The Knowledge Engineering Review, Vol. 9:3, 1994, 229–268

Fuzzy systems and fuzzy expert control: An overview

SPYROS G. TZAFESTAS

Intelligent Robotics and Control Unit (IRCU), Department of Electrical and Computer Engineering, National Technical University of Athens, Zografou 15773, Athens, Greece

Abstract

This paper presents an overview of fuzzy set theory and its application to the analysis and design of fuzzy expert control systems. Starting with a short account of the basic concepts and properties of fuzzy sets and fuzzy reasoning, a few fuzzy rule-based controllers, *viz.* basic single-input single-output fuzzy control, self-organizing fuzzy control, fuzzy PID supervisor, and the fuzzy PID incremental controller, are described in some detail. Then a survey of the theoretical results and applications is provided which gives a good picture of the current status of the field. This survey includes the work on neuro-fuzzy systems, and software systems for the representation and processing of fuzzy information. The paper closes with four application examples which show the type of results that must be expected from fuzzy expert control.

1 Introduction

Fuzzy set and fuzzy logic theory was initiated by Zadeh (1965), and permits the treatment of vague, imprecise and ill-defined knowledge and concepts in an exact mathematical way. Throughout the years, this theory was fully studied and used for the analysis, modelling and control of technological and non-technological systems. Actualy, our life and world obey the *principle of compatibility* of Zadeh, according to which "the closer one looks at a 'real' world problem, the fuzzier becomes its solution". Stated informally, the essence of this principle is that, as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold beyond which precision and significance (relevance) become almost exclusive characteristics.

To understand what actually inspired and pushed Zadeh to initiate fuzzy system theory through his pioneering article of 1965 (Zadeh, 1965), one must resort to his earlier studies. In one of them he introduced the concept of an *abstract object* to define a system:

> ". . . A system is designed as a collection of ordered pairs, representing inputs and outputs and defining abstract objects. This way, one may treat various systems as a whole by their causal relationships only, thus dealing with a large variety of problems in a unified way."

The next steps, of course, are to locate an input-output with the property characterizing the system. From that point, the problem falls within the framework of classical system theory approach, and leads to the development of concepts like the transfer function, frequency response, state-space model, etc. In the above arguments, it was tacitly assumed that the inputs and outputs of the system can be measured quantitavely (numerically). This is true for technological systems that do not involve the "human" as part of them. However, this ceases to be true when the "human" is entered as a component of the system control loop. This is because the human accepts stimulations (inputs) and replies with responses (outputs) that are expressed better by words (linguistically) than by numbers (numerically). According to Zadeh (1973), the linguistic description of a system is much more effective, but less specific, than the numerical (mathematical) description.

Fuzzy or linguistic controllers and fuzzy reasoning have found particular applications in industrial systems which are very complex and cannot be modelled precisely, even under various assumptions and approximations. The control of such systems by experienced human operators was proved to be, in many cases, more successful and efficient than by classical automatic controllers. The human controllers employ experiential rules which can cast into the fuzzy logic framework. These observations inspired many investigators to work in this area, with the result being the development of so-called "fuzzy logic" and "fuzzy rule-based" control. (Chang and Zadeh, 1972; Tong, 1977a; Larsen, 1980; Negoita and Ralescu, 1975; Lee, 1990).

On the theoretical side, so-called "fuzzy relational equations" have been developed, which are analogous to the normal finite difference equations and can be used to represent fuzzy dynamic systems in both the input-output and state-space model forms (Pappis and Sugeno, 1976; Tong, 1976, 1977b; Pedrycz, 1993; DiNola et al., 1985; Miyakoshi et al., 1985; Miyakoshi and Shimbo, 1986; Hirota, 1979).

Our purpose in this review paper is to present the state-of-art of this field, starting from the fundamental concepts and arriving at the most recent developments. Of course, a complete coverage of the fuzzy systems and control topic is not possible in a single paper, but an effort has been made to include most representative techniques and results.

The structure of the paper is as follows. Section 2 gives some background material on fuzzy sets and fuzzy reasoning which will help the reader to fully understand the rest of the paper. Section 3 outlines four representative fuzzy controllers, namely the basic fuzzy controller, a self-organizing fuzzy controller, a fuzzy PID supervisor and a fuzzy incremental controller. Section 4 presents a comprehensive survey of the theoretical results on fuzzy systems and neuro-fuzzy systems, which combine the features of neural and fuzzy systems. Finally, section 5 gives a survey of the applications and software systems that implement fuzzy reasoning and control algorithms, and section 6 presents a few representative examples of fuzzy expert control. Some directions for further work are pointed out in the conclusions section.

2 Background: Fuzzy sets and fuzzy reasoning

The concept of "set" plays a fundamental role in mathematics. Actually, one cannot define in a rigorous unique way what is a set, what is number or what is a straight line (Black, 1963). These concepts can be understood better not via definitions but via examples (Bellman and Giertz, 1973).

Let U be a classical set and x an element. Then one of the following holds (how much this holds will be discussed soon): the element x belongs to U (symbolically $x \in U$) or x does not belong to U ($x \notin U$). This is the so called *principle of dichotomy*. By disputing dichotomy, the classical (crisp) set theory breaks down and the *fuzzy set* theory naturally emerges (Gain, 1983).

Let $U = \{x_1, x_2, x_3, x_4, x_5\}$ be a classical (crisp) set which is called *universe of discourse*. Now let $A = \{x_1, x_3, x_5\}$ be a crisp subset of U. The set A can be equivalently described by $A = \{(x_1, 1), (x_2, 0), (x_3, 1), (x_4, 0), (x_5, 1)\}$, i.e. as a set of pairs $(x, \mu_A(x))$, where x is the element of interest, and $\mu_A(x)$ is the *membership function* of x in the subset A, where

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

The question now arises whether $\mu_A(x)$ can take only the values 0 and 1 or any value between 0 and 1. In other words, why $\mu_A: A \to \{0, 1\}$ *and not* $\mu_A: A \to [0, 1]$? This question was first examined by Zadeh, and is the starting point for the development of fuzzy sets (Zadeh, 1965, 1969).

2.1 Fuzzy sets

A fuzzy subset F of a universe of discourse $U = \{x\}$ is defined as a mapping

$$\mu_F(x): U \to [0, 1] \tag{1}$$

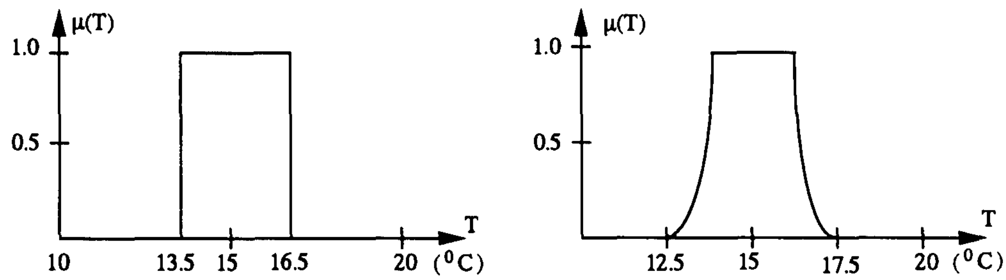


Figure 1 (a) Ordinary set representation; (b) fuzzy set representation

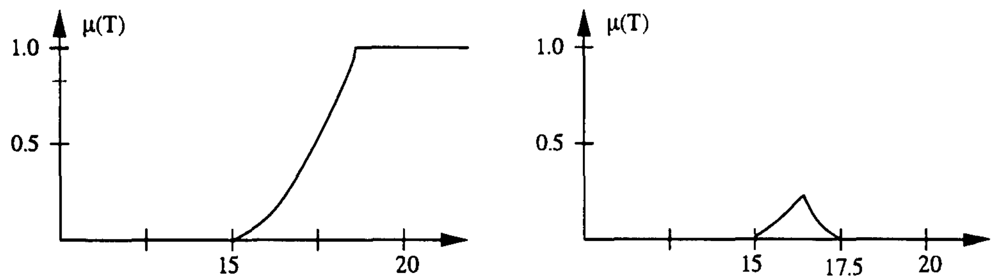


Figure 2 (a) Another fuzzy set of temperature; (b) intersection of the fuzzy sets of figs. 1b and 2a

by which x is assigned a number in [0, 1], indicating the extent to which x belongs to (has the attribute) F. For example, if we wish to specify linguistic measures of temperature on the interval [10°C, 20°C], and we are given such a measure of {temperatures about 15°}, we can use either an ordinary set representation or a fuzzy set representation, as shown in fig. 1a,b. In the ordinary set we have a sharp (definite) transition of membership to non-membership, whereas in the fuzzy set we have a gradual membership transition.

Clearly, using the fuzzy set concept allows imprecise and qualitative information to be represented in an exact mathematical way.

2.2 Operations on fuzzy sets

Given the fuzzy sets A and B of U with membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively, the complement (negation) of A, the union of A and B, and the intersection of A and B are defined as

$$\text{Complement } \bar{A} \quad : \mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2a}$$

$$\text{Union } A \cup B \quad : \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \tag{2b}$$

$$\text{Intersection } A \cap B \quad : \mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \tag{2c}$$

One can easily verify that linguistic measures are not necessarily mutually exclusive, and hence one can obtain information in what seem to be disjoint situations. For example, the ordinary temperature sets: (i) temperatures in the interval [13.5°C, 16.5°C], and (ii) temperatures greater than 17.5°C are disjoint, but the fuzzy set of fig. 1b {temperatures about 15°C} with the fuzzy set of fig. 2a {temperatures much greater than 10°C} may have a non-null intersection as that of fig. 2b.

2.3 Fuzzy rules

An example of linguistic rule (or functional relationship) is

IF (fuel input is high)
Then (engine speed is much greater than 100 rpm)

Expressed in terms of fuzzy sets, this rule is represented by a 2-dimensional fuzzy set R, in the product space $F \times S$, which has the membership function

$$\mu_R = \mu_{A \times B} = \min \{\mu_A(F), \mu_B(S)\} \tag{3}$$

indicating the extend to which R is true for (F, S).

If we have several fuzzy rules each one producing a set $R_i (i = 1, 2, \ldots, N)$, an overall set R is obtained by using the union

$$R = \bigcup_{i=1}^{N} R_i \tag{4}$$

where R_i is the 2-dimensional set produced by the ith rule.

According to Zadeh, such a collection of fuzzy rules is called a *fuzzy algorithm* or otherwise a *fuzzy model*.

2.4 Generalized fuzzy operators

The families of T-operators provide suitable generalizations of Zadeh's three fundamental operations max (•, •), min (•, •), 1-(•) for the union, section and complement, respectively. We have three kinds of T-operators, namely T-norms, T-conorms and N-negation functions. The definitions of them are as follows.

Definition 1
The function $T: [0, 1] \times [0, 1] \to [0, 1]$ is called T-norm if and only if T possesses the following properties for all $x, y, z \in [0, 1]$:

1. $T(x, y) = T(y, x)$
2. $T(x, y) \leqslant T(x, z)\ \forall y \leqslant z$
3. $T(x, T(y, z)) = T(T(x, y), z)$
4. $T(x, 1) = 1$

Definition 2
The function $T^*: [0, 1] \times [0, 1] \to [0, 1]$ is called T-conorm if and only if T^* possesses the following properties for all $x, y, z \in [0, 1]$:

1. $T^*(x, y) = T^*(y, x)$
2. $T^*(x, y) \leqslant T^*(x, z)\ \forall y \leqslant z$
3. $T^*(x, T^*(y, z)) = T^*(T^*(x, y), z)$
4. $T^*(x, 0) = 0$

Definition 3
The function $N: [0, 1] \to [0, 1]$ is called negation function if and only if N possesses the following properties for all $x, y \in [0, 1]$:

1. $N(0) = 1, \quad N(1) = 0$
2. $N(x) \leqslant N(y)\ \forall x \geqslant y$

Throughout the years several T-operators have been proposed. Some of them are listed in Table 1.

Actually, no single triple of T-operators is the best in all decision making problems. The simplest triple is the one suggested by Zadeh, but it is not the most suitable in many cases. For a detailed discussion of T-operators, readers are referred to (Gupta, 1991).

2.5 The representation theorem

The possibility of approximating a fuzzy set by a crisp set is a fundamental topic of fuzzy set theory. The dominating role for this approximation is played by the concept of α-cut. This is because the α-

Table 1 A set of T-operators

N	$T_N(x, y)$	$T_N^*(x, y)$	$N(x)$	Researcher
1.	$\min(x, y)$	$\max(x, y)$	$1 - x$	Zadeh
2.	$x \cdot y$	$x + y - x \cdot y$	$1 - x$	Gogeuen
3.	$\max(x + y - 1, 0)$	$\min(x + y, 1)$	$1 - x$	Giles
4.	$\frac{x \cdot y}{x + y - x \cdot y}$	$\frac{x + y - 2 \cdot x \cdot y}{1 - x \cdot y}$	$1 - x$	Bandler
5.	$\frac{\lambda \cdot x \cdot y}{1 - (1 - \lambda) \cdot (x + y - x \cdot y)}$	$\frac{\lambda \cdot (x + y) + x \cdot y \cdot (1 - 2 \cdot \lambda)}{\lambda + x \cdot y \cdot (1 - \lambda)}$	$1 - x$	Hamaster
6.	$\frac{x \cdot y}{\max(x, y, \lambda)}$	$1 - \frac{(1 - x) \cdot (1 - y)}{\max(1 - x, 1 - y, \lambda)}$	$1 - x$	Dubois

cut is a crisp set that approximates the fuzzy set with "accuracy α". The α-cut of a fuzzy set A is the set

$$L_\alpha A = \{x \in X : \mu_A(x) \geq \alpha\}$$

A first idea is to approximate the fuzzy set with a sequence of α-cuts where $0 \leq \alpha \leq 1$. Then it is obvious that $L_0 A = X$ and $\alpha \leq \beta \Leftrightarrow L_\beta B \subseteq L_\alpha A$.

The approximation problem is formulated as follows. Let A, B, C, . . . be fuzzy sets and $L_\alpha A$, $L_\alpha B$, $L_\alpha C$, . . . their α-cuts. As a first step in the approximation, one works with $L_\alpha A$, $L_\alpha B$, $L_\alpha C$, . . . which are crisp sets, and from them produce a set, say ${}_\alpha Z$, for some $\alpha \in [0, 1]$. This is repeated many times, and so from the given families of the α-cuts $L_\alpha A$, $L_\alpha B$, $L_\alpha C$, . . . a sequence of results ${}_\alpha Z$ is produced for all α, $\alpha \in [0, 1]$. The question now arises, does there exist a fuzzy set Z which would give the above family of sets ${}_\alpha Z$ for all α, $0 \leq \alpha \leq 1$ in the α-cut sense? The answer to this question is positive as stated in the following theorem:

Theorem 1
Let ${}_\alpha Z$, $0 \leq \alpha \leq 1$ be the family of crisp sets with the following properties:

(i) ${}_0 Z = X$

(ii) $\alpha \leq \beta \Leftrightarrow {}_\alpha Z \leq {}_\beta Z$

(iii) $\alpha_1 \leq \ldots \leq \alpha_n, \quad \lim_{n \to \infty} \alpha_n = \alpha \Rightarrow {}_\alpha Z = \bigcap_{n=1}^{\infty} {}_{\alpha_n} Z$

Then there exists a unique fuzzy set Z such that $L_\alpha Z = {}_\alpha Z$.

For proof the reader is referred to Negoita and Ralescu (1987).

2.6 The fuzzy extension principle

The extension principle helps in the fuzzification of mathematical laws, and thus plays a dominant role in fuzzy set theory.

Definition 4 (Extension Principle)
Let $X_1, X_2, \ldots, X_r$ be universes of discourse; $X = X_1 \times X_2 \times \ldots \times X_r$ their Cartesian product; A_1, $A_2, \ldots, A_r$ respective fuzzy subsets; and $f: X_1 \times X_2 \times \ldots \times X_r \rightarrow Y$ with $y = f(x_1, \ldots, x_r)$ a crisp

function. Then the *extension principle* transfers the fuzziness of $A_1, \ldots, A_r$ into a fuzzy set B of Y, where

$$B = \{(y, \mu_B(y)) \mid y = f(x_1, \ldots, x_r), \quad (x_1, \ldots, x_r) \in X_1 \times \ldots \times X_r\}$$

and

$$\mu_B(y) = \begin{cases} \sup\limits_{(x_1,\ldots,x_r)\in f^{-1}(y)} \min\{\mu_{A_1}(x), \ldots, \mu_{A_r}(x)\}, & f^{-1}(y) \neq \emptyset \\ 0, & f^{-1}(y) = \emptyset \end{cases}$$

For $n = 1$ the extension principle coincides with the concept of "image of a fuzzy set", i.e.

$$Y = \{(y, \mu_Y(y)) \mid y \in Y, \quad \mu_X(y): Y \to [0, 1]\}$$

where

$$\mu_Y(y) = \begin{cases} \sup\limits_{x\in f(x)} \mu_X(x), & f^{-1}(x) \neq \emptyset \\ 0, & f^{-1}(x) = \emptyset \end{cases}$$

The *extension principle* allows a crisp function f, with a definition domain of the crisp set X, to change its definition domain to, say, certain fuzzy subsets of X (instead of X itself). For example, if A is a fuzzy subset of the crisp set X, and f is a crisp function defined on X, then the extension principle shows how the function f can be applied upon the fuzzy set A. In particular, if $A = \mu_A(x_1)/x_1 + \cdots + \mu_A(x_n)/x_n$, then

$$\begin{aligned} f(A) &= f(\mu_A(x_1)/x_1 + \cdots + \mu_A(x_n)/x_n) \\ &= \mu_A(x_1)/f(x_1) + \cdots + \mu_A(x_n)/f(x_n) \end{aligned}$$

It is noted that f is always applied upon the elements of the crisp set X.

