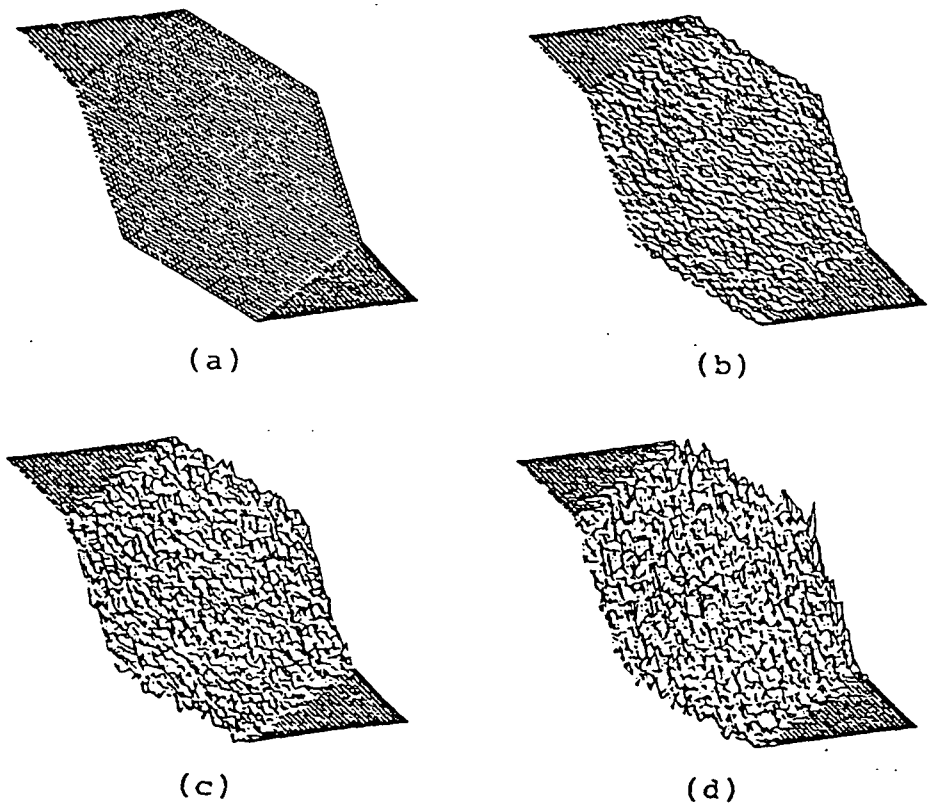


FIGURE 11 Realizations of the Kalman filter's random control surface with $e_k = 0$ for different values of the variance $Var(w)$ and steady-state values of the standard deviation σ_k: (a) $Var(w) = \sigma_k = 0$, (b) $Var(w) = .05$, $\sigma_k = .22$; (c) $Var(w) = .25$, $\sigma_k = .46$; (d) $Var(w) = 1.0$, $\sigma_k = .79$.

SIMULATION RESULTS

Our target-tracking simulations model several realworld scenarios. Suppose we have mounted the target tracking system on the side of a vehicle, aircraft, or ship. The system tracks a missile that cuts across the detection range on a straight flight path. The target maintains a constant speed of 1,870 miles-per-hour and comes within 3.5 miles of the

platform at c'osest approach. The platform can scan from 0 to 180 degrees in azimuth at a maximum rate of 36 degrees-per-second, and from 0 (vertical) to 90 degrees in elevation at a maximum rate of 18 degrees-per-second. The sampling interval is 1/4 of a second. The gain of the fuzzy controller equals 0.9. So the maximum error considered is 10 degrees azimuth and 5 degrees elevation. We threshold all error values above this level.

Figure 12 demonstrates the best performance of the fuzzy controller for a simulated scenario. The solid lines indicate target position. The dotted lines indicate platform position. To achieve this performance, we calibrated the three design parameters—upper and lower trapezoid bases and the gain. Figures 13 and 14 show examples of uncalibrated systems. Too much overlap causes excessive overshoot. Too little overlap causes lead or lag for several consecutive time intervals. A gain of 0.9 suffices for most scenarios. We can fine-tune the fuzzy control system by altering the percentage overlap between adjacent fuzzy sets.

Figure 15 demonstrates the best performance of the Kalman-filter controller for the same scenario used to test the fuzzy controller. For simplicity, $R_k = P_{0|0}$ for all values of k. For this study we chose the values 1.0 (unit variance) for azimuth and 0.25 for elevation. This 1/4 ratio reflects the difference in scanning range. We set Q_k to 0 for optimal performance. Figure 16 shows the Kalman-filter controller's performance when $Q_k = 1.0$ azimuth, 0.25 elevation.

Sensitivity Analysis

We compared the uncertainty sensitivity of the fuzzy and Kalman-filter control systems. Under normal operating conditions, when the FAM bank contains all fuzzy control rules, and when the unmodeled-effects noise variance $Var(w)$ is small, the controllers perform almost identically. Under more uncertain conditions their performance differs. The Kalman filter's state equation (26) contains the noise term w_k whose variance we must assume. When $Var(w)$ increases, the state equation becomes more uncertain. The fuzzy control

FAM rules depend implicitly on this same equation, but without the noise term. Instead, the fuzziness of the FAM rules accounts for the system uncertainty. This suggests that we can increase the uncertainty of the implicit state equation by omitting randomly selected FAM rules. Figures 17 and 18 show the effect on the *root-mean-squared error* (RMSE) in degrees when we omit FAM rules and increase $Var(w)$. Each data point averages ten runs.

The controllers behave differently as uncertainty increases. The RMSE of the fuzzy controller increases little until we omit nearly sixty percent of the FAM rules. The RMSE of the Kalman filter increases steeply for small values of $Var(w)$, then gradually levels off.

We also tested the fuzzy controller's robustness by "sabotaging" the most vulnerable FAM rule. This could reflect lack of accurate expertise, or a highly unstructured problem. Changing the consequent of the steady-state FAM rule $(ZE, ZE, ZE; ZE)$ to LP gives the following nonsensical FAM rule:

IF the platform points directly at the target
    AND both the target and the platform are stationary,
THEN turn in the positive direction with maximum velocity.

Figure 19 shows the fuzzy system's performance when this sabotage FAM rule replaces the steady-state FAM rule. When the sabotage FAM rule activates, the system quickly adjusts to decrease the error again. The fuzzy system is piecewise stable.