2.7 Linguistic variables

From the previous discussion, one can see that the adherence in the extension of classical mathematical concepts (like the number, the matrix, the determinant, etc.) to fuzzy logic faces many difficulties, and in some cases makes the situation more complex than it actually is. Thus to overcome these difficulties, one has to resort to new concepts. A new concept which plays a basic and dominant role in fuzzy logic and reasoning is the concept of a linguistic variable. Zadeh justifies the introduction of linguistic variables as follows:

> ". . . The motivation for the use of words or sentences rather than numbers is that linguistic characterizations are, in general, less specific than numerical ones . . ."

Definition 5
A linguistic variable is a variable the values of which are not numbers but words or sentences or propositions in a natural or artificial language.

More formally, the linguistic variable is defined as follows:

Definition 6
A linguistic variable is the quintuple $\langle x, T(x), U, G, M\rangle$ where x is the name of the linguistic variable, $T(x)$ is the set of its values (*term set*), U is the universe of discourse upon which $T(x)$ is structured (recall that the values of a linguistic variable are fuzzy sets), G is a syntactic rule that generates the names x, and M is a semantic rule that gives sense (meaning) to the names.

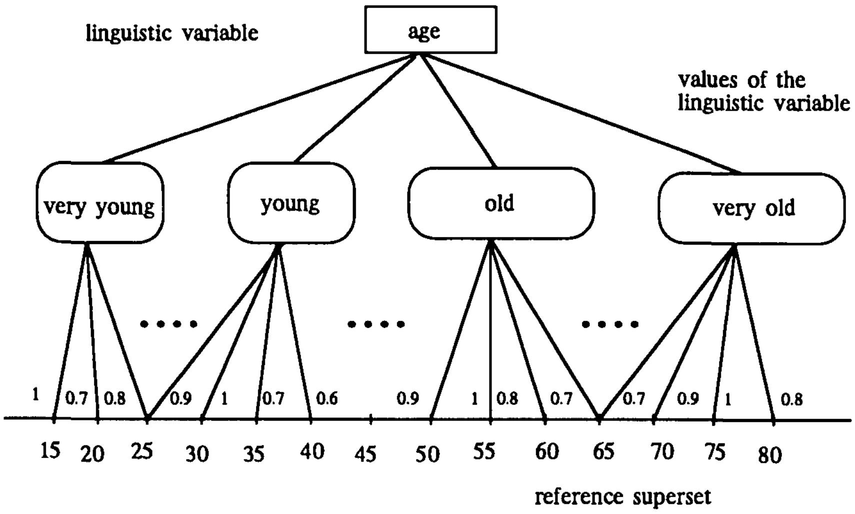


Figure 3 Linguistic variable "age" and its "linguistic values"

Example 1

Consider the linguistic variable X-"age" with $T(x) = \{$very young, not very young, pretty young, a little old, pretty old, not very old, very old$\}$. Then, a meaning of the fuzzy set "old" is $M_{\text{old}} = \{(u, \mu_{\text{old}}(u)),\ u \in [0, 100]\}$, where

$$\mu_{\text{old}}(n) = \begin{cases} 0, & u \in [0, 50] \\ \left[1 + \left(\dfrac{u - 50}{5}\right)^{-2}\right]^{-1}, & u \in [50, 100] \end{cases}$$

The linguistic variable "age" and its values are pictorially depicted in fig. 3.

Definition 7

A linguistic variable x is said to be *structured* if its set of values $T(x)$ and the set of meanings $M(x)$ that are given to them can be determined algorithmically.

Example 2

Consider the linguistic variable "truth" with the following set of values: $T(\text{truth}) = \{$true, not true, very true, not very true, false, not false, very false, not very false$\}$. It is clear that $T(x)$ can be determined algorithmically, and so the variable "truth" is a structured variable.

2.8 Fuzzy composition

Given a fuzzy relation R from U to V, and a fuzzy subset A of U, a fuzzy subset of V is inferred given by the max-min compositional rule:

$$\mu_B(x) = \max_{x} \{\min\{\mu_R(x, y), \mu_A(x)\}\} \tag{5}$$

where $U = \{x\}$ and $V = \{y\}$, or compactly

$$B = A \circ R \tag{6}$$

One can observe that the output fuzzy set has a maximum membership function much below unity. Thus, the resultant (output) set must be regarded as a best estimate given the inadequate information for the input, and the relation between input and output.

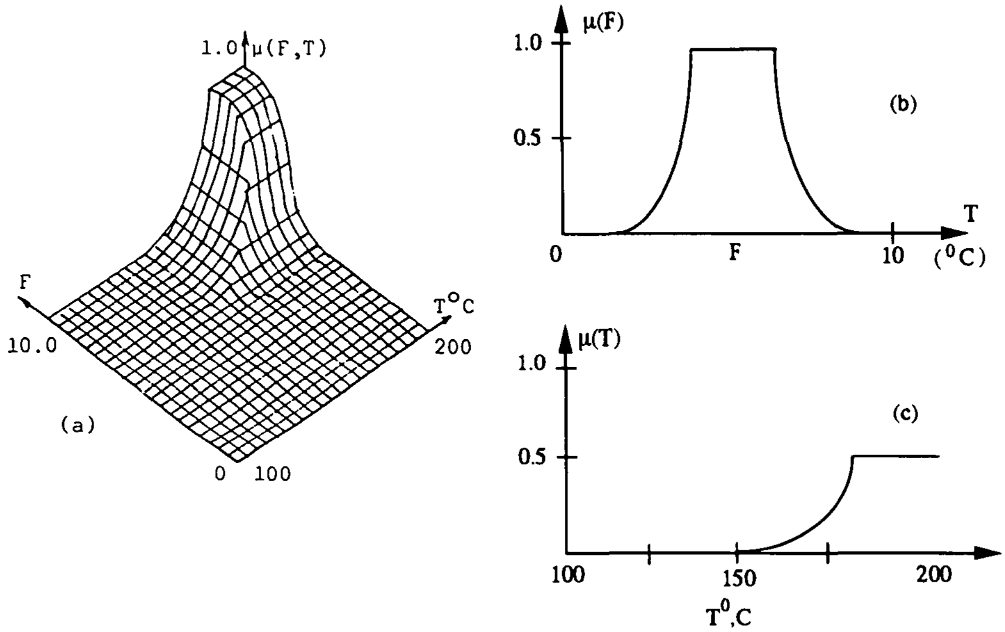


Figure 4 (a) 2-dimensional fuzzy relation (rule); (b) fuzzy input set; (c) fuzzy output set

Example 3

Consider a rule R linking fuel and engine speed (fig. 4a) and assume that fuel input is medium (fuzzy set of fig. 4b). Then $B = A \circ R$ given by equation (6) is the fuzzy set of fig. 4c.

Example 4

Consider the following: (a) $X = \{1, 2, 3, 4\}$, (b) A = "x small" = $\{(1, 1), (2, 0.6), (3, 0.2), (4.0)\}$, (c) R = "x nearly equal to y" with a matrix as given below.

R = "x nearly equal to y"					
R		x			
		1	2	3	4
	1	1	0.5	0	0
	2	0.5	1	0.5	0
y	3	0	0.5	1	0.5
	4	0	0	0.5	1

It is desired to find what happens with the variable y, i.e. to determine the fuzzy set B = "y". To this end, use of the max-min composition rule of inference is made, i.e. $B = A \circ R$, where

$$\mu_B(y) = \max_x\{\min\{\mu_A(x), \mu_R(x, y)\}\} = \{(1, 1), (2, 0.6), (3, 0.5), (4, 0.2)\}$$

Thus, IF "x small" and "x nearly equal to y" THEN "y nearly small".

Using fuzzy sets, the uncertainty and qualitativeness are expressed by a single curve. A collection of fuzzy rules provides the system model, replacing the conventional mathematical system model. The fuzzy composition rule provides the tool for developing fuzzy state-space models (analogously to the models obtained through matrix multiplication).

2.9 Multidimensional fuzzy rules and reasoning

Let us denote the truth value of a proposition "x is A and y is B" by

$$|x \text{ is } A \text{ and } y \text{ is } B| = A(x) \wedge B(y)$$

and consider the multidimensional fuzzy rule

$$R: \quad \text{IF } f(x_1 \text{ is } A_1, \ldots, x_k \text{ is } A_k)$$
$$\text{THEN } y = h(x_1, x_2, \ldots, x_k)$$

where x_i $(i = 1, 2, \ldots, k)$ are the input (premise) variables, y is the output (consequence) variable, A_i $(i = 1, 2, \ldots, k)$ are fuzzy sets with linear membership functions, f is a logical constraint on the input propositions, and h is a function that implies the output value y when $\{x_1, x_2, \ldots, x_k\}$ satisfy the premise. If, for some input x_i, A_i is equal to the universe of discourse of x_i, the corresponding term is omitted (i.e. x_i is unconditioned).

An example of multidimensional fuzzy rule is the following:

$$R: \quad \text{IF } x_1 \text{ is very small and } x_2 \text{ is large}$$
$$\text{THEN } y = x_1 + x_2 + 4x_3$$

where x_3 is unconditioned.

The general form of such a rule (with only "and" connectives in the premise) is

$$R: \quad \text{IF } x_1 \text{ is } A_1 \text{ and } \ldots \text{ and } x_k \text{ is } A_k$$
$$\text{THEN } y = a_0 + a_1 x_1 + \ldots + a_k x_k$$

Given the rules (implications) R_i $(i = 1, 2, \ldots, N)$ of the above format, and assuming that $\{x_1 = x_1^*, \ldots, x_k = x_k^*\}$ where x_i^* $(i = 1, 2, \ldots, k)$ are singletons, the value of y is obtained (inferred) using the following algorithm (Takagi and Sugeno, 1985):

(i) For each R_i, y_i is inferred using h in the consequence, i.e.

$$y_i = h_i(x_1^*, \ldots, x_k^*)$$
$$= a_0^i + a_1^i x_1^* + \ldots + a_k^i x_k^*$$

(i) The truth value of the proposition $y = y_i$ is calculated by

$$|y = y_i| = |x_i^* \text{is } A_1^i \text{ and } \ldots \text{ and } x_k^* \text{ is } A_k^i| \wedge |R_i|$$
$$= \{A_1^i(x_1^*) \wedge \ldots \wedge A_k^i(x_k^*)\} \wedge |R_i|$$

where $|p|$ denotes the truth value of the proposition p, $\wedge$ stands for the "min" operation, and $|x^* \text{ is } A| = A(x^*)$, i.e. the grade of membership of x^*.

When $|R_i| = 1$, we have

$$|y = y_i| = A_1^i(x_1^*) \wedge \ldots \wedge A_k^i(x_k^*)$$

(iii) The final output value y inferred from N rules is given by

$$y = \Sigma\, |y = y_i| \cdot y_i / \Sigma\, |y = y_i|$$

i.e. by the average of all y_i with weights $|y = y_i|$.

Example 5

Consider the three values

R_1: IF x_1 is small_1 and x_2 is small_2 THEN $y = x_1 + x_2$

R_2: IF x_1 is big_1 THEN $y = 2x_1$

R_3: IF x_2 is big_2 THEN $y = 3x_2$

with

$$x_1^* = 12, \quad x_2^* = 5$$

$$\text{small\_1}(x_1^*) = 0.25, \quad \textit{small\_2}(x_2^*) = 0.375$$

$$\text{big\_1}(x_1^*) = 0.2, \quad \text{big\_2}(x_2^*) = 0.375$$

We have $y_1 = x_1^* + x_2^* = 12 + 5 = 17$ with membership grade

$$|y = y_1| = |x_1^* = \text{small\_1}| \wedge |x_2^* = \text{small\_2}|$$

$$= \text{small\_1}(x_1^*) \wedge \text{small\_2}(x_2^*) = \mu_{x_1}(x_1^*) \wedge \mu_{x_2}(x_2^*)$$

$$= 0.25$$

Similarly

$$y_2 = 2 \times 12 = 24 \quad \text{with } |y = y_2| = 0.2$$

and

$$y_3 = 3 \times 5 = 15 \quad \text{with } |y = y_3| = 0.375$$

Thus the value of y obtained from the three rules together is

$$y = \frac{0.25 \times 17 + 0.2 \times 24 + 0.375 \times 15}{0.25 + 0.2 + 0.375}$$

$$= 17.8$$

Fuzzy reasoning (inference) is convenient for treating both "soft" (humanistic) and "hard" (technological) systems involving qualitative and experiential knowledge expressed in a linguistic way.

2.10 Estimation of the membership function: defuzzification

The estimation of the membership function which leads to *defuzzification* can be done in several ways that are arbitrary.

The membership function $\mu_A(x)$ that is used most is the triangular one which possesses a maximum value (usually 1) in the most *representative* and *characteristic* value of the fuzzy variable x. Thus in this case, the defuzzification leads to the value of maximum $\mu_A(x)$.

Five other more systematic methods of defuzzification (estimation of $\mu_A(x)$) are:

1. *Average guess method*
   In this method we calculate the average value of the values of $\mu_A(x)$ that are suggested by experts. This is used as the best estimate of $\mu_A(x)$. The corresponding value of x is the required deffuzified value of x.
2. *Distance function method*
   Initially, we compute a distance $\boldsymbol{d(x)}$ of the arbitrary point x from the fuzzy set A under consideration. For the elements that belong to A, this distance is obviously zero. For those x not belonging to A, this distance takes some maximum value say **sup*d***. On the basis of the above, the membership value is given by

$$\mu_A(x) = 1 - \frac{d(x)}{\sup d}$$

3. *Intuitive relation method*
   It is intuitively true that the rate of change of $\mu_A(x)$ must increase when the belief that x belongs to the fuzzy set A is strengthened. Analytically, this is expressed by the relation

   $$\frac{d\mu_A(x)}{dx} = K\mu_A(x)[1 - \mu_A(x)]$$

   which, upon integration, gives the membership function

   $$\mu_A(x) = \frac{1}{1 + \exp(a - bx)}$$

   The constants a and b are determined by the other data of the problem.
4. *Binary polling method*
   In this method we form a group of experts and ask them whether "x belongs to the set A". Their reply should be binary, i.e. yes or no. Then the value of $\mu_A(x)$ is estimated by

   $$\mu_A(x) = \frac{\text{number of positive replies "yes"}}{\text{total number of replies}}$$

5. *Relative preference method*
   Let A be a discrete fuzzy set

   $$A = \{(x, \mu_A(x)) \mid x \in X, \quad \mu_A(x): A \rightarrow [0, 1]\}$$

   As the *relative preference* of x_i with respect to x_j is defined the quantity

   $$p_{ij} = \frac{\mu_A(x_i)}{\mu_A(x_j)}$$

   We now form the matrix $P = [p_{ij}]$, which possesses the following three properties:

   - $p_{ii} = 1$ and $p_{ij} = 1/p_{ji}$.
   - All eigenvalues of P are zero except one, which is equal to the cardinality n of A.
   - The eigenvector w that corresponds to the nonzero eigenvalue n has as its elements the values of the membership function corresponding to the elements of A, i.e. $w = [\mu_A(x_i)] = [\mu_A(x_1) \ldots \mu_A(x_n)]^T$.

   Of course, this method is applicable to discrete finite fuzzy sets, not to continuous fuzzy sets that have an infinite number of elements.

3 Review of some fuzzy or linguistic controllers

3.1 Basic fuzzy controller

We start with a brief description of the structure of the basic fuzzy control loop. Fuzzy control uses linguistic variables (this is why it is sometimes called linguistic control), and mimics the human action more closely than traditional control. For comparison, fig. 5 depicts the architecture of both the traditional and the basic fuzzy control loop.

The problem in the traditional control loop is to design a controller which will accept the error $e = r - y$ as input and will give an output (control signal) u such that the output y of the overall system follows the set point (reference input) r as near as possible.