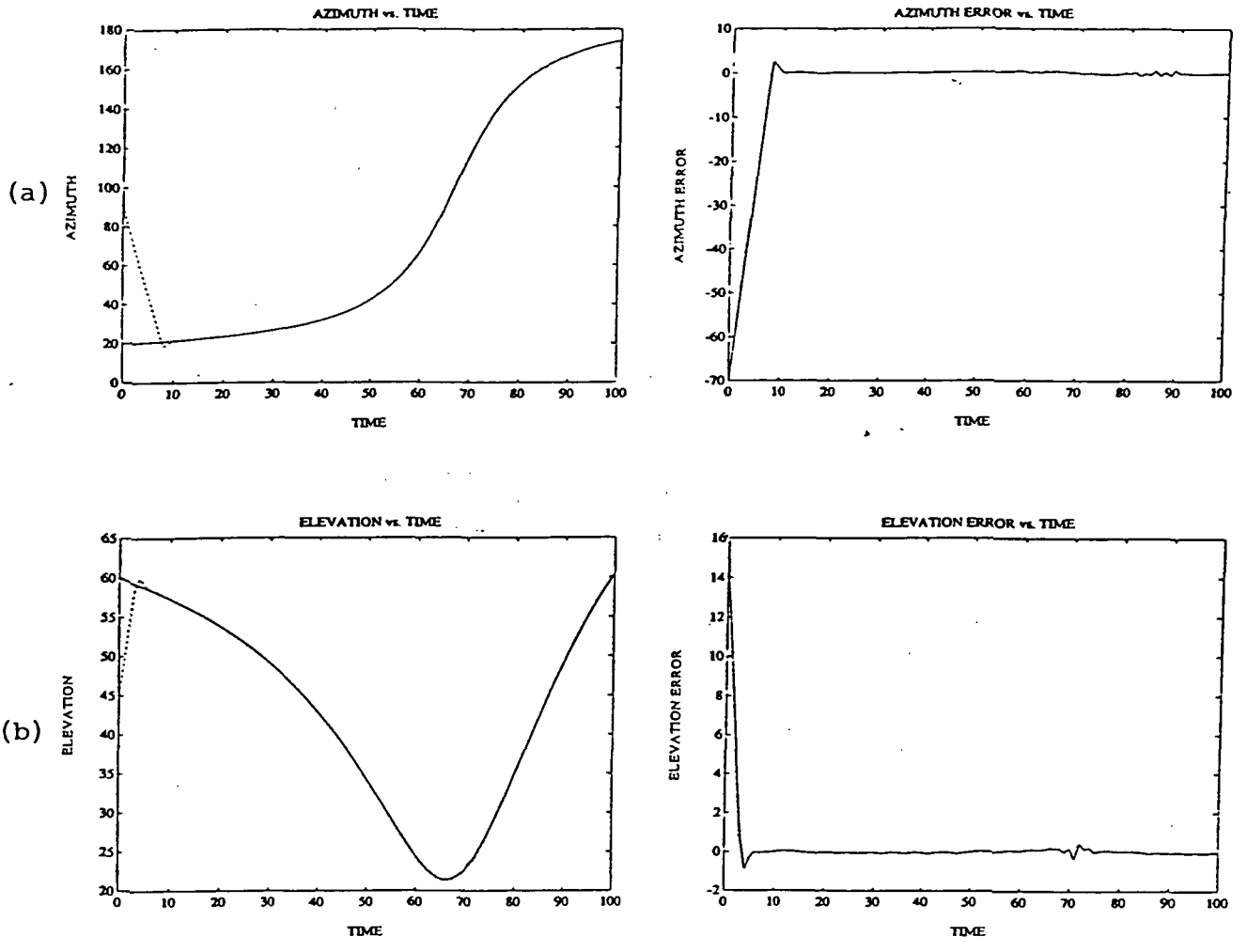


FIGURE 12 Best performance of the fuzzy controller: (a) azimuth position and error, (b) elevation position and error. Fuzzy set overlap is 26.2%.

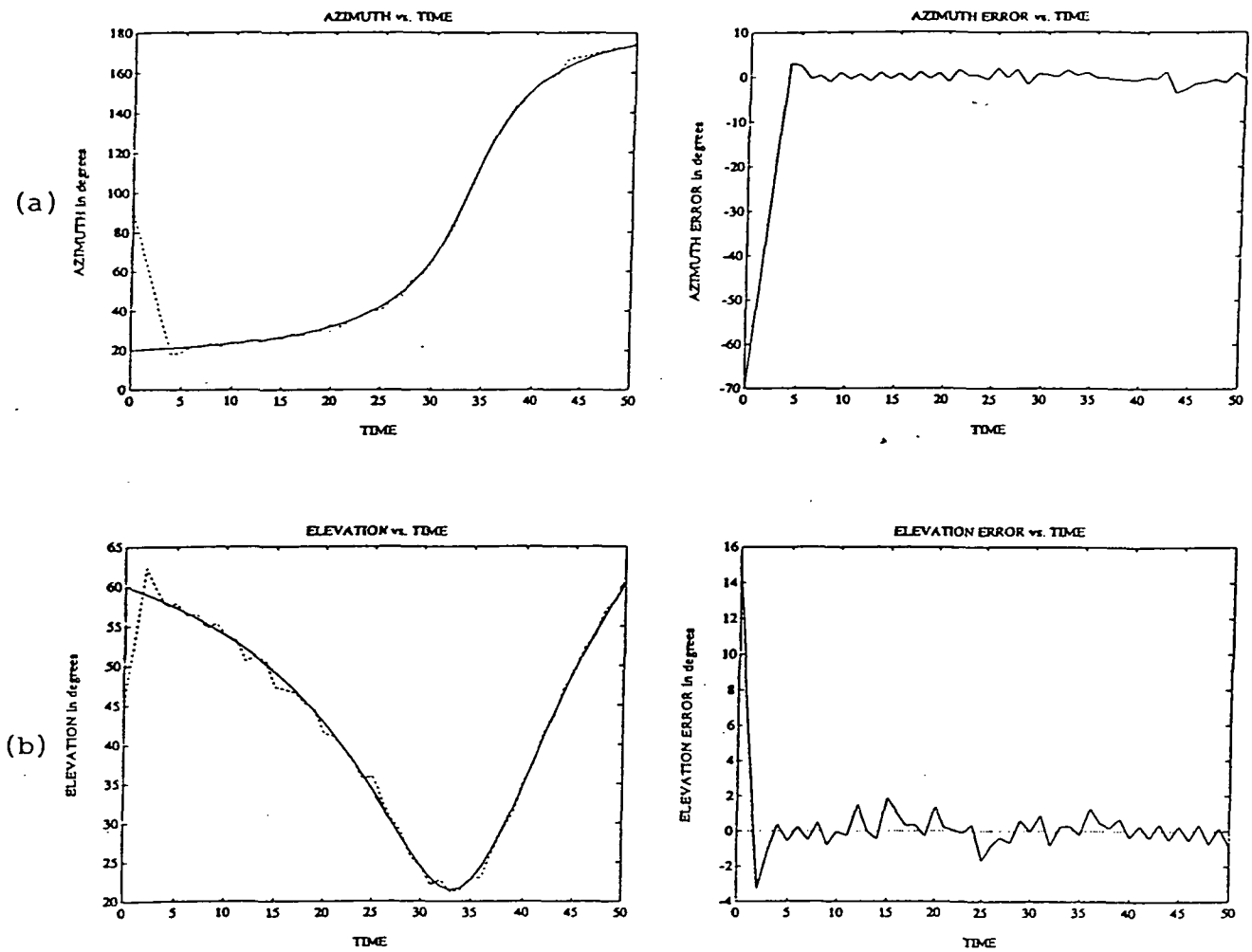


FIGURE 13 Uncalibrated fuzzy controller: (a) azimuth position and error, (b) elevation position and error. Fuzzy set overlap equals 33.3%. Too much overlap causes excessive overshoot.