The problem in the fuzzy control loop is to design an inference mechanism (fuzzy controller) which will mimic the human type of syllogism and will ensure a desired performance of the overall

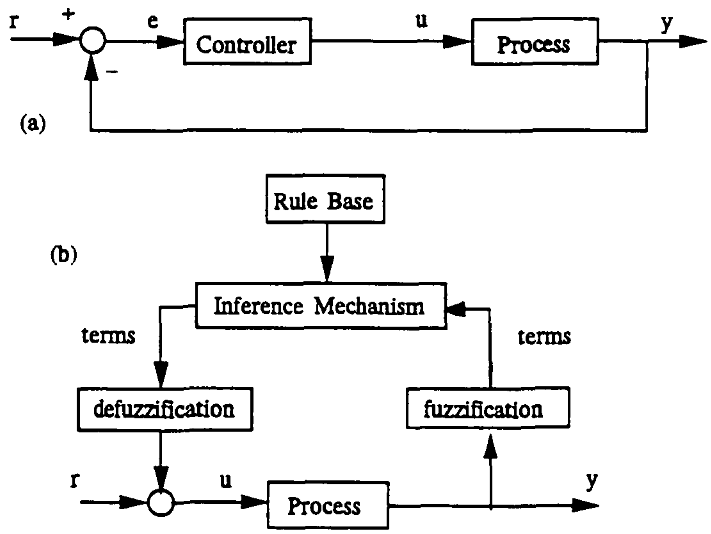


Figure 5 (a) Traditional control loop; (b) basic fuzzy control loop

system. The fuzzy control loop involves the following blocks: a rule base, an inference mechanism and the fuzzification/defuzzification blocks. The fuzzification/defuzzification blocks have been discussed earlier. The fuzzy knowledge base contains the rules that are to be used for control of the process. These rules are usually the result of interviews with the expert operators (very rarely come out of mathematical analysis or simulations), and have the form IF-THEN. In the general case, the rules have many inputs and many outputs (MIMO). However, it can be shown (Lee, 1990) that a set of MIMO rules can be transformed to a set of MISO (multi input–single output) rules.

As an example, we consider here rules with two inputs and one output, i.e. rules of the form

$$R_i\text{: IF "}x \text{ is } A_1\text{" } \textit{and} \text{ "}y \text{ is } B_1\text{" THEN "}z \text{ is } C_1\text{"} \tag{7}$$

A rule of this type has two *assumptions* (premises), "x is A_1" and "y is B_1", which are defined on the Cartesian product $U \times V$. The whole rule constitutes a fuzzy relation in the fuzzy Cartesian product $U \times V \times W$, i.e. $\mu_{R_i} = \mu_{(A_i \text{ and } B_i \text{ then } C_i)}(u, v, w)$ where U, V and W are the respective fuzzy sets.

An easy way to use method for the calculation of the relation R makes use of the "min" operator, namely

$$\mu_{R_i} = \mu_{(A_i \text{ and } B_i \text{ then } C_i)}(u, v, w) = \min\{\mu_{A_i}(u), \mu_{B_i}(v), \mu_{C_i}(w)\} \tag{8}$$

If the knowledge based contains a total of n rules (R_i, $i = 1, 2, \ldots, n$), then it can be regarded as a unique relation R, where

$$R = \bigcup_{i=1}^{n} R_i$$

Suppose now that at a certain instant of time we have observed in the process that "x is A'" and "y is B'". The problem is to combine this fact with the rules of the knowledge base to produce a suitable control y. To this end, one must use the max-min composition scheme

$$(A', B') \circ \bigcup_{i=1}^{n} R_i = \bigcup_{i=1}^{n} (A', B') \circ R_i$$

The right-hand side of the above identity suggests that instead of applying the fact (A', B') to the knowledge base $\cup_{i=1}^{n} R_i$ as a whole, one can apply (A', B') to each rule R_i of the knowledge base separately. The proof of the validity of the above identity can be found in Lee (1990). In this way, for each input one can evaluate the individual contribution of each rule to the final result. This

Table 2 Fuzzy error look-up table (13 quantization levels with two levels of zero)

	−6	−5	−4	−3	−2	−1	−0	+0	+1	+2	+3	+4	+5	+6
PB	0	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0
PM	0	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2
PS	0	0	0	0	0	0	0	0.3	0.8	1.0	0.5	0.1	0	0
PO	0	0	0	0	0	0	0	1.0	0.6	0.1	0	0	0	0
NO	0	0	0	0	0.1	0.6	1.0	0	0	0	0	0	0	0
NS	0	0	0.1	0.5	1.0	0.8	0.3	0	0	0	0	0	0	0
NM	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0	0
NB	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0	0

PB: positive big; PM:positive medium; PS: positive small; PO: positive zero; NB: negative big; NM: negative medium; NS: negative small; NO: negative zero.

Table 3 Fuzzy change-in-error look-up table

	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6
PB	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0
PM	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2
PS	0	0	0	0	0	0	0	0.9	1.0	0.7	0.2	0	0
NO	0	0	0	0	0	0.5	1.0	0.5	0	0	0	0	0
NS	0	0	0.2	0.7	1.0	0.9	0	0	0	0	0	0	0
NM	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0
NB	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0

PB: positive big; PM:positive medium; PS: positive small; PO: positive zero; NB: negative big; NM: negative medium; NS: negative small; NO: negative zero.

facilitates the control process, reduces considerably the computational load, and helps to have a better monitoring of which rules are fired and how much they are fired.

From the above, one concludes that two basic problems are encountered when attempting to apply fuzzy control in real systems:

1. Choice of primary fuzzy sets, to be used together with the rules which constitute the control law or algorithm (fuzzy control structure).
2. Numerical description of the linguistics to implement a fuzzy control algorithm in a computer.

The above involve several subproblems, such as:

- determination of the range of the values of the control variables,
- selection of the quantization intervals for the inputs and outputs of the controller,
- precise definition of the primary fuzzy sets on the quantized spaces, and
- selection of the sampling interval.

In the literature one can find a large repertory of fuzzy controllers or fuzzy relational matrices. However, since the relational matrix, even for SISO controllers, has very large dimensionality and demands large computer memory (for a 2-input 2-output system more than 700K storage locations are necessary), we need to introduce some kind of simplification. Two ways in which to do this are the following:

- store the control algorithm as a set of rules, and
- form a look-up table, which is precalculated from the control rules before the controller is run.

Three representative look-up tables used in various applications are shown in Tables 2–4 (Mamdani and Assilian, 1975).

Table 4 Look-up table for heat-change variable

	−7	−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6	+6
PB	0	0	0	0	0	0	0	0	0	0	0	0.1	0.4	0.8	1.0
PM	0	0	0	0	0	0	0	0	0	0.2	0.7	1.0	0.7	0.2	0
PS	0	0	0	0	0	0	0	0.4	1.0	0.8	0.4	0.1	0	0	0
NO	0	0	0	0	0	0	0.2	1.0	0.2	0	0	0	0	0	0
NS	0	0	0	0.1	0.4	0.8	1.0	0.4	0	0	0	0	0	0	0
NM	0	0.2	0.7	1.0	0.7	0.2	0	0	0	0	0	0	0	0	0
NB	1.0	0.8	0.4	0.1	0	0	0	0	0	0	0	0	0	0	0

PB: positive big; PM:positive medium; PS: positive small; PO: positive zero; NB: negative big; NM: negative medium; NS: negative small; NO: negative zero.

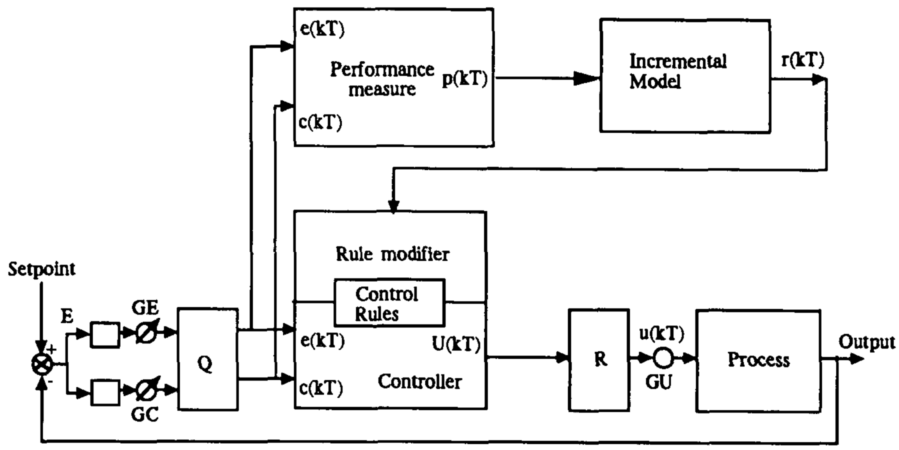


Figure 6 SISO fuzzy self-organizing control loop

3.2 Self-organizing fuzzy controllers (SOC)

A second fuzzy controller, which is an improved version of the basic fuzzy controller studied in the previous section, is the so-called *self-organized controller* (SOC). This controller has more *intelligence* in the sense that it is capable of measuring its own performance and modifying the control rules according to a measure of deviation of each output from the trajectory $p(kT)$, where

$$p(kT) = \Theta\{e(kT), c(kT)\} \tag{9}$$

and $\Theta(e(kT), c(kT)\}$ represents the performance decision look-up table (relational matrix) used (fig. 6).

The performance decision look-up table adopted in Mamdani and Procyk (1979) is shown in Table 5. In this table, zero entries indicate no correction. The further away from desired trajectories the output is, the greater the required control corrections are in one or the other direction.

To implement this SOC controller, which is a kind of fuzzy adaptive controller, we need to have an incremental model relating input changes to output changes. For a system with state equation $\dot{x} = f(x, u)$ the incremental model matrix M is given by

$$M = TJ \tag{10}$$

where J is the system Jacobian matrix $J = \partial f/\partial u$ and T the sampling period. Thus, if an output correction $p(kT)$ is needed, then the required control rule correction is determined by solving the equation $p(kT) = Mr(kT)$ as

$$r(kT) = M^{-1}p(kT) \tag{11}$$

Table 5 Performance measure look-up table

		Change in error $e(kT)$												
		Towards set-point								Away from set-point				
		−6	−5	−4	−3	−2	−1	0	+1	+2	+3	+4	+5	+6
	−6	0	0	0	0	0	0	6	6	6	6	6	6	6
	−5	0	0	0	2	2	3	6	6	6	6	6	6	6
	−4	0	0	0	2	4	5	6	6	6	6	6	6	6
Below set-point	−3	0	0	0	2	2	3	4	4	4	4	5	5	6
	−2	0	0	0	0	0	0	2	2	2	3	4	5	6
	−1	0	0	0	0	0	0	1	1	1	2	3	4	5
	−0	0	0	0	0	0	0	0	0	0	1	2	3	4
Error $e(kT)$	+0	0	0	0	0	0	0	0	0	0	−1	−2	−3	−4
	+1	0	0	0	0	0	0	−1	−1	−1	−2	−3	−4	−5
	+2	0	0	0	0	0	0	−2	−2	−2	−3	−4	−5	−6
Above set-point	+3	0	0	0	−2	−2	−3	−4	−4	−4	−4	−5	−5	−6
	+4	0	0	0	−2	−4	−5	−6	−6	−6	−6	−6	−6	−6
	+5	0	0	0	−2	−2	−3	−6	−6	−6	−6	−6	−6	−6
	+6	0	0	0	0	0	0	−6	−6	−6	−6	−6	−6	−6

The operation of this controller is now obvious from fig. 6. The performance measure (9) and the incremental model (11) constitute the higher level of the SOC which coordinates the simple fuzzy controller of the lower level. The control input correction is fed to the rule modifier which modifies the linguistic rules such that future control actions lead to the appropriate output improvement.

Some key features of this controller are:

- The model matrix $\boldsymbol{R}$ is related to the system Jacobian which, for SISO systems consist of a single coefficient (usually normalized to unity).
- For linear MIMO systems the model matrix $\boldsymbol{R}$ involves constant elements (which, if normalized, lie between -1.0 and $+1.0$).
- For nonlinear or nonmonotonic systems, the model matrix elements depend upon the system state.

The incremental model required is much simpler than the exact (possibly nonlinear) model of the system, but it requires knowledge of the Jacobian $\boldsymbol{J}$ which, in many cases, is difficult to obtain on-line. To overcome this difficulty, one can compute $\boldsymbol{J}$ at the beginning of each control period, keeping it constant throughout, and leave to the learning process the balance of any inaccuracies. The controller is always initiated by an initial set of control rules, which are usually crude and are obtained from existing experience. These rules are then iteratively corrected through the rule modification process.

The control rule modification procedure involves the following:

(a) Fuzzification of uncorrected and corrected control input along with the error and change in error, i.e.

$$E(kT-mT) = F\{e(kT-mT)\} \quad \text{(error)}$$

$$C(kT-mT) = F\{c(kT-mT)\} \quad \text{(change-in-error)}$$

$$U(kT-mT) = F\{u(kT-mT)\} \quad \text{(uncorrected control)}$$

$$U^*(kT-mT) = F\{u(kT-mT)+r(kT)\} \quad \text{(corrected control)}$$

where $F\{\cdot\}$ *represents the fuzzification process*, mT is the time interval between successive control corrections, and $r(kT)$ is the control correction given by (11).

(b) Determine the uncorrected and corrected (modified) relation matrices R' and R'' from the corresponding implications

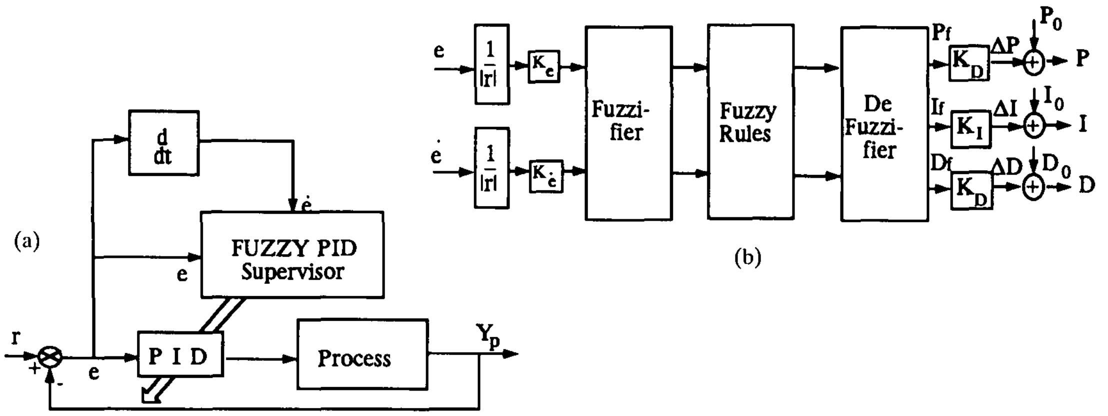


Figure 7 (a) Supervised PID controller; (b) architecture of the supervisor

$$E(kT - mT) \rightarrow C(kT - mT) \rightarrow U(kT - mT)$$
$$E(kT - mT) \rightarrow C(kT - mT) \rightarrow U^*(kT - mT)$$

That is

$$R'(kT) = E(kT - mT) \times C(kT - mT) \times U(kT - mT)$$
$$R''(kT) = E(kT - mT) \times C(kT - mT) \times U^*(kT - mT)$$

(c) Express the control relation matrix modification linguistically, i.e.

$$R(kT + T) = \{R(kT) \quad \text{but not } R'(kT) \text{ else } R''(kT)\} \tag{12}$$

where $R(kT)$ is the current controller relation matrix and $R(kT + T)$ is the modified one. In terms of the set operations "∩" and "∪", equation (12) takes the form

$$R(kT + T) = \{R(kT) \cap \sim R'(kT)\} \cup R''(kT) \tag{13}$$

The above results for the SISO SOC controller can be easily extended to the multivariable case.

3.3 Fuzzy PID supervisor

This type of control was introduced by van Nauta Lemke and Dezhao (1985), and consists of a conventional PID controller whose settings are adjusted by a fuzzy supervisor, as shown in fig. 7a,b. The fuzzy supervisor is designed on the basis of the intuition and knowledge of an experienced human operator to improve the performance of the resulting overall supervised PID controller. The architecture of the fuzzy supervisor has the form of fig. 7b, involving as usual a fuzzifier, a set of fuzzy rules and a defuzzifier. The supervisor accepts the error e and the change-in-error de/dt and adjusts the proportional (P), integral (I) and derivative (D) settings according to the relations

$$P = P_o + \Delta P, \quad I = I_o + \Delta I, \quad D = D_o + \Delta D$$

where P_o, I_o and D_o are the initial settings of the controller, and ΔP, ΔI and ΔD are their adjustments (corrections). As shown in fig. 7b, five scaling factors (gains) are used in the supervision, namely K_e, $K_{\dot{e}}$, K_P, K_I and K_D, which are defined by

$$x = \frac{1}{|r|} K_e e, \quad \dot{x} = \frac{1}{|r|} K_{\dot{e}} \dot{e}$$

$$\Delta P = K_P P_f, \quad \Delta I = K_I I_f, \quad \Delta D = K_D D_f$$

with $|r| = 0.1$ for $-0.1 < r < 0.1$.