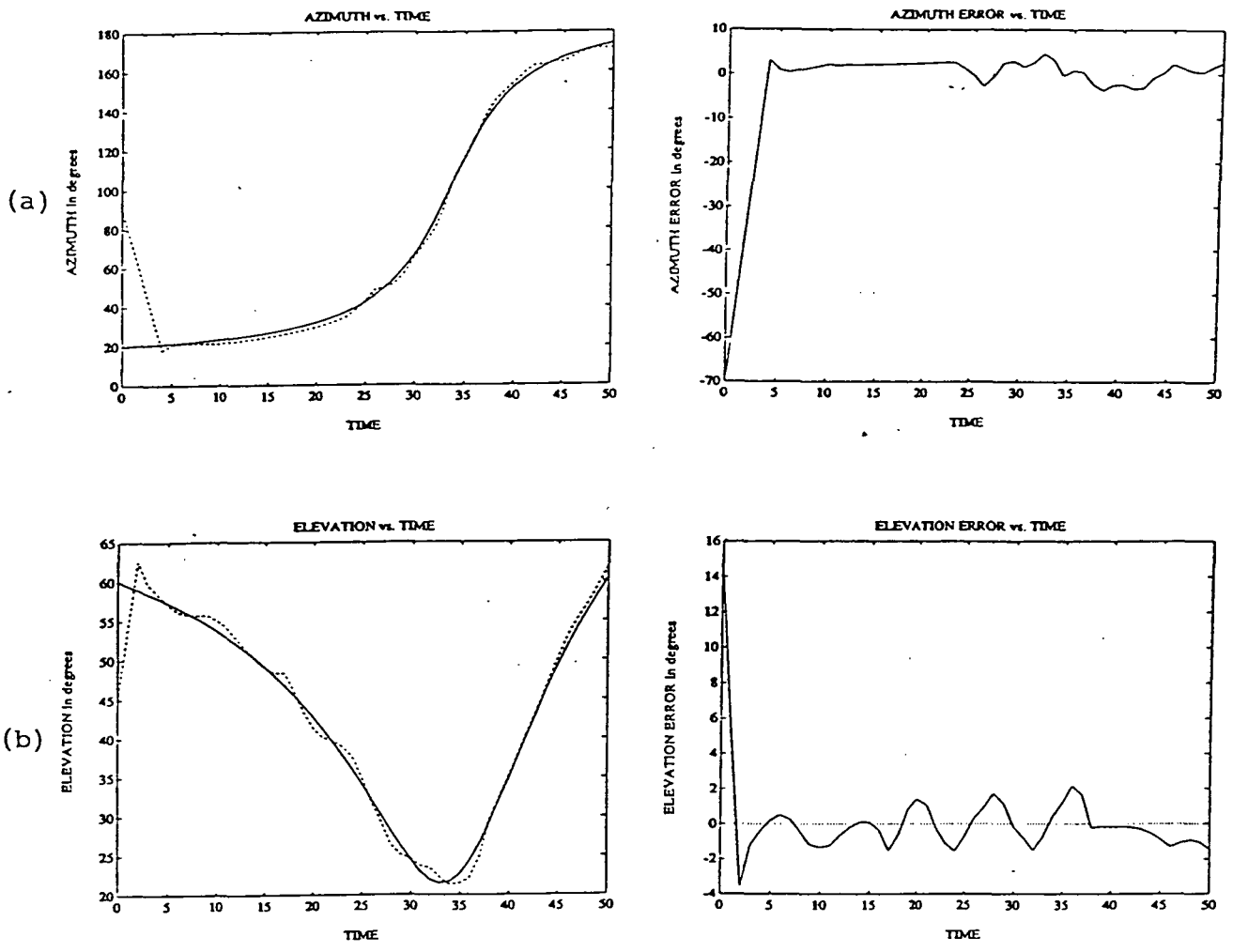


FIGURE 14 Uncalibrated fuzzy controller: (a) azimuth position and error, (b) elevation position and error. Fuzzy set overlap equals 12.5%. Too little overlap causes lead or lag for several consecutive time intervals.

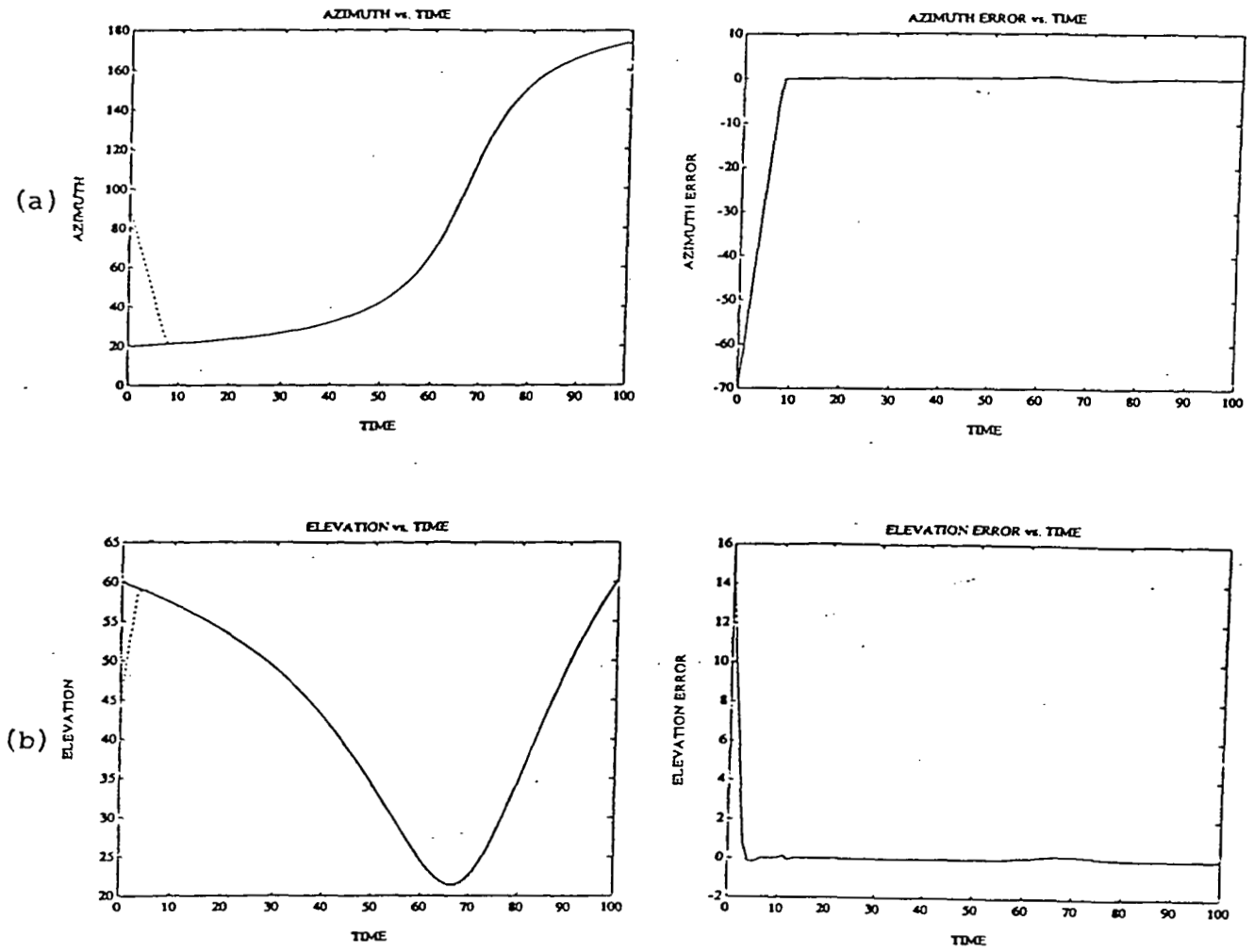


FIGURE 15 Kalman filter controller with unmodeled-effects noise variance $Var(w) = 0$: (a) azimuth position and error, (b) elevation position and error.

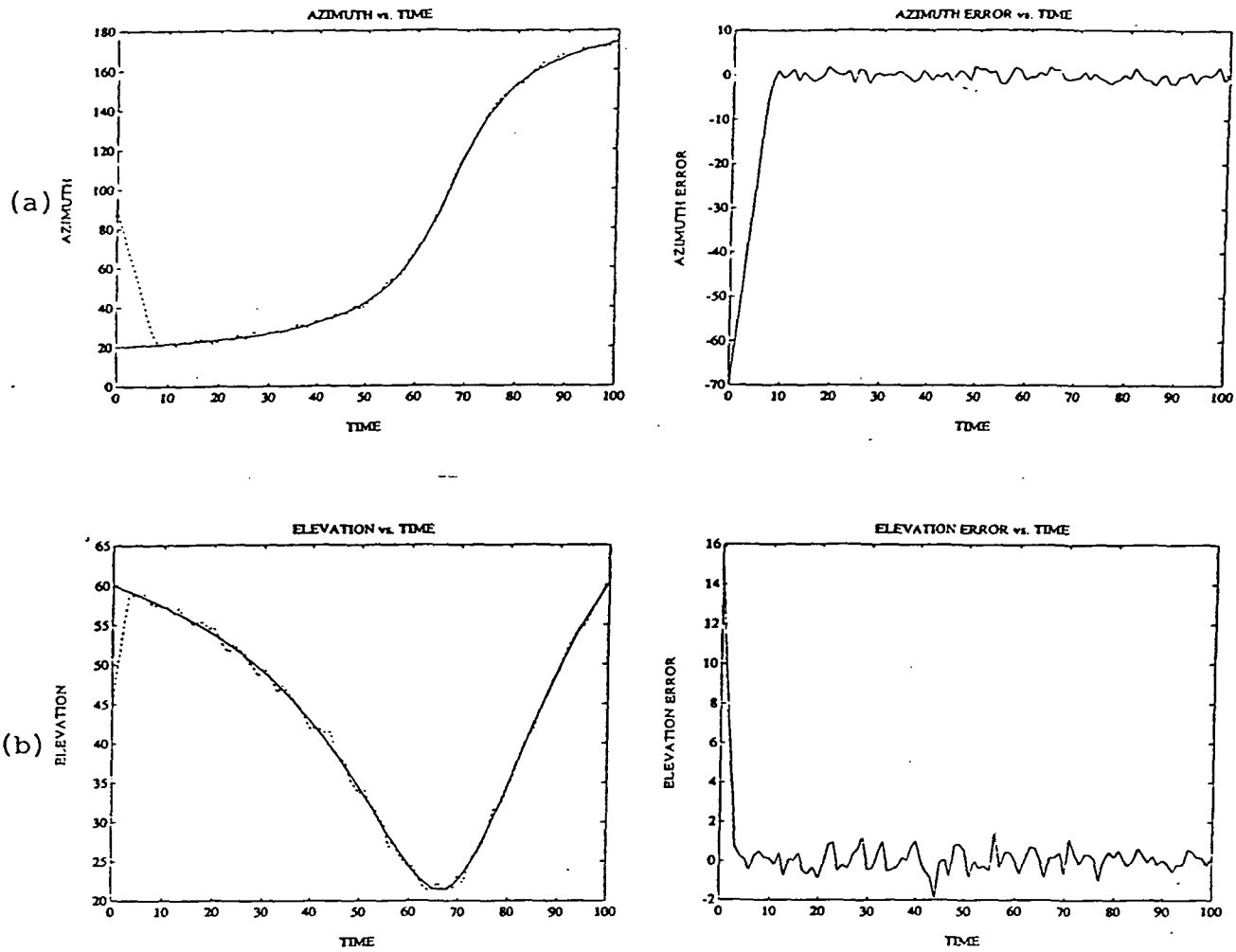


FIGURE 16 Kalman filter controller with $Var(w) = 1.0$ azimuth, 0.25 elevation: (a) azimuth position and error, (b) elevation position and error.