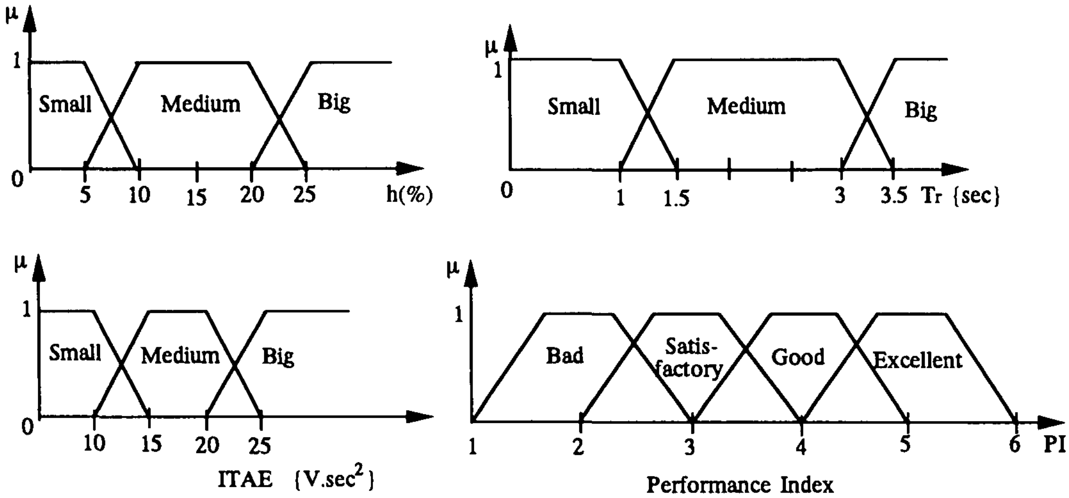


Figure 8 Fuzzy subsets for h, $ITAE$, T_r and performance index

The relative error x and its derivative $\dot{x}$ are used in place of e and $\dot{e}$ to obtain an acceptable performance for a wide range of values of the reference input. The fuzzifier determines the fuzzy regions for the inputs, and after implementation of the rules of these regions, the resulting fuzzy subset is defuzzified to modify the settings of the controller by ΔP, ΔI and ΔD. The defuzzification rule applied by Van Nauta Lemke and Dezhao is the weighted-sum rule

$$V_f = \sum_{i=1}^{m} r_i v_i \Big/ \sum_{i=1}^{m} r_i$$

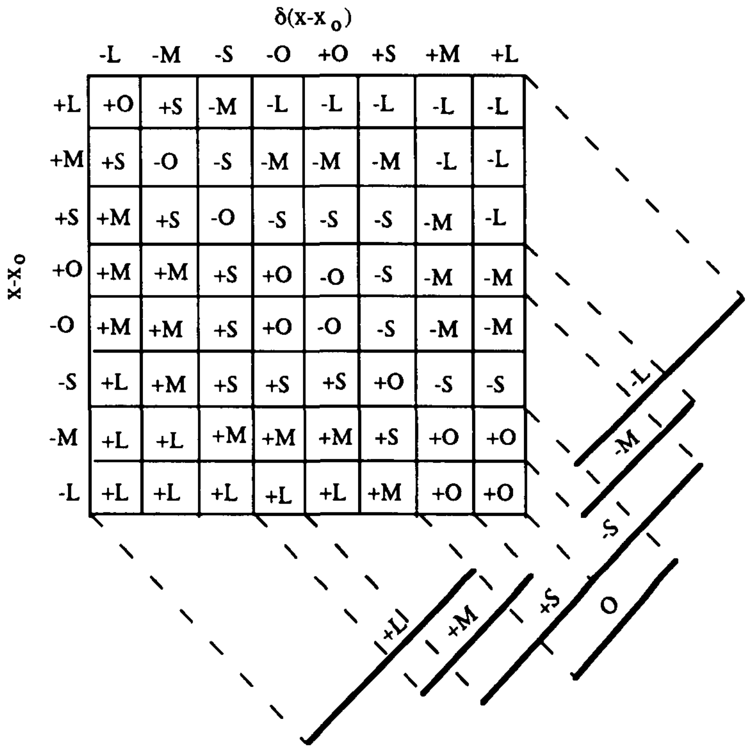
where v_i is the geometric centre of the component $\mu_{ci}(v)$ in the membership function

$$\mu_c(v) = \sum_{i=1}^{m} r_i \mu_{ci}(v)$$

Other defuzzification rules that can be used are those of section 2.10 or the "centre-of-gravity" rule and the "equal-surface" rule (Van Nauta Lemke and Dezhao, 1985).

The performance index is in line with human judgment, and uses the rise time T_r, the overshoot h and the ITAE (Integrated Time Absolute Error) criterion to qualify the overall performance of the controlled system as *bad*, *satisfactory*, *good* and *excellent*. The selected fuzzy subsets for T_r, h, $ITAE$ and the overall performance index (PI) are shown in fig. 8, from which the following linguistic rules are obtained:

R1: If h is *big* OR $ITAE$ is *big*
THEN the performance is *bad*
R2: If h is *medium* AND $ITAE$ is *medium*
THEN the performance is *satisfactory*
R3: If h is *medium* AND $ITAE$ is *small* AND T_r is *big*
THEN the performance is *satisfactory*
R4: If h is *small* AND $ITAE$ is *medium* AND T_r is NOT *big*
THEN the performance is *good*
R5: If h is *medium* AND $ITAE$ is *small* AND T_r is NOT *big*
THEN the performance is *good*
R6: If h is *small* AND $ITAE$ is *small*
THEN the performance is *excellent*



$x-x_0$ \ $\delta(x-x_0)$	-L	-M	-S	-O	+O	+S	+M	+L
+L	+O	+S	-M	-L	-L	-L	-L	-L
+M	+S	-O	-S	-M	-M	-M	-L	-L
+S	+M	+S	-O	-S	-S	-S	-M	-L
+O	+M	+M	+S	+O	-O	-S	-M	-M
-O	+M	+M	+S	+O	-O	-S	-M	-M
-S	+L	+M	+S	+S	+S	+O	-S	-S
-M	+L	+L	+M	+M	+M	+S	+O	+O
-L	+L	+L	+L	+L	+L	+M	+O	+O

Figure 9 MacVicar-Whelan fuzzy matrix

As usual, the performance index is computed by the fuzzy Cartesian product of the above linguistic rules and the weighted sum defuzzifier rule.

3.4 Fuzzy PID incremental controller

This controller (Tzafestas and Papanikolopoulos, 1988, 1990) is based on the same basic philosophy as the fuzzy PID supervisor presented in section 3.3. The difference is that the controller parameter changes are determined using MacVicar-Whelan's table (MacVicar-Whelan, 1976). The PID controller settings are initially determined by some standard nonfuzzy technique such as the Ziegler–Nichols technique, the analytical technique, the Kalman digital technique, etc. (Tzafestas and Papanikolopoulos, 1988, 1990). The idea is to slightly change these settings during the system transient so as to improve the step response characteristics.

MacVicar-Whelan (1976) observed that the fuzzy control matrices of King and Mamdani (1975) were not complete. There are situations where the action to be taken is not defined. That is, there is not always agreement about just what value the dependent variable should take for a given value of the independent variable (i.e. the error) and its derivative.

The fuzzy control matrix proposed by MacVicar-Whelan is shown in fig. 9, and is based on the following principles:

- If the output has the desired value and the error derivative is zero, then we keep the output of the controller constant.
- If the output diverges from the desired value, then our action depends upon the signum and the value of the error and its derivative. If the conditions are such that the error can be corrected quickly by itself, then we keep the controller output constant or almost constant. Otherwise, we change the controller output to achieve satisfactory results.

Most human controllers act by following the philosophy of this matrix. The logic of this matrix approaches human logic more closely than other fuzzy control matrices. This matrix also shows something very fundamental. The designer, prior to designing a fuzzy controller, should study the

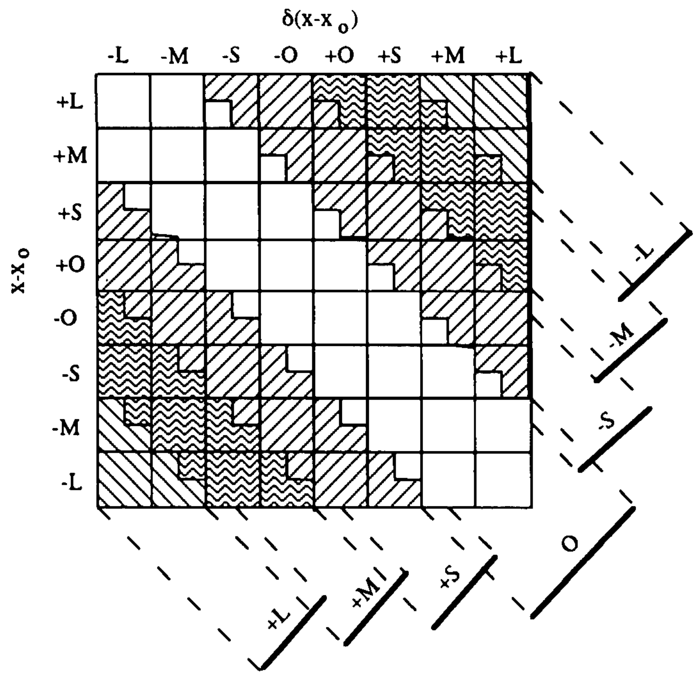


Figure 10 Increasing the control resolution by a factor of 2

actions of the human controller, in order to be able to give a more realistic behaviour to the fuzzy controller. Note that if we project the policy of the matrix on an axis perpendicular to the main diagonal, then we will see clearly the form of the policy. One can increase the resolution of the matrix by a factor of 2. This increases the number of projected parts and reduces the fuzziness of the control policy (fig. 10). The analysis of the projections of the boundaries can be regarded as a study of the variance of the neutral point and the amplitude of the fuzzy function μ_A. MacVicar-Whelan has asserted that using this analysis one can predict the resolution of the projections of the boundaries. An alternative solution would be to start with a small resolution and slowly increase it until we get the desired performance.

This matrix was not used by others, probably because it implements the fuzzy controller in a direct way, and so for each particular case with special requirements it should be varied. Another drawback is the small number of quantization levels used. However, we believe that the logic and structure of this matrix are valuable, and thus we used it for improving existing analogue and digital PID tuning procedures, as described below.

The rules used for slightly changing the controller parameter settings during transient are empirical and are involved in a fuzzy 14 × 14 control matrix. The problems which must be faced refer to the *stability*, *computation time*, *cost*, *quantization* and *generality* of the approach.

The main issues of these control rules are as follows:

(a) Since the integral term is responsible for the overshoot, slightly decreasing it at the moment when the system response exceeds the value 1, one can reduce considerably the overshoot. On the other hand, a small increase of the integral term during the rising of the response leads to a 10–20% improvement of the rise time.
(b) Since the derivative term is responsible for the flatness of the step response, a small increase of it during rising and in the steady state eliminates the small oscillations that usually occur.
(c) Increasing the proportional term leads to reduced rise time and increased oscillations. Thus, finally, this term should be decreased to avoid oscillatory behaviour.

$\Delta E(X_o\text{-}X)$

E	-6	-5	-4	-3	-2	-1	-0	+0	+1	+2	+3	+4	+5	+6
-6	-6	-6	-5	-5	-4	-4	-3	-3	-2	-2	-1	-1	0	0
-5	-6	-5	-5	-4	-4	-3	-3	-2	-2	-1	-1	0	0	0
-4	-5	-5	-4	-4	-3	-3	-2	-2	-1	-1	0	0	0	+1
-3	-5	-4	-4	-3	-3	-2	-2	-1	-1	0	0	0	+1	+1
-2	-4	-4	-3	-3	-2	-2	-1	-1	0	0	0	+1	+1	+2
-1	-4	-3	-3	-2	-2	-1	-1	0	0	0	+1	+1	+2	+2
-0	-3	-3	-2	-2	-1	-1	0	0	0	+1	+1	+2	+2	+3
+0	-3	-2	-2	-1	-1	0	0	0	+1	+1	+2	+2	+3	+3
+1	-2	-2	-1	-1	0	0	0	+1	+1	+2	+2	+3	+3	+4
+2	-2	-1	-1	0	0	0	+1	+1	+2	+2	+3	+3	+4	+4
+3	-1	-1	0	0	0	+1	+1	+2	+2	+3	+3	+4	+4	+5
+4	-1	0	0	0	+1	+1	+2	+2	+3	+3	+4	+4	+5	+5
+5	0	0	0	+1	+1	+2	+2	+3	+3	+4	+4	+5	+5	+6
+6	0	0	+1	+1	+2	+2	+3	+3	+4	+4	+5	+5	+6	+6

Figure 11 The present fuzzy PID control matrix

All these issues are experiential observations of human controllers. Our problem was to accommodate them into a matrix and find the criteria of actually carrying out these changes. The amount of increase/decrease of the proportional, integral and derivative terms is expressed by three coefficients, k_1, k_2 and k_3.

The fuzzy PID control matrix is shown in fig. 11. The basis for the control rules is the error $E = X_o - X(X_o$ is the desired and X the actual system output) and its derivative $\Delta E = \Delta(X_o - X)$. The sampling of E and ΔE should be made at small time intervals of the order 0.1–0.2 s.

This table involves 14 quantization levels for both E and ΔE. This ensures a closer follow-up of the unit step response. The linguistic code is

+ ⇔ positive change	− ⇔ negative change	1 ⇔ extra small
6 ⇔ extra large	4 ⇔ big	0 ⇔ zero (null)
5 ⇔ large	3 ⇔ medium	
	2 ⇔ small	

The changes of the three terms, which showed good success, are given by the expressions

$$\begin{aligned} P &= P + CV\{E, \Delta E\} \times k_1 \quad (\textit{Proportional}) \\ I &= I + CV\{E, \Delta E\} \times k_2 \quad (\textit{Integral}) \\ D &= D + CV\{E, \Delta E\} \times k_3 \quad (\textit{Derivative}) \end{aligned} \tag{14}$$

The parameters k_1, k_2 and k_3 play an important role in the whole procedure, since they determine the range of variation of each term. For example, if some tuning method ensures a very small rise time and large overshoot, then the integral term should have a large range of variation, whereas the other terms can remain unchanged. This range of variation should be matched with the stability intervals to guarantee stability. Thus the values of k_1, k_2 and k_3 are determined from both the stability analysis and the particular characteristics of the closed-loop response. In general, the parameters k_1, k_2 and k_3 provide great flexibility and can be used in conjunction with all the

$\Delta E(X_o\text{-}X)$

E \ ΔE	-6	-5	-4	-3	-2	-1	-0	+0	+1	+2	+3	+4	+5	+6
-6	-1.79	-1.79	-1.60	-1.60	-1.38	-1.38	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00
-5	-1.79	-1.60	-1.60	-1.38	-1.38	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00
-4	-1.60	-1.60	-1.38	-1.38	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00
-3	-1.60	-1.38	-1.38	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-2	-1.38	-1.38	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69
-1	-1.38	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69
-0	-1.10	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10
+0	-1.10	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	0.69	+1.10	+1.10
+1	-0.69	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10	+1.10	+1.38
+2	-0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10	+1.10	+1.38	+1.38
+3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10	+1.10	+1.38	+1.38	+1.60
+4	0.00	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10	+1.10	+1.38	+1.38	+1.60	+1.60
+5	0.00	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10	+1.10	+1.38	+1.38	+1.60	+1.60	+1.79
+6	0.00	0.00	0.00	0.00	+0.69	+0.69	+1.10	+1.10	+1.38	+1.38	+1.60	+1.60	+1.79	+1.79

Figure 12 An alternative fuzzy PID control matrix

available PID tuning algorithms. We have also used the fuzzy control matrix of fig. 12. The only difference with the matrix of fig. 11 is that here the Neperian logarithm of $|CV\{E, \Delta E\}|$ is used. This ensures greater stability around the set point and smoother transition from one value to another. The results obtained using this table are also satisfactory, but the values of k_1, k_2 and k_3 are found to be larger.

The stability can be tested by various techniques. In our studies we used Routh's table in the continuous-time case and the corresponding table, obtained through the bilinear transformation $z = (w + 1)/(w - 1)$ in the discrete-time case. The most serious problem faced in our experiments was that of quantization. The right quantization of E and ΔE ensures a good evaluation of the controller performance. We found logarithmic quantization to be the most successful. We have used the following quantization levels: $Q_{\max} = 0.02$, $DQ_{\max} = 0.02$. Let us see in detail the logarithmic quantization. If the error E is greater than $Q_{\max}/2$, then it is quantized in a value between +1 and +6. If $E < -Q_{\max}/2$, then it is quantized in the interval between -6 and -1. If $0 < E < Q_{\max}/2$, then it is quantized to +0. If $-Q_{\max}/2 < E < 0$, then it is quantized to -0. Analogous quantization is applied to ΔE. It should be noted that these quantization levels are sufficiently large to cover the presence of noise. However, this quantization behaved quite well.

4 Survey of theoretical results

Having obtained an adequate background knowledge on fuzzy reasoning and fuzzy or linguistic controllers, we now proceed to a survey of the theoretical results and applications of fuzzy system control. Unavoidably, this survey will not be exhaustive, but will include most of the representative works of the field. All comments and observations are left for the conclusions.