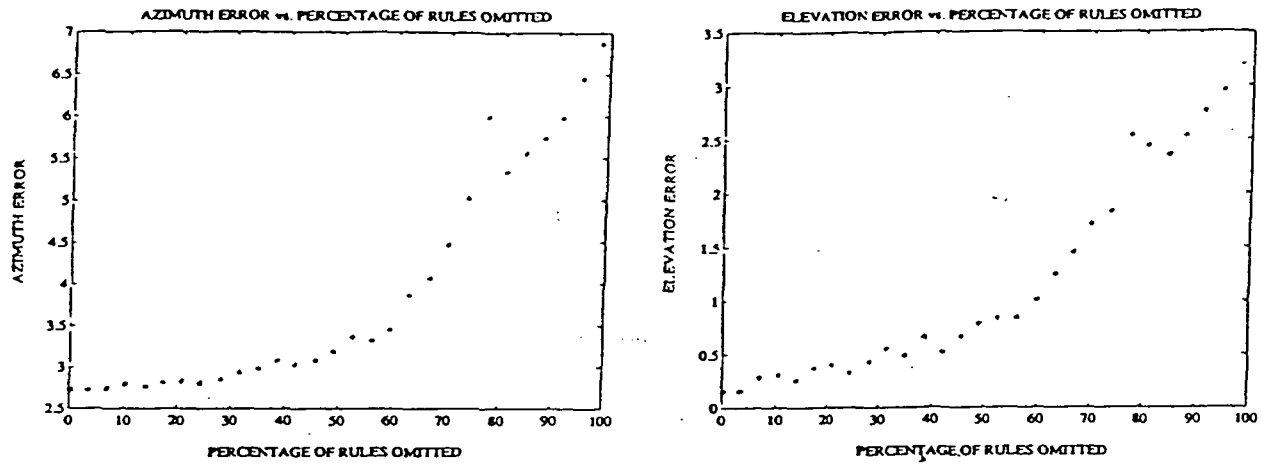


FIGURE 17 Root-mean-squared error of the fuzzy controller with randomly selected FAM rules omitted.

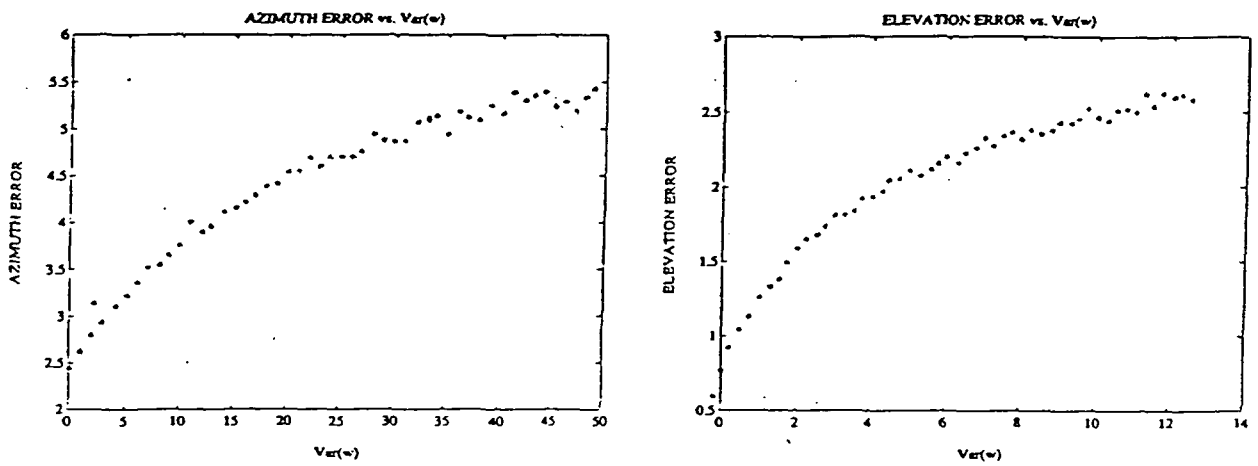


FIGURE 18 Root-mean-squared error of the Kalman filter controller as $Var(w)$ varies.

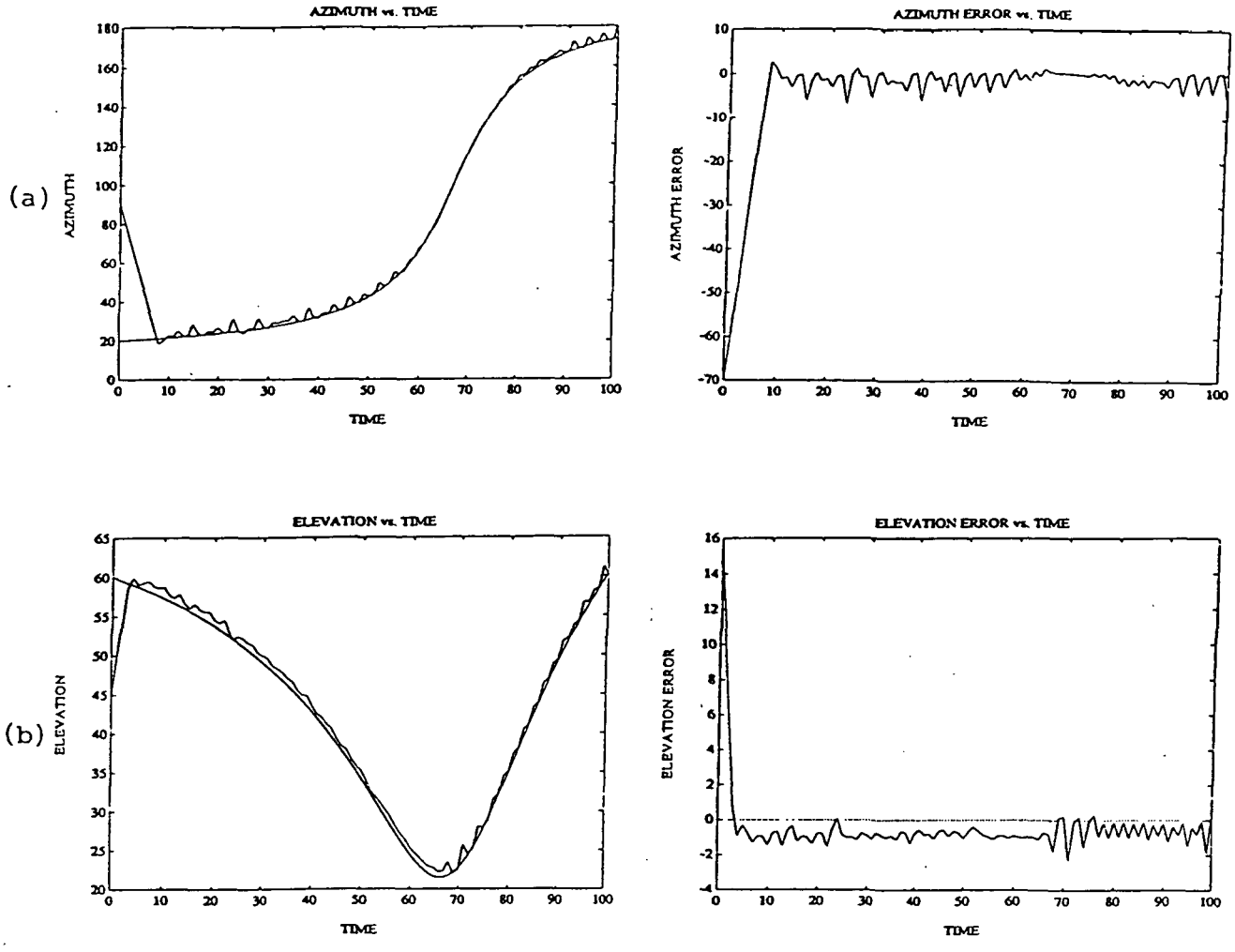


FIGURE 19 Fuzzy controller with a "sabotage" FAM rule: (a) azimuth position and error, (b) elevation position and error. The sabotage rule $(ZE, ZE, ZE; LP)$ replaces the steady-state FAM rule $(ZE, ZE, ZE; ZE)$. The system quickly adjusts each time the sabotage rule activates.