4.1 Fuzzy systems

Chang and Zadeh (1972) introduce three concepts: (i) an observation operator through which knowledge of the state of the controlled process is made more precise; (ii) a control goal G, a fuzzy set on the state space X; and (iii) a process description, i.e. next state mapping $f: X \times U \rightarrow X$, observation operator $O: y = xO$ and control mapping $g: y \rightarrow U$. However, they do not solve the fuzzy control problem for a given system and goal. Negoita and Ralescu (1975) follow the algebraic approach through the next state mapping $f: F(X) \times F(U) \rightarrow F(X)$ and the output mapping

$g: F(X) \rightarrow F(Y)$, where $F(X)$, $F(U)$ and $F(Y)$ are the sets of all fuzzy subsets of the state space X, the input space U and the output space Y, respectively, to extend the nonfuzzy concepts of reachability, observability and stability to fuzzy systems. The fuzzy system is *reachable* if f is "*onto*", it is observable if g is "*one-to-one*", and finally it is *stable* if the state trajectory can be kept within a given area by restricting the initial state to lie within another given area. Clearly, this stability definition is similar to the Lyapunov stability concept.

Pappis and Sugeno (1976) present a general methodology for solving basic forms of fuzzy relational equation, and Thomason (1977) derives sufficient conditions for the convergence of fuzzy relation matrices. Tong (1976) considers the question, how is the structure of a fuzzy algorithm affected by changes in the implementation? The conclusions drawn are the following: (i) the fuzzy algorithm is not altered, but the implied model of the process is changed; (ii) the implementation is fixed by the required quality of the modified model; and (iii) the algorithm is developed and modified only by changing the rules themselves. This means that to optimize the performance of a fuzzy closed-loop system we must modify the rules.

The implementation is important in fixing the observed behaviour of the process, and so it should be selected with care. Having chosen it, it should not be changed. The controller design problem is actually that of determining an algorithm with a structure which, when combined with the process, produces the desired closed-loop response.

Kickert and Mamdani (1978) use describing functions for single-input single-output fuzzy systems under certain conditions. Braae and Rutherford (1979) develop an algebraic model of fuzzy logic controllers and use it to study the loop stability and controller adjustments associated with practical applications.

Tong (1977) presents a method for analyzing closed-loop fuzzy systems through finite discrete relations. Use is made of Zadeh's max-min composition rule {see equations (5) or (6)} and of a fuzzy error function. The concepts of controllability and stability are defined using a kind of equivalence between fuzzy sets.

Hirota (1979, 1981) introduces the concept of probabilistic set which is tied up with the efforts to overcome difficulties in determining a precise value for grades of membership forming the membership functions. Instead of treating a fuzzy set as a mapping of X into the [0, 1] interval, the probabilistic set is characterized by a mapping of $X \times \Omega$ into [0, 1], where Ω plays the same role as in probability theory. An interesting view of probabilistic sets can be obtained by averaging over the space Ω. Two important functions of the probabilistic set A are:

Mean-value membership function $E(A)$:

$$E(A)(x) = \int_{\Omega} A(x, \omega)\, \mathrm{d}P(\omega) = m_1(A)x$$

Vagueness function $V(A)$:

$$V(A)(x) = \int_{\Omega} (A(x, \omega) - m_1(x))^2\, \mathrm{d}P(\omega)$$

The standard deviation $\sigma(A)$ is obtained as usual by $\sigma(A)(x) = \{V(A)(x)\}^{1/2}$.

Mizumoto (1985) has proposed an extension of Zadeh's (1970) *generalized modus ponens* rule

$$((A \rightarrow B) \wedge A') \rightarrow B'$$

where A and A' are fuzzy sets in the universe of discourse U, and B, B' are fuzzy sets in the universe of discourse V. Mizumoto's extension is

$$\{[(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \rightarrow B] \wedge (A_1' \wedge A_2' \wedge \ldots \wedge A_n')\} \rightarrow B'$$

which can again be implemented using Zadeh's max-min composition operator.

The concept of fuzzy relation and its properties and/or extensions has been studied by several workers, among which DiNola et al. (1985), Higashi et al. (1984), Miyakoshi and Shimbo (1985, 1986) and Pappis and Sugeno (1985) are noteworthy.

A scheme of self-organizing fuzzy control other than that described in section 3.2 was proposed by Bartolini et al. (1985), and applied to a continuous casting plant.

Further developments in the fuzzy controller involve more advanced concepts and architectures, such as the many-valued logic (e.g. Lukasiewicz logic) and their fuzzy-set-based extensions. An extended (fuzzified) version of Lukasiewicz logic) was presented by Tsukamoto (1979) and Pedrycz (1980). In Lukasiewicz's three-valued logic the following result is true:

$$r = (1 - p + q) \wedge 1 \tag{15}$$

where p and q denote the values of propositions, namely antecedent P and consequent Q, respectively, while r is the value of the implication $P \rightarrow Q$ (P implies Q). All the truth values are treated as pointwise values lying in the interval [0, 1]. After calculation, Tsukamoto (1979) reached the following conclusions;

- If $r = 1$, then $q \in [p, 1]$
- If $0 \leqslant p + r - 1 < 1$and $r = 1$, *then* $q = p + r - 1$
- If $p + r < 1$, there is no solution.

The set of solutions of (15) can be obtained as a multivalued mapping, $[0, 1] \times [0, 1] \rightarrow 2^{[0,1]}$, so that

$$q = G(p, r) = \begin{cases} p + r - 1, & \text{if } 0 \leqslant \text{p} + \text{r} - 1 < 1, r = 1 \\ [p, 1], & \text{if } r = 1 \\ 0, & \text{otherwise} \end{cases}$$

Thus, assuming that the extended Lukasiewics logic, where the truth values of P, Q and R are fuzzy sets defined in the unit interval, is used, Q is calculated by applying the extension principle to the multivalued mapping $G(p, r)$, i.e.:

$$Q(q) = \sup_{(p,q) \in [0,1] \times [0,1]: G(p,r) = q} [P(p) \wedge R(r)], \quad q \in [0, 1]$$

Starting from the observation that the small perurbations in the membership function have minor effect on the inferences (control variables, etc.), Mizumoto and Tanaka (1976) have developed the concept of *similarity* of the inputs and influences in a fuzzy reasoning (or control) system.

Definition 8

Consider the fuzzy subsets A and A' of the universe of discourse U with membership functions $\mu_A(x)$ and $\mu_{A'}(x)$, respectively. Then A and A' are said to be similar, notationally $A \approx A'$, if $\|\mu_A - \mu_{A'}\| \leqslant \varepsilon$, where ε is a small positive number, and the norm $\|\mu_A - \mu_{A'}\|$ is given by $\|\mu_A - \mu_{A'}\| = \max_i |\mu_{A_i} - \mu_{A'_i}|$ (the maximum of the absolute values of the difference between all corresponding elements of A and A').

Bounas and King (1994) further study the consequences of this similarity, and prove that the fuzzy operations (union, intersection, etc.) of two fuzzy sets remain unchanged when one of the members is similar. Specifically, the following have been shown to be true (Bounas and King, 1994):

1. (Intersection) If A, A' and B are fuzzy subsets of U, and $A \approx A'$, then

   $$A \cap B \approx A' \cap B,$$

2. (Union) If A, A' and B are fuzzy subserts of U, and $A \approx A'$, then $A \cup B \approx A' \cup B$.
3. (Product) If A, A' and B are fuzzy subsets of U, and $A \approx A'$, then $A \cdot B \approx A' \cdot B$.
4. (max-min) (a) If $A \approx A'$ then $A \circ R \approx A' \circ R$, and (b) if $R \approx R'$ then $A \circ R \approx A \circ R'$, where R and R' are fuzzy relations over two universes of discourse U and V.

Bounas (1993) considered several fuzzy conditional propositions, and proved that they preserve a specific ordering useful in fuzzy reasoning processes.

Celmins (1987) has studied the fuzzification of the least squares problem and developed an elegant but quite involved (computationally) method along with a software package that implements it. An alternative least squares method was given by Tzafestas et al. (1994), where use is made of the concept and the algebraic properties of the so-called "L-R fuzzy numbers" (Dubois and Prade, 1978). Minimizing their distance with respect to the unknown parameters reduces to a classical minimization problem, and has small computational demand. Other works on fuzzy estimation include Bardossy (1990), Diamond (1990), Jajuga (1986), Jossef (1992), Tanaka (1982, 1988) and Tanaka and Ishibuchi (1991). Besides the fuzzy parameter estimation, many authors have considered and solved several problems of fuzzy state estimation (Bertsekas, 1971; Schweppe, 1968; Sira-Ramirez, 1979, 1980; Witsenhaussen, 1968).

Petri nets have been developed for information flow by Petri (1962), and extended by various authors (e.g. Holt and Commoner, 1970; Peterson, 1981; Reisig, 1985). Petri nets are used to model flows such as parts and materials in manufacturing processes (Zhou and Leu, 1991), and logical truths in machine reasoning systems (Looney, 1988a; Murata et al., 1991).

A *logic Petri net P* consists of (i) a *net architecture* and (ii) an *operational procedure*. Architecturally, it is a directed graph that consists of two kinds of nodes: (a) *conditions*, designated by circles, and (b) *events*, denoted by bars. The conditions and events are connected by *arrows* according to two rules: (1) an arrow may connect from a condition to an event, or from an event to a condition; and (2) an arrow may never connect two nodes of the same kind.

Procedurally, the net uses tokens, represented graphically by dots inside condition nodes, to activate the conditions where they appear. A token denotes a Boolean truth value of 1 for the condition, while the absence of a token indicates a value of 0. An event is *enabled* when every arrow entering it comes from a condition that contains a token. An enabled event *fires* to *activate* (i.e. make true) all conditions to which its departing arrows directly connect, by sending them tokens. Looney (1988b, 1994) develops a class of fuzzy Petri nets together with a suitable fuzzy Petri net algorithm.

The fuzzy truth of a fuzzy Petri net with n conditions $C_1, C_2, \ldots, C_n$ is the current vector of fuzzy markings $M = (m_1, m_2, \ldots, m_n)$, where m_i is the current fuzzy truth of C_i. Let $E = (e_1, e_2, \ldots, e_m)$ be the vector of event enablings, where the Event E_j has the enabling value e_j given by the minimum of all marking values (activations) of its preset conditions in $*E_j$. An algorithm to propagate the initial fuzzy truths of the conditions through the system needs two stages: (i) for each Event E_j, $j = 1, \ldots, m$, get the markings of the conditions in the preset $*E_j$, compute the event enabling value e_j by minimizing these markings, and then compute the activating value $a_j = \min\{e_j, f_j\}$; and (ii) for each Condition C_j, get the activating values of its preset events in $*C_i$ and compute the new marking m as the maximum of the current marking m_i of C_i and all activations from preset events. A fuzzy Petri net is a model for decision making via a fuzzy rule-based system. It is not a traditional fuzzy controller. The fuzzification and defuzzification processes are simplified by using proportional mappings. Fuzzy Petri nets are also discussed in (Tzafestas, 1990a,b).

4.2 Neuro-fuzzy systems

The problem of combining neural networks with fuzzy systems, in an attempt to arrive at better performance of decision making processes, is attracting increasing attention (Cohen and Hudson, 1992; Keller and Tahani, 1992; Kosko, 1991; Kuncicky and Kandel, 1989; Pao, 1989; Takagi, 1990). The uncertainties involved in the inputs and outputs of the decision process are taken care of by using fuzzy set representations, while the neural nets are used for generating the required (linearly nonseparable) decision regions. Mitra and Pal (1994) and Takagi (1990) present useful reviews of the theoretical results in this direction.

Neural networks or connectionist models are designed to emulate human performance at high computation rates through appropriate massively parallel interconnections of simple neurons. On the other hand, fuzzy systems are capable of modelling uncertain situations that occur in real life environments. One way of implementing fuzzy reasoning systems is through expert system architectures, the alternative being through neural network structures. Neural systems and fuzzy systems have essential differences in the way in which they function. Neural systems are based on pattern recognition supervised or unsupervised learning techniques, whereas fuzzy systems use fuzzy logic and fuzzy reasoning mechanisms. However, neural and fuzzy systems have two important characteristics in common; namely, they both possess a high degree of parallelism and they both belong to the class of the model free estimators. Therefore, the combination of fuzzy and neural systems is possible, and indeed, there are currently several types of fuzzy-neural systems (Berenji, 1992; Nakanishi et al., 1990; Uehara and Fujise, 1990; Yamakawa, 1990). Keller and Tahani (1992) use a multi-layer perception (MLP) model which receives the possibility at the input, uses a hidden layer to generate an internal representation of the relationship, and finally produces the possibility distribution of the consequent at the output. The model is expected to function as an inference engine with each small sub-network learning the functional input-output relationship of a rule. Conjunctive antecedent clauses are also modelled using separate groups of hidden nodes for each clause. Keller et al. (1992) explicitly encode the knowledge of each rule among the connection weights of the neural net. A measure of disagreement between the input possibility distribution and the antecedent clause distribution is used at the *clause-checking* and *combination* layers to determine the uncertainty in the consequent part of the fired rule. Tzafestas et al. (1994) provide an improvement of the Keller–Yager–Tahani neuro fuzzy model (Keller et al., 1992), and use Hamacher's intersection function and Sugeno's complement function (Klir and Folger, 1988). Their system can accept any meaningful fuzzy intersection, union or complement function. A modified version of the back propagation algorithm is used in Ishibuchi et al. (1992) to allow the introduction of interval arithmetic. This is needed since fuzzy inputs and outputs in an MLP are represented through the use of interval vectors. Different fuzzy If-Then rules are interpolated from a few sample rules used during training. Okada et al. (1992) treat the bond rating problem of investors by interpreting the weights and threshold levels of the neurons after training as adaptations and fuzzy rules.

Masuoka et al. (1990) build and preweight a structured neural network using knowledge represented by fuzzy membership functions and fuzzy (AND/OR) rules extracted from experts. This neural model which consists of the input variable membership net, the rule net and the output variable net, is then tuned using selected learning data. Modified fuzzy rules, extracted from the trained neural network using pruning, can then be evaluated and inappropriate rules corrected using relearning. Yager (1992) deals with the modelling of fuzzy logic controller rules by using membership neural modules for the antecedents, inverse membership modules for the consequents, and a rule neural module with a combiner (via min or product functions). The various weights are learned and the importance of the antecedent clauses simulated.

Takagi and Hayashi (1991) present a fuzzy reasoning scheme that is able to learn the membership function of the IF part, and determine the amount of control in the THEN part of inference rules. The input data is clustered for finding the optimal number of partitions corresponding to the number of inference rules applicable to the reasoning problem. Each rule is modelled by a single neural net block. To avoid overlearning, the optimum number of cycles required is determined, and the minimum number of input variables for inferring the control values is selected. The role of fuzzy logic in the control of the activation, training, reliability and performance of neural networks has been studied by Yen (1990). He adopted a hybrid architecture, and used fuzzy rules to detect situations under which certain actions need to be invoked for neural network modules based on their performance measures. A production system which takes into account the degree of partial matching of the fuzzy action rules enables the system to perform in a robust manner, even in the face of incomplete or noisy data.

Two types of practical and intelligent solving methods of fuzzy networks are the frame knowledge representation method (Kawamura et al., 1990), and the inference tree method (Kawamura et al., 1991, 1992). In the first, the most terminal variables or states are searched out (backward inference) by using suitably designed tables, where a certain variable or state is denoted as a final goal. In the second method, a hierarchical tree with the top event of the final goal is constructed, and the most terminal variables are pointed out according to given rules (backward inference). When the most terminal variables are given, the goal is obtained according to the max-min operation (forward inference).

Zhuang et al. (1990) provide a fuzzy inference method for controlling a mathematically intractable system, using a multilayer network with each node standing for a fuzzy subset of a linguistic variable. The network weights are associated with ordered pairs of real numbers lying in the interval $[-1, 1]$, and indicate the certainty factors used to represent the reliability of the control rules. The fuzzy operations are implemented at the neuron levels.

5 Survey of applications and software systems

5.1 Applications

On the application side, there is a vast amount of work. MacVicar-Whelan (1975) provided a study of the classification of human heights by humans into the qualitative classes: very very short, very short, short, tall, very tall and very very tall. Several experiments were performed to better understand classification of height. Siy and Chen (1974) applied Zadeh's fuzzy reasoning concepts for recognizing hand-written characters. The accuracy obtained in recognizing an IEEE database of 500 samples was 98.4%. Character recognition is complicated by the context and learning aspects. The application that has received the greatest attention is that of controlling industrial and other technological processes. In a series of papers, Mamdani and his co-workers (1974, 1975, 1976, 1977, 1979) and Mandič et al. (1985) developed various fuzzy controllers for several real systems such as steam engines, boilers and robotic manipulators. Ostergaard (1976) applied fuzzy control to heat exchangers, and Jensen (1979), Larsen (1980) and Umbers and King (1980) considered fuzzy control of cement kilns. Kickert and Van Nauta Lemke (1976) considered warm water plants, and Van Amerongen, Van Nauta Lemke and Van der Veen (1977) used the fuzzy controller for a ship autopilot. Furthermore, Tong, Beck and Latten (1980) examined the behaviour of an experimental fuzzy controller applied to the actuated sludge wastewater treatment process, Yasunobu and Hasegawa (1986) developed a predictive fuzzy controller and applied it to an automatic container crane operation, and Hirota, Arai and Hachisu (1986) investigated the moving mark recognition and moving object manipulation problem of a fuzzy controlled robot. They constructed an overall real time controller utilizing one 16-bit personal computer.