Adaptive FAM (AFAM)

We used unsupervised product-space clustering [Kosko, 1990a] to train an adaptive FAM (AFAM) fuzzy controller. Differential competitive learning (DCL) adaptively clustered input-output pairs. The Appendix describes product-space clustering with DCL. For this study, there were four input neurons in F_x. A manually-designed FAM bank and 80 random target trajectories generated 19,236 training vectors. Each product-space training vector $(e_k,\ \dot{e}_k,\ v_{k-1},\ v_k)$ defined a point in R^4.

Symmetry allowed us to reflect about the origin all sample vectors with negative errors e_k. We then trained 3,000 synaptic quantization vectors ($p = 3,000$) in the positive error half-space. For each sample vector, we defined the 10 closest synaptic vectors as "winners" ($N = 10$). The matrix W of F_Y within-field synaptic connection strengths had diagonal elements $w_{ii} = 2.9$, off-diagonal elements $w_{ij} = -0.1$. After training, we reflected the 3,000 synaptic quantization vectors about the origin to give 6,000 trained synaptic vectors.

The product-space FAM cells uniformly partitioned the four-dimensional product space. Each FAM cell represented a single FAM rule. The four fuzzy variables could assume only the 7 fuzzy-set values LN, MN, SN, ZE, SP, MP, and LP. So the product space contained $7^4 = 2401$ FAM cells.

At the end of the DCL training period, we defined a FAM cell as occupied only if it contained at least one synaptic vector. For some combinations of antecedent fuzzy sets, synaptic vectors occupied more than one FAM cell with different consequent fuzzy sets. In these cases we computed the centroid of the consequent fuzzy sets weighted by the number of synaptic vectors in their FAM cells. We chose the consequent fuzzy set as that output fuzzy-set value with centroid nearest the weighted centroid value. We ignored other FAM rules with the same antecedents but different consequent fuzzy sets.

Figure 20(a) shows the $e_k = ZE$ cross section of the original FAM bank used to generate the training samples. Figure 20(b) shows the same cross section of the DCL-estimated FAM bank. Figure 21 shows the original and DCL-estimated control surfaces for constant error $e_k = 0$.

for constant error $e_k = 0$.

The regions where the two control surfaces differ correspond to infrequent high-velocity situations. So the original and DCL-estimated control surfaces yield similar results. Table 2 compares the controllers' root-mean-squared errors for 10 randomly-selected target trajectories.

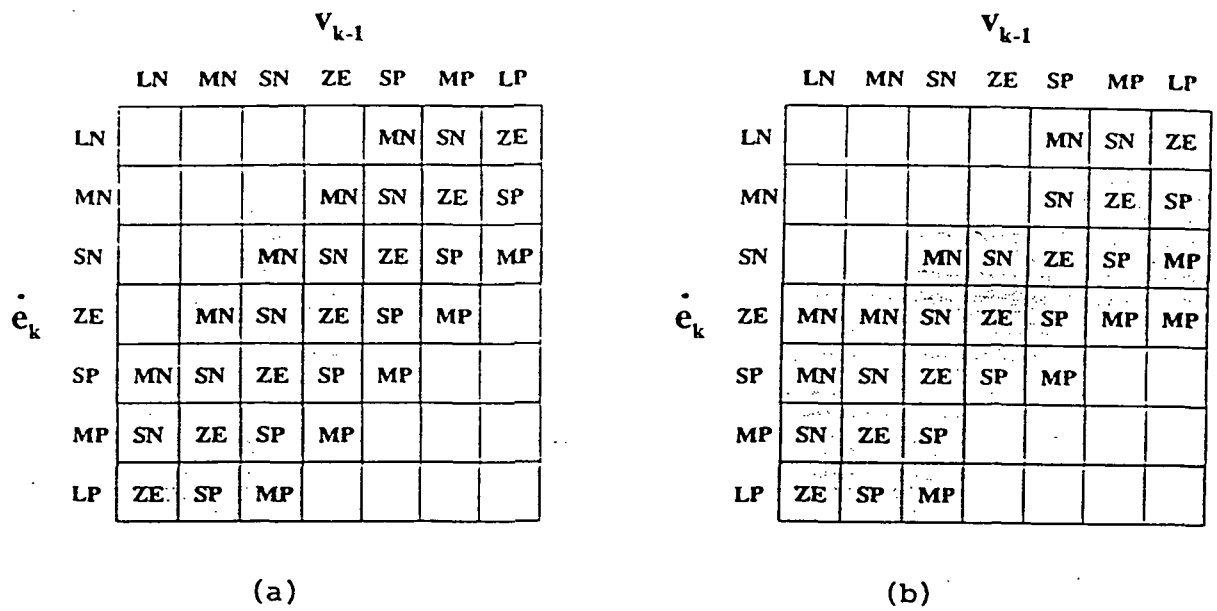
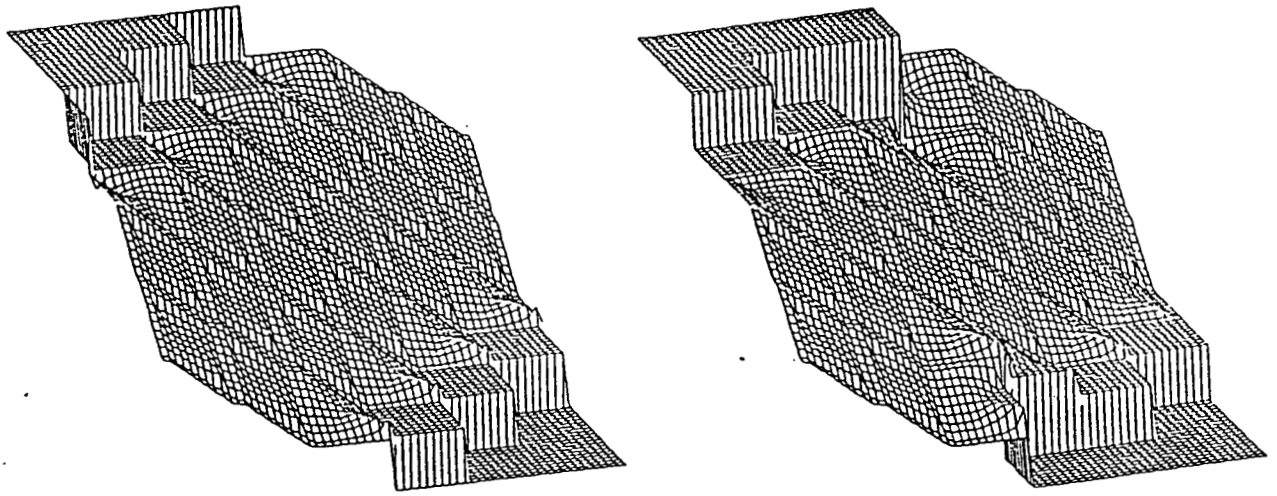


FIGURE 20 Cross sections of the original and DCL- estimated FAM banks when $e_k = ZE$: (a) original, (b) DCL- estimated.