Actually, the fuzzy controller is a tool for processing a fuzzy form of information in a nonfuzzy or fuzzy scheme of reasoning. Two applications of fuzzy controllers in the so-called "soft" sciences, where the human plays a central role, were given by Wenstop (1976) and Kickert (1978). The first provides a model of organizational behaviour using linguistic rules that describe the relationships of a mining company. The second provides a model of the sociopsychological power theory of Mulder that has been analysed in the framework of fuzzy sets. The class of fuzzy linear programming and decision making problems with multiple objectives has been considered by Sakawa and Yano (1983, 1986).

A considerable effort has been concentrated since the mid 1980s on the application of fuzzy control techniques to industrial and other robots. This work can possibly be classified according to the particular type of control designed as follows (Gorez and DeNeyer, 1994):

- position control of robots and servo systems (Hirota et al., 1989; Kwok et al., 1990; Li and Lau, 1989; Lin and Sheu, 1992; Murakami et al., 1989; Palm, 1989; Scharf, 1985; Tzafestas, 1991; Wakileh and Gille, 1988) including self-organizing control (Koh et al., 1990; Mandic et al., 1985; Scharf and Mandic, 1985; Swevers, 1992; Transcheit and Scharf, 1988).
- force and force-position control of robotic manipulators, where in some cases there is a two-level hierarchical structure where fuzzy logic is used only at a high level of coordination (Deepak et al., 1989; Dote et al., 1990; Han-Gyoo, 1991; Hong et al., 1991; Kang and Vachtsevanos, 1990; Seung-Woo and Mignon, 1991; Vachtsevanos et al., 1987; Xu et al., 1991).

- motion control of mobile robots and unmanned vehicles (Brown et al., 1991; Dai et al., 1992; Deyong et al., 1992; Gasos et al., 1990; Harris and Brown, 1991; Ishii and Misra, 1990; Ishikawa, 1991; Isik and Meystel, 1988; Kemal, 1989; Kubota and Hashimoto, 1990; Takeuchi et al., 1988, 1991; Tzafestas, 1991; Tzafestas and Stamou, 1994).
- path generation of robots including coordinate transformations, teleoperators and operation planning in automated manufacturing systems (Aliev et al., 1986; Cassinis et al., 1988; Dodds, 1988; Kalley and McDermott, 1989; Nedungadi and Wenzel, 1991; Stellakis and Valavanis, 1991 Valavanis and Stellakis, 1991; Zimmermann, 1992).
- use of fuzzy logic in knowledge-based control systems for robotic applications and in modelling and/or monitoring of robotic systems (De Silva and McFarlane, 1989; Dodds, 1988; Fukuda et al., 1991; Garcia-Gerezo et al., 1991; Isik, 1988; Lakov, 1985; Lim and Hiyama, 1991; Stellakis and Valavanis, 1991b; Zimmermann, 1992).
- use of fuzzy logic in vision systems and other sensory devices for robotic applications (Dodds, 1988; Hong et al., 1991; Li, 1986, Tzafestas, Hatzivasiliou and Kaltsounis, 1994; Watanabe et al., 1994).

Finally, some application work on the neuro-fuzzy approach is:

- Detection of low back disorders through MLP and radial basis functions using crisp input output values and considering networks of multiple single-class nets in the process (Bounds et al., 1990).
- Recognition of tumours at various scales, orientations and locations from ultrasonic images (Silverman and Noetzel, 1990). Here the model acts as classifier, and the pixel information is used as input along with the crisp output values.
- Rule-based phomene recognition of Japanese words (Amano and Aritsuka, 1989).
- Modelling of the subjective evaluation of humans using a linguistic technique (Takahashi and Minami, 1989)
- Knowledge-based image interpretation (Zahzah et al., 1992)

5.2 *Software systems*

The software representation problem of fuzzy sets and fuzzy logic has also attracted a great deal of attention, and many software systems based on fuzzy set/logic theory have been developed. These systems can be classified in three categories (Yamamoto, 1994):

- Software for management of uncertainty.
- Fuzzy control shells.
- Fuzzy set representation softwware

The first category includes software systems for the management of uncertainty. One of the earliest works is PRUF (Zadeh, 1978), the aim of which was to represent the uncertainty in natural language in a formal way. Uncertainty or vagueness in natural language was represented via possibility distributions and fuzzy propositions. A second system for management of uncertainty in a programming language is the L.P.L. system (Adamo, 1980a,b). This system has the ability to treat fuzzy predicates and control structures with fuzzy notion.

Another approach for the management of uncertainty is to introduce fuzziness in the PROLOG language. Most fuzzified Prologs are based on fuzzy logic, where the truth values of predicates and clauses are allowed in the interval [0, 1]. Two such systems are the FS-Prolog (Umano, 1987) and F-prolog (Martin et al., 1987). Other systems include the fuzziness in certainty factor or degree of confidence form. A system of this type is FLOPS (Siler, 1987; Buckley and Siler, 1987; Siler and Buckley, 1987), which can be compared to the expert system MYCIN or to the system designed in Tzafestas, Palios and Cholin (1994). Other systems allow rules of the type "*If x is rather short then* . . ." or facts of the type "*John is tall*". A system of this kind which uses fuzzy set and certainty factor theory is the system Z-III (Leung and Lam, 1988).

The second category includes control shells that have been proposed to support the development and execution of fuzzy inference used in fuzzy control systems. The "Fuzzy-C Compiler System" (Teichrow et al., 1989) is one of the most refined systems of this kind. Actually, this system is a translator from FPL (Fuzzy Programming Language) to C language which is oriented toward fuzzy control systems. In general, most of the systems of this kind consist of a graphical user interface for editing fuzzy rule bases and membership functions, and sometimes of a simulator or knowledge acquisition tools working as development tools. Many systems involve dedicated hardware for high-speed processing of fuzzy inference which is a must in real-time control. For example, the Fuzzy-C Compiler system includes the chip FC-110 which works in a machine language and provides fuzzy inference and other fuzzy logic related instructions.

The third category embraces all software systems that are designed for fuzzy set processing in general. These systems are normally based on some standard programming language, and take fuzzy set processing into account. They are designed so as to offer high flexibility in manipulating fuzzy information. The FSTDS (Fuzzy Set Theorectic Data Structure) system (Umano, Mizumoto and Tanaka, 1978), which is an extension of the FORTRAN language, represents one of the earliest attempts in this direction. FSTDS has a pointer-connected data structure, and the data structure makes use of the "paired representation". A second system of this type was later designed by Umano (1987), which is an extension of Lisp and represents fuzzy sets in paired style. This system allows for general representation of fuzzy sets through the use of "tuples" for representing grades of L-fuzzy sets and elements of relations, and also provides a simple way in which to describe fuzzy sets in ordinary Lisp programs. Another similar Lisp based system is FLISP (Sosnawski, 1990, 1991).

6 Some examples

Here a few representative examples will be reviewed to give a picture of the type of results obtained by fuzzy expert control. More details and other application examples can be found in the references.

6.10 Fuzzy control of cement kilns

Cement kilns are industrial processes where fuzzy logic controllers are currently being applied with particular success. This is due to the complexity and inherent vagueness in this class of industrial process (Larsen, 1980; Umbers and King, 1980; King, 1986; King and Karonis, 1986). Cement is produced by heating a slurry consisting of clay, limestone, sand and iron ore to a temperature at which the complex compounds (C_2S, C_3S, C_3A and C_4AF) of the cement are formed. The kiln consist of a long steel shell (~130 m long, ~5 m diameter) mounted at a slight inclination to the horizontal and in line with the fire bricks.

The primary control variables of the cement kiln are the coal-feed rate, the air flow, the kiln feed rate and the kiln rotational speed. The kiln feed rate and the kiln speed are synchronized to obtain an approximately constant filling of material. Thus, since the kiln is operated at constant production, only the air flow and the coal-feed rate are used as control variables. The coal feed rate is actually linked with the oxygen percentage of the exhaust gas through a PI controller. Thus the actual control variables are: the set point of the oxygen percentage which controls the coal-feed rate, and the exhaust gas fan louvre damper position which controls the air flow.

The measured process variables are:

- the kiln drive load (i.e. motor current) that provides an indirect measure of the burning zone temperature,
- the free lime content providing a measure of the clinker quality,
- the oxygen percentage in the smoke chamber,
- the temperature in the smoke chamber

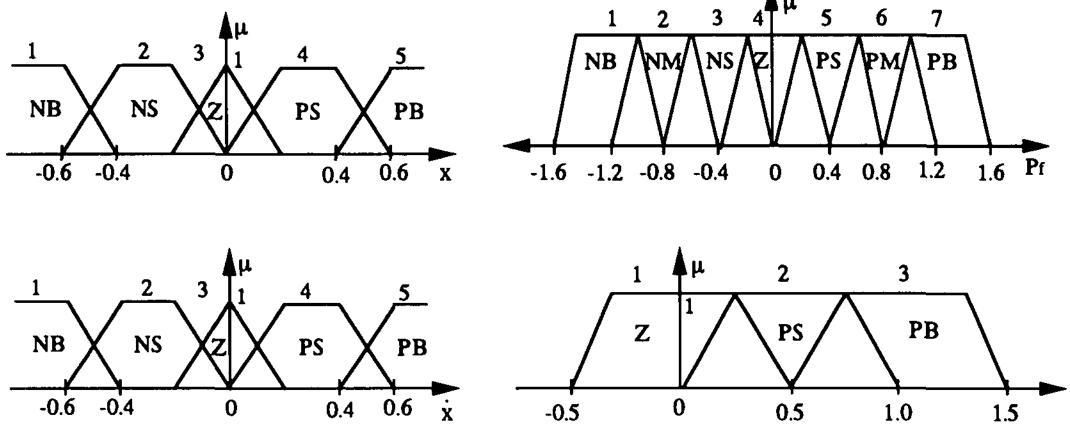


Figure 13 Fuzzy subsets for x, $\dot{x}$, P_f and I_f

The purpose of the fuzzy rule-based controller is to automate the control procedure applied by an experienced operator.

Jensen (1979) proposed a fuzzy control algorithm involving 75 fuzzy rules, which was reduced by Ostergaard (1976) to an algorithm with 13 rules of the type:

IF drive load gradient is (SN, ZE, SP)
  AND drive load is (LN, LP)
  AND free-lime content is (LO, OK, HI)
THEN change burning zone temperature is (LN, MN, SN, ZE, SP, MP, LP)

where the fuzzy terms involved are

LP = large positive, MP = medium positive, SP = small positive, ZP = zero positive, ZE = zero, ZN = zero negative, SN = small negative, MN = medium negative, LN = large negative, HI = high, OK = ok, LO = low.

Jensen (1979) also developed a special Fuzzy Controller Design System (FCDS) which involves an interactive design facility that allows modification of the fuzzy controller at the same time as the system is controlling the process (for details see Larsen, 1980). The general conclusion is that the fuzzy control is better than the control by a human operator followed by a reduction in fuel consumption. In Greece, fuzzy expert control was applied to Heracles' Group cement plan with many reported benefits over classical control (King, 1986; King and Karonis, 1986).

6.2 Experimental results with the fuzzy PID supervisor

The fuzzy PID supervisor (Van Nauta Lemke and De-zhao, 1985) was applied to a process with transfer function

$$G_p(s) = K_{dc}\,e^{-T_d s}/(1 + sT_p)$$

which, among others, provides a simplified model of a cement kiln process. The time constant T_p and the delay time T_d were assumed in the intervals [0.8 s,. 5.2 s] and [0 s, 0.64 s], respectively. The selected fuzzy subsets for the variables x, $\dot{x}$, P_f and I_f are shown in fig. 13.

The initial settings used for a system with $T_p = 1$ s and $T_d = 0.32$ s are $P_o = 1.2$, $I_o = 1.1$ and $D_o = 0$.

The operating conditions of the supervisor are *Run time*: $T_{run} = 10$ s, *Sampling time*: $T_s = 0.08$ s, *Step input amplitude*: 5 V. The performance criteria used are: *Rise time*: T_r, *Overshoot*: h, and

$$ITAE = T_s \sum_{k=0}^{N} k * e(kT_s), \quad N = T_{run}/T_s - 1 = 124$$

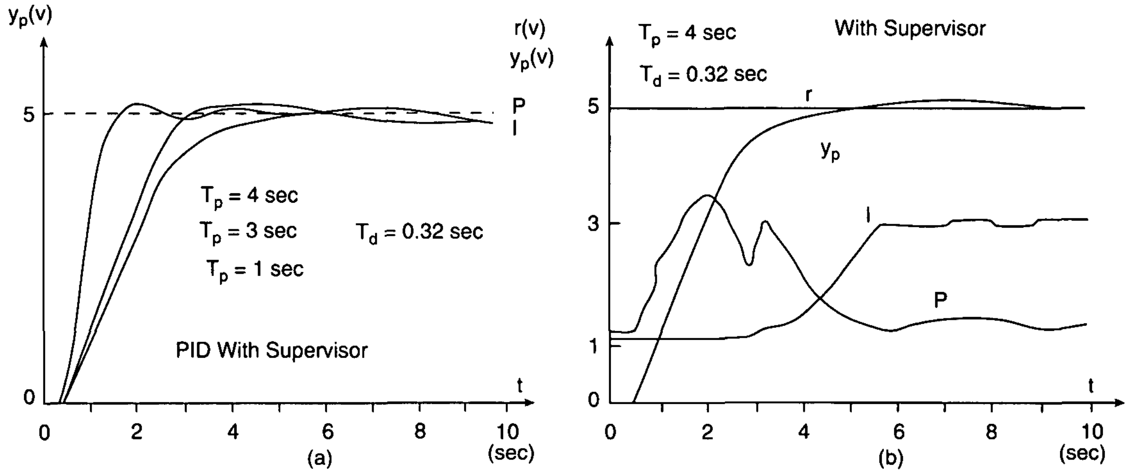


Figure 14 (a) Step response for several time constants; (b) evolution of adjusted P and I

The supervisor settings are

$$K_e = 0.6, \quad K_{\dot{e}} = 1.1, \quad K_p = 5.6, \quad K_I = 2, \quad K_D = 0$$

Some of the result obtained by van Nauta Lemke and Dezhao (1985) are shown in fig. 14.

The response obtained by using a conventional PID controller was shown to be much worse in the cases $T_p = 3.5$ and $T_p = 4.5$.

6.3 Experimental results with the fuzzy PID incremental controller

The fuzzy matrix of fig. 12 was applied to a PID controller tuned through the Ziegler–Nichols technique (Tzafestas and Papanikolopoulos, 1988, 1990). The system under control has three time constants, T_1, T_2 and T_3, i.e.

$$G_2(s) = K/\{(1 + sT_1)(1 + sT_2)(1 + sT_3)\}$$

To determine the critical parameters T_x and K_c, we remove the integral and derivative terms. The resulting characteristic polynomial is

$$A_1s^3 + A_2s^2 + A_3s + A_4$$

with

$$A_1 = T_1T_2T_3$$

$$A_2 = T_1T_2 + T_1T_3 + T_2T_3$$

$$A_3 = T_1 + T_2 + T_3$$

and

$$A_4 = 1 + KK_p$$

Routh's table gives the value

$$K_c = \frac{A_2A_3/A_1 - 1}{K}, \quad T_x = 2\pi\left(\frac{A_2}{1 + KK_c}\right)^{1/2}$$

The parameters of the PID controller

$$u(s) = K_p\left(1 + \tau_d s + \frac{1}{\tau_i s}\right)e(s)$$

are set to

$$K_p = 0.6K_c, \quad \tau_i = T_x/2, \quad \tau_d = \tau_i/4$$

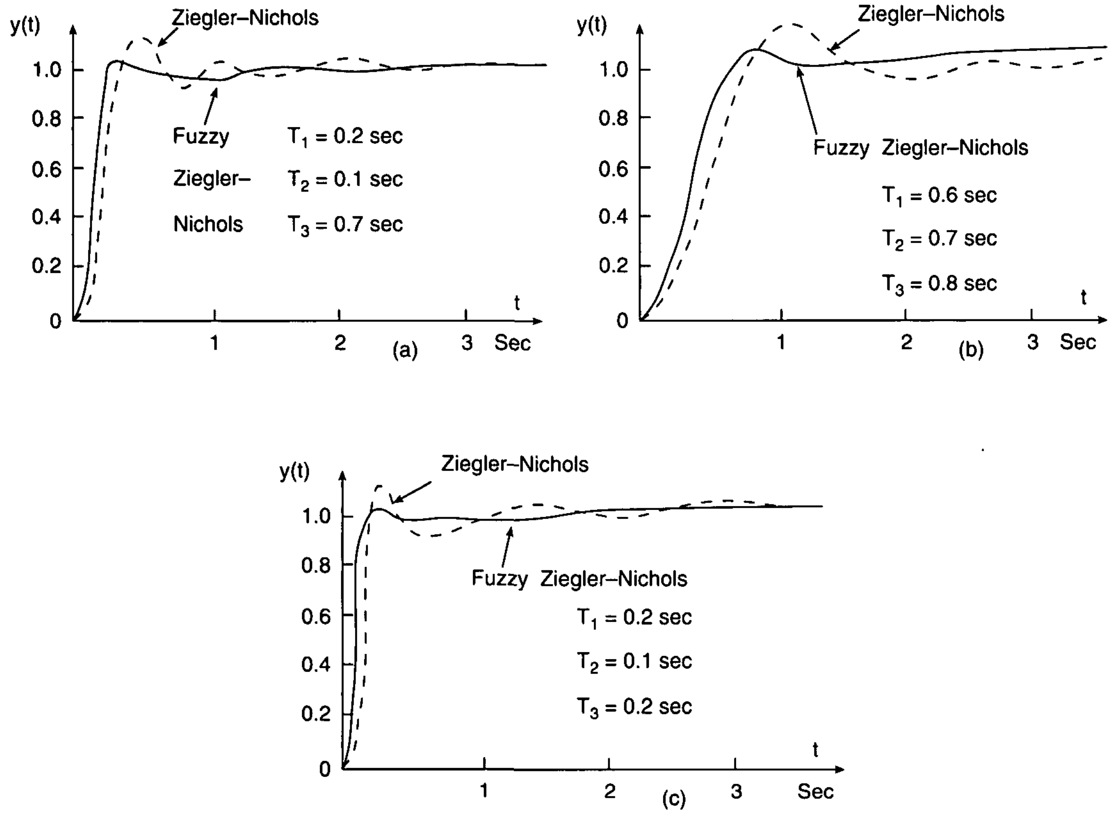


Figure 15 Step response with Ziegler–Nichols parameter tuning and fuzzy control correction

The resulting closed-loop system was simulated with the 4th-order Runge–Kutta technique (dashed curves in fig. 15a–c).