(a) (b)

FIGURE 21 Control surfaces for constant error $e_k = 0$: (a) original, (b) DCL-estimated.

Trajectory	*Azimuth*		*Elevation*	
Number	Original	Estimated	Original	Estimated
1	2.33	2.33	3.31	3.37
2	4.14	4.14	3.03	2.89
3	6.11	6.11	3.69	3.68
4	3.83	3.83	3.32	3.30
5	4.02	4.02	3.11	3.10
6	2.84	2.84	1.20	1.21
7	3.22	3.22	3.04	2.98
8	0.75	0.74	2.00	2.00
9	9.28	9.27	5.50	5.41
10	1.81	1.81	2.29	2.29
Average	3.83	3.83	3.05	3.02

TABLE 2 Root-mean-squared errors for 10 randomly-selected target trajectories. The original and DCL-estimated FAM banks yielded similar results since they differed only in regions corresponding to infrequent high-velocity situations.

Conclusion

We developed and compared a fuzzy control system and a Kalman-filter control system for realtime target tracking. The fuzzy system represented uncertainty with continuous or fuzzy sets, with the partial occurence of multiple alternatives. The Kalman-filter system represented uncertainty with the random occurence of an exact alternative. Accordingly, our simulations tested each system's response to a different family of uncertainty environments, one fuzzy and the other random. In general representative training data can "blindly" generate the governing FAM rules.

These simulations suggest that in many cases fuzzy controllers may be a robust, computationally effective alternative to linear Kalman filter, indeed to nonlinear extended Kalman filter, approaches to realtime system control—even when we can accurately articulate an input-output math model.

REFERENCES

Huber, P.J., *Robust Statistics*, Wiley, 1981.

Kong, S.-G., Kosko, B., "Differential Competitive Learning for Centroid Estimation and Phoneme Recognition," *IEEE Transactions of Neural Networks*, to appear, January 1991.

Kosko, B., "Fuzzy Entropy and Conditioning," *Information Sciences*, vol.40, 165-174, 1986.

Kosko, B., *Foundations of Fuzzy Estimation Theory*, Ph.D. dissertation, Department of Electrical Engineering, University of California at Irvine, June 1987; Order number 8801936, University Microfilms International, 300 N. Zeeb Road, Ann Arbor, Michigan 48106.

Kosko, B., *Neural Networks and Fuzzy Systems: A Dynamical System Approach to Machine Intelligence*, Prentice-Hall, 1990.

Kosko, B., "Stochastic Competitive Learning," *Proceedings of the Summer 1990 International Joint Conference on Neural Networks (IJCNN-90)*, vol.II, 215-226, June 1990.

Kosko, B., "Unsupervised Learning in Noise," *IEEE Transactions on Neural Networks*, vol.1, no.1, 44-57, March 1990.

Mendel, J.M., *Lessons in Digital Estimation Theory*, Prentice-Hall, 1987.

Appendix: Product-space Clustering with Differential Competitive Learning

Adaptive Vector Quantization

Product-space clustering [Kosko, 1990a] is a form of stochastic adaptive vector quantization. Adaptive vector quantization (AVQ) systems adaptively quantize pattern clusters in R^n. Stochastic competitive-learning systems are neural AVQ systems. Neurons compete for the activation induced by randomly sampled patterns. The corresponding fan-in vectors adaptively quantize the pattern space R^n. The p synaptic vectors $\mathbf{m}_j$ define the p columns of the synaptic connection matrix M. M interconnects the n input or linear neurons in the input neuronal field F_X to the p competing nonlinear neurons in the output field F_Y. Figure 22 below illustrates the neural network topology.

Learning algorithms estimate the unknown probability density function $p(\mathbf{x})$, which describes the distribution of patterns in R^n. More synaptic vectors arrive at more probable regions. Where sample vectors $\mathbf{x}$ are dense or sparse, synaptic vectors $\mathbf{m}_j$ should be dense or sparse. The local count of synaptic vectors then gives a nonparametric estimate of the volume density $P(V)$ for volume $V \subset R^n$:

$$P(V) = \int_V p(\mathbf{x})d\mathbf{x} \tag{34}$$

$$\approx \frac{\text{Number of } \mathbf{m}_j \in V}{p} \quad . \tag{35}$$

In the extreme case that $V = R^n$, this approximation gives $P(V) = p/p = 1$. For improbable subsets V, $P(V) = 0/p = 0$.

Stochastic Competitive Learning Algorithms

The metaphor of competing neurons reduces to nearest-neighbor classification. The AVQ system compares the current vector random sample $\mathbf{x}(t)$ in Euclidean distance to the p columns of the synaptic connection matrix M, to the p synaptic vectors $\mathbf{m}_1(t), \ldots, \mathbf{m}_p(t)$. If the jth synaptic vector $\mathbf{m}_j(t)$ is closest to $\mathbf{x}(t)$, then the jth output neuron "wins" the competition for activation at time t. In practice we sometimes define the nearest N synaptic vectors as winners. Some scaled form of $\mathbf{x}(t) - \mathbf{m}_j(t)$ updates the nearest or "winning" synaptic vectors. "Losers" remain unchanged: $\mathbf{m}_i(t+1) = \mathbf{m}_i(t)$. Competitive synaptic vectors converge to pattern-class centroids exponentially fast [Kosko, 1990b].

The following three-step process describes the competitive AVQ algorithm, where the third step depends on which learning algorithm updates the winning synaptic vectors.

Competitive AVQ Algorithm

1. Initialize synaptic vectors: $\mathbf{m}_i(0) = \mathbf{x}(i), i = 1, \ldots, p$. Sample-dependent initialization avoids many pathologies that can distort nearest-neighbor learning.

2. For random sample $\mathbf{x}(t)$, find the closest or "winning" synaptic vector $\mathbf{m}_j(t)$:

$$\|\mathbf{m}_j(t) - \mathbf{x}(t)\| = \min_i \|\mathbf{m}_i(t) - \mathbf{x}(t)\| \quad , \tag{36}$$

   where $\|\mathbf{x}\|^2 = x_1^2 + \ldots + x_n^2$ defines the squared Euclidean vector norm of $\mathbf{x}$. We can define the N synaptic vectors closest to $\mathbf{x}$ as "winners."