Then use of the fuzzy matrix of fig. 12 was made with the quantization of E and ΔE being made every 0.08 s. To assure stability of the closed-loop system we again used Routh's table to the closed loop polynomial $s^4 + b_1 s^3 + b_2 s^2 + b_3 s + b_4$. The conditions for stability are

$$b_1 b_2 > b_3, \quad b_3 L_1 > b_1 L_2, \quad b_3 = (1 + KK_p)/A_1$$

$$b_1 = A_2/A_1, \quad b_2 = (A_3 + KK_d)/A_1, \quad L_1 = b_2 - (b_3/b_1), \quad L_2 = b_4$$

A Pascal subroutine was developed that implements this stability analysis. Some actual experimental results obtained using the policy (14) are shown in fig. 15a–c). Figure 15a refers to $T_1 = 0.2$ s, $T_2 = 0.1$ s and $T_3 = 0.7$ s. The overshoot from 15% is reduced to 1% and the rise time from 0.175 s to 0.1 s ($K_1 = 0.5$, $K_2 = 0.2$, $K_3 = 0.4$). Figure 15b corresponds to $T_1 = 0.6$ s, $T_2 = 0.7$ s and $T_3 = 0.8$ s. The overshoot is reduced from 14% to 2% and the rise time from 0.6 s to 0.55 s ($K_1 = 0.2$, $K_2 = 0.27$, $K_3 = 0.9$). Finally, in fig. 15c the overshoot is reduced from 17% to nearly 0% and the rise time from 0.5 s to 0.3 s ($K_1 = 0.4$, $K_2 = 0.4$, $K_3 = 0.4$).

Similar improvements were also obtained for a system with two time constants and a time delay.

The incremental PID controller is of the "intelligent" type, since the changes to the parameter settings are made in a fashion similar to the human operator action. It is actually a controller of a higher hierarchical level than the classical analogue or digital PID controllers, like the fuzzy supervisor, and posseses in an accumulated form all the relative human experience. One of the main characteristics of this controller is its small computational requirement. The implementation on a digital PID controller needs 3 multiplications, 6 divisions, 7 additions and 1 subtraction. The total implementation time on a Z-80 processor is 959 μs (i.e. about 1 ms). Analogous computation time is required for the implementation of the method on an analogue PID controller. With a 16-bit

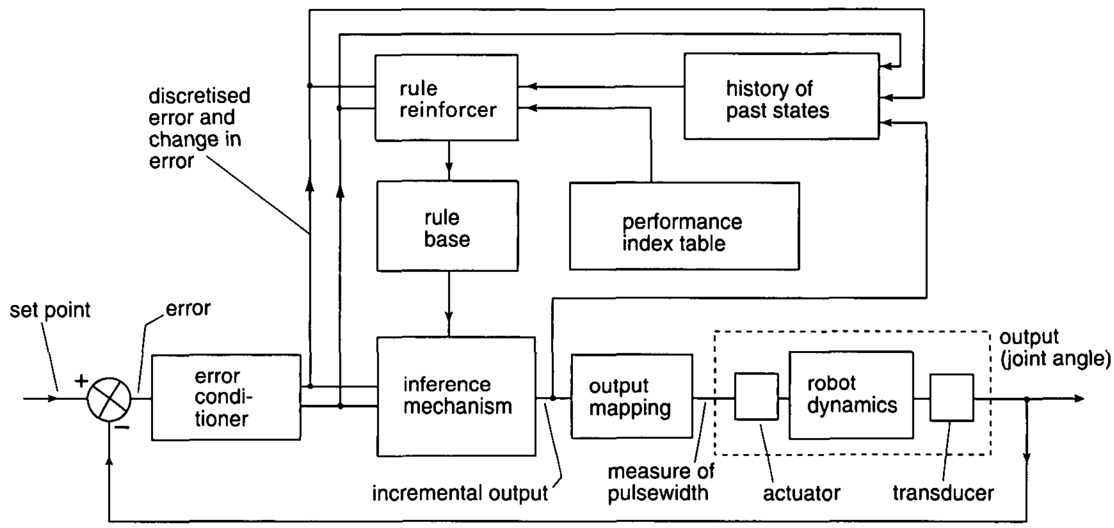


Figure 16 Fuzzy expert self-organized robot control system

Table 6 Maximum x, z deviations for square test[a]

Experiment	x-deviation (cm)	z-deviation (cm)
Continuous PID	1.97	1.97
Quantized PID	3.50	2.30
CG SOC	1.97	0.87

[a]For a vertical z-x square 55 cm × 55 cm with 9 s traverse time.

microprocessor, the computational times are reduced to 10% of the above figure. The substantial improvement of the performance achieved in combination with the above small computational time, makes the method ideal. The additional memory required for the *a priori* storage of the fuzzy matrix is very small. The 1 ms computational time is about 10% of the computational times needed for the SOC controller of Mamdani.

6.4 *Application of the self-organizing controller to a robot arm*

Mandič, Scharf and Mamdani (1985) have applied the self-organizing controller (SOC) technique to a 2-joint articulated robotic arm interfaced to a DEC LSI 11/23 processor. This *Smart Arms* 6R/1000 robot has a reach of 1 m, and can pick-up objects of weight up to 1.5 kg. The numerical figures of the robot are: shoulder length 0.57 m, arm joint length 0.43 m, arm motion range 90°, shoulder movement range 120° and gripper rotation range 300°. Each axis is driven by a DC motor.

The defuzzification of the error (E) and change-in-error (C E) was based either on the *centre-of-gravity* (CG) or on the *mean of maxima* (MM) method. Each robot joint is controlled by a separate single-input single-output SOC. By controlling the motion of each link (axis) we can ensure that the end effector follows a desired 3-dimensional path.

The structure of the complete control loop of each axis is shown in fig. 16.

The most advanced experiments carried out by Mamdani's group concern the simultaneous control of the arm and shoulder motion, which could be accommodated within the 20 ms cycle of the computer's real-time clock, and corresponds to a very large variation in moment of inertial applied to the motors. The tasks of the robot in these experiments were (i) the tracking of a square, (ii) the tracking of a sinusoid, (iii) position step response, and (iv) step response under load cycling.

The performance of the fuzzy SOC controller, compared to that of the "continuous" or "quantized" PID controller, achieved in the square tracking test with the CG method, is shown in Table 6.

In the sinusoidal tests the shoulder and arm motions were required to follow sinusoids with periods of 4.5 s and 3 s, respectively, with the *MM* defuzzification method. The results compared to those obtained by the continuous PID controller showed a deviation of the same order. However, the arm response for the 3 s sinusoid with the SOC was much better, and did not saturate to the extent shown by the continuous PID controller. This means that the SOC technique performs better in the case of fast trajectories.

The load cycling tests were carried out with both the *MM* and *CG* versions of the SOC. The results for SOC with *CG* are cleaner than those with the *MM* method. Details for these experiments can be found in Mandič, Scharf and Mamdani (1985).

7 Conclusions

In this paper, an overview of the principal results of fuzzy expert controllers and their applications to industrial and robotic systems was given. Fuzzy controllers are inherently more appropriate for systems which cannot be modelled accurately and involve uncertainties and ambiguities for which there is no available standard statistical data (probability distributions, moments, etc.). The vagueness and inexactness involved in the system is transformed into fuzzy set form, which is further represented in linguistic (rule-based) form. The conversion of the fuzzy information into deterministic decisions/actions is carried out through various defuzzifications techniques.

Fuzzy controllers are used in cases where control is left to the operator. Hence, the fuzzy controller, in addition to its obvious advantages, possesses some shortcomings, arising in this kind of control. Specifically, it reaches the set point more quickly and with decreased overshoot over standard control, but may have some oscillations around the set point or a steady-state error.

The dynamic properties of the fuzzy controller can be adjusted by a series of carefully designed experiments, at two conceptually distinct levels (Pedrycz, 1993):

- At a *numerical level*. The fuzzy controller is regarded as a nonlinear numerical transformation that, in conjunction with the system under control, gives rise to a certain phase portrait of the overall structure.
- At a *linguistic level*. At this lelvel, the linguistic phase plane is exploited and the resulting trajectories generated by the system-controller structure are mapped there. This implies that the overall analysis is performed at a completely different level.

The fuzzy expert controller can be implemented by using different operators to compute its fuzzy relation (Gupta, 1991; Mizumoto et al., 1979), but the max-min composition as a dominant construct is preserved (Mizumoto and Zimmermann, 1982).

Fuzzy rule-based controllers have found wide applicability in complex industrial plants, but there is still large room for further theoretical developments and practical improvements. Of particular interest should be to develop more types of adaptive-like fuzzy control, examples of which are the self-organizing controller, the fuzzy PID supervisor and the intelligent incremental controller discussed in the paper. Of theoretical interest is to study the performance characteristics of different fuzziness representations and fuzzy control matrices in a unified way. Another direction for further study is the field of fuzzy-learning controller design, i.e. the design of fuzzy controllers which automatically learn about the system under control as well as about their behaviour, and add new rules (or modify themselves) so as to face the learned situations. Still another direction is to accommodate in the fuzzy controller design more capabilities, such as a decoupling ability, robustness, closed-loop stability, a self-tuning/identification ability, etc. To accommodate such control features in a linguistic, humanlike, experiential form, one needs to employ artificial intelligence techniques and tools (Tzafestas, 1987, 1988, 1989).

Acknowledgement

The author expresses his thanks to the anonymous referees for their useful and constructive comments which improved the contents of the paper.

References

Adamo, JM, 1980a. "L.P.L. A fuzzy programming language: 1. Syntactic Aspects". *Fuzzy Sets and Systems* **3** 151–179.

Adamo, JM, 1980b. L.P.L. A fuzzy programming language: 2. Semantic Aspects". *Fuzzy Sets and Systems* **3** 261–289.

Aliev, RA, Oulanov, GM and Tserkovnyi, AE, 1986. "Decision making in intelligent robots". *Soviet Physics Doklady* **31**(10) 780–782.

Amano, A and Aritsuka, T, 1989. "On the use of neural networks and fuzzy logic in speech recognition". In: *Proc. Int. Joint Conf. on Neural Networks* Washington, DC, pp. 301–305.

Bardossy, A, 1990. "Note on fuzzy regression". *Fuzzy Sets and Systems* **37** 65–75.

Bartolini, G, Casolino, G, Daoli, F and Mortem, M, 1985. "Development of performance adaptive fuzzy controllers with application to continuous casting plans". In: *Industrial Applications of Fuzzy Control* (M. Sugeno, ed.). North-Holland, pp. 73–86.

Bellman, R and Giertz, M, 1973. "On the analytic formalism of the theory of fuzzy sets". *Information Science* **5** 149–156.

Berenji, HR, 1992. "Fuzzy and neural control". In: *An Introduction to Intelligent and Autonomous Control* (P. Antsaklis and K. Passino, eds.). Kluwer.

Bertsekas, DP, 1971. "Control of uncertain systems with a set-membership description of the uncertainty". MIT Electr. Syst. Lab. Rept ESL-R-447, Cambridge, MA.

Black, M, 1963. "Reasoning with loose concepts". *Dialogue* **2** 1–12.

Bounas, A, 1993. "Criteria for fuzzy reasoning through relations for conditional propositions". *The Journal of Fuzzy Mathematics* **1**(3) 587–591.

Bounas, A and King, RE, 1994. "Criteria for value approximations of fuzzy systems". Tech. Report, NRC "Demokritos", Aghia Paraskevi, Attiki, Greece.

Bounds, DG, Lloyd, PJ and Mathew, BG, 1990. "A comparison of neural network and other pattern recognition approaches to the diagnosis of low back disorders". *Neural Networks* **3** 583–591.

Braee, M and Rutherford, DA, 1979. "Theoretical and linguistic aspects of the fuzzy logic controller". *Automatica* **15** 553–577.

Brown, M, Fraser, R, Harris, CJ and Moore, CG, 1991. "Intelligent self-organizing control for autonomous guided vehicle: comparative aspects of fuzzy logic and neural nets". In: *'91 IEE International Conference on Control*, pp. 134–139.

Buckley, JJ and Siler, W, 1987. "Managing uncertainty in a fuzzy expert system. Part 1: Combining uncertainties".Preprints: *2nd IFSA Congress*, pp. 737–739.

Cassinis, R, Biorli, E, Meregalli, A and Scalise, F, 1988. "Behavioral model architecture: a new way of doing real-time planning in intelligent robots". *International Society for Optical Engineering* **852** 275–280.

Celmins, A, 1987. "Least squares model fitting to fuzzy linear data". *Fuzzy Sets and Systems* **22** 245–269.

Chang, SSL and Zadeh, LA, 1972. "On fuzzy mapping and control". *IEEE Trans. Sys. Man Cybern* **2** 30–34.

Cohen, ME and Hudson, DL, 1992. "Approaches to the handling of fuzzy input data in neural networks". In: *Proc. 1st IEEE Conf. on Fuzzy Systems* San Diego, CA. pp. 93–100.

Deepak, S, Repko, MC, Mood, NC and Kelley, RB, 1989. "A fuzzy approach to the interpretation of robot assembly forces". *International Society for Optical Engineering* **1002** 426–433.

De Silva, CW and McFarlane, AG, 1989. "Knowledge-based control approach for robotic manipulators". *International Journal of Control* **50**(1) 249–273.

Deyong, R, Polson, J, Moore, R, Weng, C and Lara, J, 1992. "Fuzzy and adaptive control simulation for a walking machine". *IEEE Control Systems Magazine* **12**(3) 43–50.

Di Nola, A, Pedrycz, W and Sessa, S, 1985. "On measures of fuzziness of solutions of fuzzy relation equations with generalized connectives". *J. Math. Anal. and Appl* **106** 443–453.

Diamond, P, 1990. "Higher-level fuzzy numbers arising from fuzzy regression models". *Fuzzy Sets and Systems* **36** 265–275.

Dodds, DR, 1988a. "Fractals, fuzzy sets and image representation". *International Society for Optical Engineering* **1001** 87–94.

Dodds, DR, 1988b. "Fuzziness in knowledge-based robotic systems". *Fuzzy Sets and Systems* **26**(2) 179–193.

Dote, Y, Suyitno, A and Strefezza, M, 1990. "Fuzzy learning grasping force controller for manipulator hand". In: *Proc. IECON '90: 16th Annual Conference of IEEE Industrial Electronics Society*, pp. 1259–1265.

Dubois, D and Pradé, H, 1978. "Operations on fuzzy numbers". *Int. J. Systems Sci* **9**(6) 613–626.

Fukami, S, Mizumoto, M and Tanaka, K, 1980. "Some considerations of fuzzy conditional inference". *Fuzzy Sets and Systems* **4** 243–273.

Fukuda, T, Fujiyoshi, T, Arai, F and Matsuura, H, 1991. "Design and dextrous control of micromanipulator with 6 dof". In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 1628–1633.

Gain, B, 1983. "Precise past–fuzzy future". *Int. J. Man-Machine Studies* **19** 117–134.

Garcia-Cerezo, A, Barreiro, A and Ollero, A, 1991. "Design of robust intelligent control of manipulators". In: *Proc. IEEE International Conference on Systems Engineering*, pp. 225–228.

Gasos, J, Garcia-Allegre, MC and Garcia Rosa, R, 1990. "Fuzzy local navigation of a simulated real robot in unknown environment". In: *Proc. IEEE International Workshop on Intelligent Motion Control*, pp. 445–449.