3. Update the winning synaptic vector(s) $\mathbf{m}_j(t)$ with an appropriate learning algorithm.

Differential Competitive Learning (DCL)

Differential competitive "synapses" learn only if the competing "neuron" changes its competitive status [Kosko, 1990c]:

$$\dot{m}_{ij} = \dot{S}_j(y_j)[S_i(x_i) - m_{ij}] \quad , \tag{37}$$

or in vector notation,

$$\dot{\mathbf{m}}_j = \dot{S}_j(y_j)[\mathbf{S}(\mathbf{x}) - \mathbf{m}_j] \quad , \tag{38}$$

where $\mathbf{S}(\mathbf{x}) = (S_1(x_1), \ldots, S_n(x_n))$ and $\mathbf{m}_j = (m_{1j}, \ldots, m_{nj})$. m_{ij} denotes the synaptic value between the ith neuron in input field F_X and the jth neuron in competitive field F_Y. Nonnegative signal functions S_i and S_j transduce the real-valued activations x_i and y_j into bounded monotone nondecreasing signals $S_i(x_i)$ and $S_j(y_j)$. $\dot{m}_{ij}$ and $\dot{S}_j(y_j)$ denote the time derivatives of m_{ij} and $S_j(y_j)$, synaptic and signal velocities. $S_j(y_j)$ measures the competitive status of the jth competing neuron in F_Y. Usually S_j approximates a binary threshold function. For example, S_j may equal a steep binary logistic sigmoid,

$$S_j(y_j) = \frac{1}{1 + e^{-cy_j}} \quad , \tag{39}$$

for some constant $c > 0$. The jth neuron wins the laterally inhibitive competition if $S_j = 1$, loses if $S_j = 0$.

For discrete implementation, we use the DCL algorithm as a stochastic difference equation [Kong, 1991]:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + c_t \, \Delta S_j(y_j(t))[S(\mathbf{x}(t)) - \mathbf{m}_j(t)] \text{ if the } j\text{th neuron wins}, \tag{40}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) \text{ if the } i\text{th neuron loses.} \tag{41}$$

$\Delta S_j(y_j(t))$ denotes the time change of the jth neuron's competition signal $S_j(y_j)$ in the

competition layer F_Y:

$$\Delta S_j(y_j(t)) \;=\; sgn[S_j(y_j(t+1)) \;-\; S_j(y_j(t))] \quad . \tag{42}$$

We define the signum operator $sgn(x)$ as

$$sgn(x) \;=\; \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \quad . \tag{43}$$

$\{c_t\}$ denotes a slowly decreasing sequence of learning coefficients, such as $c_t = .1(1 - t/2000)$ for 2000 training samples. Stochastic approximation [Huber, 1981] requires a decreasing gain sequence $\{c_t\}$ to suppress random disturbances and to guarantee convergence to local minima of mean-squared performance measures. The learning coefficients should decrease slowly,

$$\sum_{t=1}^{\infty} c_t \;=\; \infty \quad , \tag{44}$$

but not too slowly,

$$\sum_{t=1}^{\infty} c_t^2 \;<\; \infty \quad . \tag{45}$$

Harmonic-series coefficients, $c_t = 1/t$, satisfy these constraints.

We approximate the competitive signal difference ΔS_j as the activation difference Δy_j:

$$\Delta S_j(y_j(t)) \;=\; sgn[y_j(t+1) \;-\; y_j(t)] \tag{46}$$

$$\;=\; \Delta y_j(t) \quad . \tag{47}$$

Input neurons in feedforward networks usually behave linearly: $S_i(x_i) = x_i$, or $S(\mathbf{x}(t)) = \mathbf{x}(t)$. Then we update the winning synaptic vector $\mathbf{m}_j(t)$ with

$$\mathbf{m}_j(t+1) \;=\; \mathbf{m}_j(t) \;+\; c_t\,\Delta y_j(t)[\mathbf{x}(t) \;-\; \mathbf{m}_j(t)] \quad . \tag{48}$$

We update the F_Y neuronal activations y_j with the additive model

$$y_j(t+1) \;=\; y_j(t) \;+\; \sum_i^n S_i(x_i(t))m_{ij}(t) \;+\; \sum_k^p S_k(y_k(t))w_{kj} \quad . \tag{49}$$

For linear signal functions S_i, the first sum in (49) reduces to an inner product of sample and synaptic vectors:

$$\sum_i^n x_i(t)m_{ij}(t) \;=\; \mathbf{x}^T(t)\mathbf{m}_j(t) \quad . \tag{50}$$

Then positive learning tends to occur—$\Delta m_{ij} > 0$—when $\mathbf{x}$ is close to the jth synaptic vector $\mathbf{m}_j$.

Since a binary threshold function approximates the output signal function $S_k(y_k)$, the second sum in (49) sums over just the winning neurons: $\sum_k w_{kj}$ for all winning neurons y_k .

The $p \times p$ matrix W contains the F_Y within-field synaptic connection strengths. Diagonal elements w_{ii} are positive, off-diagonal elements negative. Winning neurons excite themselves and inhibit all other neurons. Figure 22 shows the connection topology of the laterally inhibitive DCL network.

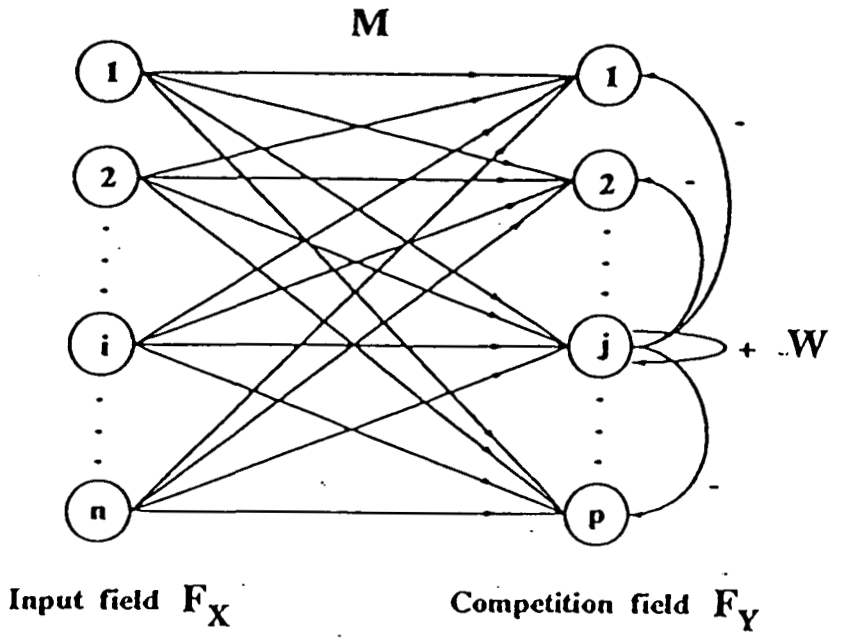


FIGURE 22 Topology of the laterally inhibitive DCL network.