Gorez, R and De Neyer, M, 1994. "Fuzzy control of robotic manipulators and mechanical systems". In: *Fuzzy Reasoning in Information and Control Systems* (SG Tzafestas and AN Venetsanopoulos, eds.). Kluwer, pp. 451–491.

Gupta, MM, 1991. "Theory of T-norms and fuzzy inference methods". *Fuzzy Sets and Systems* **40** 431–450.

Hai Quan Dai, Dalton, GR and Tulenko, J, 1992. "Fuzzy control system for a mobile robot". *Trans. of American Nuclear Society* **65** 464–465.

Han-Gyoo, KA, 1991. "A new interpretation of force/servo control of robot arms". In: *Proc IROS '91 International Workshop on Intelligent Robots and Systems*, pp. 1623–1627.

Harris, CJ and Brown, M, 1991. "Intelligent control for autonomous guided vehicles". *Proc. IEEE Colloquium on Intelligent Control* **3** 1–4.

Higashi, M, Di Nola, A, Pedrycz, W and Sessa, S, 1984. "Ordering fuzzy sets by consensus concept and fuzzy relation equations". *Int. J. General Syst.* **10** 47–56.

Hirota, K, 1979. "Extended fuzzy expressions of probabilistic sets". In: *Advances in Fuzzy Set Theory and Applications* (MM Gupta, RK Ragade and RR Yager, eds.). North Holland, pp. 201–214.

Hirota, K, 1981. "Concepts of probabilistic sets". *Fuzzy Sets and Systems* **5** 31–46.

Hirota, K, Arai, Y and Hachisu, S, 1986. "Moving mark recognition and moving object manipulation in fuzzy controlled robot". *Control-Theory and Advanced Technology* **2** 399–418.

Hirota, K, Arai, Y and Hachisu, S, 1989. "Fuzzy control robot arm playing two-dimensional ping-pong game. *Fuzzy Sets and Systems* **32**(2) 149–159.

Holt, AW and Commoner, F, 1970. "Events and conditions". New York: *Applied Data Research*. (Also in *Rec. Project MAC Conf. Concurrent Systems and Parallel Computation* New York, ACM, pp. 1–72.)

Ischibuchi, H, Okada, H and Tanaka, H, 1992. "Interpolation of fuzzy If-Then rules by neural networks". In: *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 337–340.

Ishii, K and Misra, S, 1990. "Uncertainty management in intelligent task planning of mobile robots". *International Journal of Computer Integrated Manufacturing* 3(1) 13–23.

Ishikawa, S, 1991. "A method of indoor mobile robot navigation by using fuzzy control". In: *Proc. IROS '91 International Workshop on Intelligent Robots and Systems*, pp. 1013–1018.

Isik, C, 1988. "Inference engines for fuzzy rule-based control". *International Journal of Approximate Reasoning* **2**(2) 178–188.

Isik, C and Meystel, AM, 1988. "Pilot level of a hierarchical controller for an unmanned mobile robot". *IEEE Journal of Robotics and Automation* **4**(3) 241–255.

Jajuga, K, 1986. "Linear fuzzy regression". *Fuzzy Sets and Systems* **20** 343–353.

Janko, WH, Roubens, M and Zimmermann, H, 1990. "*Progress in Fuzzy Sets and Systems*". *Kluwer.*

Jensen, JH, 1979. "Fuzzy logic control of industrial process". PhD Thesis, Electr. Power Eng. Dept., Tech. Univ. of Denmark.

Joszef, S, 1992. "On the effect of linear data transformations in possibilistic fuzzy linear regression". *Fuzzy Sets and Systems* **45** 185–188.

Kalley, GS and McDermott, KJ, 1989. "Using expert systems to incorporate the principles of motion economy as a means of improving robot manipulator path generation". *Computers and Industrial Engineering* **16**(32) 207–213.

Kang, H and Vachtsevanos, G, 1990. "An intelligent strategy to robot coordination and control". In: *Proc. CDC '90: 29th IEEE Conference on Decision and Control* pp. 2208–2213.

Kawamura, H, Tani, A, Yakamoto, K and Yamada, M, 1990. "Constitution of intelligent fuzzy network by frame knowledge representation". In: *Proc. Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 261–265.

Kawamura, H, Tani, A, Kambara, H and Yamada, M, 1991. "Intelligent fuzzy network for optimum structural planning and design". In: *Proc. 7th Fuzzy Systems Symp.* Nagoya, Japan, pp. 99–102.

Kawamura, H, Tani, A and Kambara, H, 1992. "A seismic structural planning system for fuzzy network". In: *Proc. 10th World Conf. on Earthquake Engineering* Madrid, Spain, pp. 6271–6275.

Keller, JM, Yager, RR and Tahani, H, 1992. "Neural network implementation of fuzzy logic". *Fuzzy Sets and Systems* **45** 1–12.

Keller, JM and Tahani, H, 1992. "Implementation of conjunctive and disjunctive fuzzy logic rules with neural networks". *Int. J. Approximate Reasoning* **6** 221–240.

Kemal, C, 1989. "Fuzzy rule-based motion controller for an autonomous mobile robot". *Robotica* 7(1) 37–42.

Kickert, W and Van Nauta Lemke, HR, 1976. "The application of fuzzy set theory to controller in warm water plant". *Automatica* **12** 301–308.

Kickert, WJM and Mamdani, EH, 1978. "Analysis of a fuzzy logic controller". *Fuzzy Sets and Systems* **1** 29–44.

Kickert, WJM, 1978. "*Fuzzy Theories on Decision Making*". Martinus Nijhoff.

King, RE, 1986. "Expert systems in the cement industry". *Zement Kalk. Gips*, **39**(3).

King, PJ and Mamdani, EH, 1977. "The application of fuzzy control systems to industrial processes". In: *Proc 6th IFAC World Congress (Boston 1975). Automatica* **13** 235–242.

King, RE and Karonis, FC, 1986. "Rule-based systems in the process industry". In: *Proc. 25th IEEE CDC* Athens.

Klir, GJ and Folger, TA, 1988. *Fuzzy Sets, Uncertainty and Information*. Prentice Hall.

Koh, KC, Cho, AHS, Kim, SK and Jeong, IS, 1990. "Application of self-organizing fuzzy control to the joint control of a Puma 760 robot". In: *Proc. IROS '91 International Workshop on Intelligent Robots and Systems* pp. 537–542.

Kosko, B, 1991. *Neural Networks and Fuzzy Systems: A dynamical system approach to machine intelligence*. Prentice Hall.

Kubota, TK and Hashimoto, H, 1990. "A strategy for collision avoidance among moving obstacles for a mobile robot". In: *Proc. 11th IFAC World Congress on Automatic Control in the Service of Mankind*, pp. 105–110.

Kuncicky, DC and Kandel, A, 1989. "A fuzzy interpretation of neural networks". In: *Proc. 3rd Int. Fuzzy Systems Association Congress* Seattle, WA, pp. 113–116.

Kwok, DP, Tam, P, Li, CK and Wang, P, 1990. "Linguistic PID controllers". In: *Proc. 11th IFAC World Congress on Automatic Control in the Service of Mankind* Tallin, Estonia, pp. 192–197.

Lakov, D, 1985. "Adaptive robot under fuzzy control". *Fuzzy Sets and Systems* **17**(1) 1–8.

Larsen, PM, 1980. "Industrial applications of fuzzy logic control". *Int. J. Man-Machine Studies* **12** 3–10.

Lee, CC, 1990. "Fuzzy logic in control systems: fuzzy logic controller (Part I)". *IEEE Trans. Syst. Man Cybern.* **20**(2).

Leung, KS and Lam, W, 1988. "Fuzzy concepts in expert systems". *Computer Magazine IEEE* **21**(9) 43–58.

Li, Y, 1986. "The automatic recognition of mechanical parts on membership's principles of fuzzy sets". *International Society for Optical Engineering* **697** 175–180.

Li, YF and Lau, CC, 1989. "Development of fuzzy algorithms for servo systems". *IEEE Control Systems Magazine* **9**(3) 65–71.

Lim, CM and Hiyama, TI, 1991. "Application of fuzzy logic control to a manipulator". *IEEE Trans. on Robotics and Automation* **7**(5) 688–691.

Lin, CE and Sheu, YR, 1992. "A hybrid control-approach for pendulum-car control". *IEEE Trans. on Industrial Electronics* **39**(3) 208–214.

Looney, CG, 1994. "Fuzzy Petri nets and applications". In: *Fuzzy Reasoning in Information, Decision and Control Systems* (SG Tzafestas and AN Venetsanopoulos, eds.). Kluwer, pp. 511–527.

Looney, CG, 1988a. "Expert control design with fuzzy rule matrices". *Int. J. Expert Systems* **1**(2) 159–168.

Looney, CG, 1988b. "Fuzzy Petri nets for rule based decision making". *IEEE Trans. Syst. Man and Cybern.* **18**(1) 178–183.

MacVicar-Whelan, PJ, 1975. "Fuzzy sets, the concept of height and hedge". Tech. Memorandum 1, Physics Department, Grand Valley State Colleges, Allendale, MI.

MacVicar-Whelan, PJ, 1976. "Fuzzy sets for man-machine interaction". *Int. J. Man-Machine Studies* **8** 687–697.

Mamdani, EH and Assilian, S, 1974. "An experiment in linguistic synthesis with a fuzzy logic controller". *Int. J. Man-Machine Studies* **7** 1–13.

Mamdani, EH and Assilian, S, 1975. "A fuzzy logic controller for a dynamic plant". *Int. J. Man-Machine Studies* **7** 1–13.

Mamdani, EH, 1976.. "Advances in the linguistic synthesis of fuzzy controllers". *Int. J. Man-Machine Studies* **8** 669–678.

Mamdani, EH, 1977. "Application of fuzzy logic to approximate reasoning using linguistic synthesis". *IEEE Trans. Computers* **26** 1182–1191.

Mamdani, EH and Procyk, TJ, 1979. "A linguistic self-organizing process controller". *Automatica* **15** 15–30.

Mandič, NJ, Scharf, EM and Mamdani, EH, 1985. "Practical application of a heuristic fuzzy rule-based controller to the dynamic control of a robot arm". *Proc. IEEE* **132**(D) 190–203.

Martin, TP, Baldwin, JF and Pilsworth, BW, 1987. "The implementation of FPROLOG–A fuzzy prolog interpreter". *Fuzzy Sets and Systems* **23** 119–129.

Masuoka, R, Watanabe, N, Kawamura, A, Owada, Y and Asakawa, A, 1990. "Neurofuzzy system-fuzzy inference using structured neural network". In: *Proc. 1990 Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 173–177.

Miyakoshi, M and Shimbo, M, 1985. "Solutions of composite fuzzy relational equations with triangular norms". *Fuzzy Sets and Systems* **16** 53–63.

Miyakoshi, M and Shimbo, M, 1986. "Lower solutions of systems of fuzzy equations". *Fuzzy Sets and Systems* **19** 37–46.

Miyamoto, S, 1990. *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Kluwer.

Mizumoto, M and Tanaka, K, 1976. "Some properties of fuzzy sets of type 2". *Information and Control* **31** 312–340.

Mizumoto, M, Fukami, S and Tanaka, K, 1979. "Some Methods of fuzzy reasoning". In: *Advances in Fuzzy Set Theory and Applications* (MM Gupta, RK Ragade and RR Yager, eds.) North-Holland, pp. 117–126.

Mizumoto, M and Zimmermann, H, 1982. "Comparison of fuzzy reasoning methods". *Fuzzy Sets and Systems* **8** 253–283.

Mizumoto, M, 1985. "Extended fuzzy reasoning". In: *Approximate Reasoning in Expert Systems* (M Gupta et al., eds.). North-Holland, pp. 71–85.

Mizumoto, M, 1988. "Fuzzy controls under various fuzzy reasoning methods". *Information Sciences* **45** 129–151.

Muarakami, S, Takemoto, F, Fujimura, H and Ide, E, 1989. "Weld-line tracking control of arc welding robot using fuzzy logic controlled". *Fuzzy Sets and Systems* **32**(4) 221–237.

Murata, T, Subrahmanian, VS and Wakayama, T, 1991. "A Petri net model for reasoning in the presence of inconsistency". *IEEE Trans. Knowledge and Data Engineering* **3**(3) 281–292.

Nakanishi, S, Tagaki, T, Uehara, K and Gotoh, Y, 1990. "Self-organizing fuzzy controllers by neural networks". In: *Proc. Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp.– 187–192.

Nakanishi, S and Takagi, T, 1990. "Pattern recognition by neural networks and fuzzy inference". In: *Proc. 1990 Int. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan, pp. 183–186.

Nedungadi, A and Wenzel, DJ, 1991. "A novel approach to robot control using fuzzy logic". In: *Proc. '91 IEEE International Conference on Systems, Man and Cybernetics* pp. 1925–1930.

Negoita, CV and Ralescu. D, 1975. *Application of Fuzzy Sets to Systems Analysis*. Birkhauser Verlag.

Negoita, CV and Ralescu, DA, 1987. *Simulation, Knowledge-based Computing and Fuzzy Statistics*. Van Nostrand Reinhold.

Okada, H., Watanabe, N, Kawamura, A, Asakawa, A, Taira, T, Ishida, K, Kaji, T and Narita, N, 1992.. "Knowledge implementation multilayer neural networks with fuzzy logic". In: *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks*, Iizuka, Japan, pp. 99–102.

Ostergaard, JJ, 1976. "Fuzzy logic control of a heat exchange process". Report No. 7601, Electr. Power Dept. Tech. Univ. of Denmark.

Palm, R, 1989. "Fuzzy controller for a sensor guided robot manipulator". *Fuzzy Sets and Systems* **31**(2) 133–149.

Pao, YH, 1989. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley.

Pappis, CP and Sugeno, M, 1976. "Fuzzy relational equations and the inverse problem". Internal Report. Queen Mary College, London.

Pappis, CP and Sugeno, M, 1985. "Fuzzy relational equations and the inverse problem". *Fuzzy Sets and System* **15** 79–90.

Perdrycz, W, 1980. "On the use of Lukasiewicz logic for fuzzy control". *Arch. Autom.* **3** 301–313.

Perdrycz, W, 1993. *Fuzzy Control and Fuzzy Systems*. Wiley.

Peterson, JL, 1981. *Petri Net Theory and the Modelling of Systems*. Prentice-Hall.

Petri, CA, 1962. "Kommunikation mit automaten". Schriften des Rheinish-Westfalischen Institute für Instrumentelle Mathematik an der Universität Bonn Heft 2, Bonn, Germany.

Reising, W, 1985. *Petri Nets*. Springer-Verlag.

Sakawa, M, 1983. "Interactive computer programs for fuzzy linear programming with multiple objectives". *Int. J. Man-Machine Studies* **18** 489–503.

Sakawa, M and Yano, H, 1986. "An interactive fuzzy decision making method using constraint problems". *IEEE Trans. Systems Man Cybern.* **16** 179–182.

Scharf, EM and Mandič, NJ, 1985. "The application of a fuzzy controller to the control of a multi-degree of freedom robot arm". In: *Industrial Applications of Fuzzy Control* (M Sugeno, ed.) North-Holland.

Scharf, E, 1985. "Fuzzy logic could redefine robot control". *Automation* **21**(2) 11–14.

Schweppe. FC, 1968. "Recursive state estimation: Unknown but bounded errors and systems inputs". *IEEE Trans. Auto. Contr.* **13**(1) 22–28.

Seung-Woo, K and Mignon, P, 1991. "Fuzzy compliance robot control". In: *Proc. IROS '91 International Workshop on Intelligent Robots and Systems* pp. 1628–1631.

Siler, W, 1987. "FLOPS: A fuzzy expert system shell". Preprints: *2nd IFSA Congress* pp. 848–850.

Siler, W and Buckley, JJ, 1987. "Managing uncertainty in a fuzzy expert system. Part 2: Truth maintenance system". Preprints: *2nd IFSA Congress* pp. 744–746.

Silverman, RH and Noetzel, A, 1990. "Image processing and pattern recognition in ultrasonograms by backpropagation". *Neural Networks* **3** 593–603.

Sira-Ramirez, H, 1980. "Fuzzy state estimation in linear dynamic systems". In: *Proc. IEEE Conf. on Decision and Control* **2** 380–382.

Sira-Ramirez, H, 1979. "Evolution of fuzzy sets in linear dynamic systems". In: *Proc. 1979 Int. Conf. on Cybernetics and Society* Denver, CO.

Siy, P and Chen, CS, 1974. "Fuzzy logic for handwritten numerical character recognition". *IEEE Trans. Syst., Man Cybern.* **4** 570–575.

Sosnowski, ZA, 1990. "FLISP–A language for processing fuzzy data". *Funny Sets, and Systems* **37** 23–32.

Sosnowski, ZA, 1991. "Data structures for representing and processing of fuzzy information in Lisp. *Computers and Artificial Intelligence* **10**(6) 561–571.

Stellakis, KM and Valavanis, KP, 1991. "Fuzzy logic-based formulation of the organizer of intelligent robotic systems". *Journal of Intelligent and Robotics Systems: Theory and Applications* **4**(1) 1–24.

Stipanicev, D, De Neyer, M and Gorez, R, 1991. "Self-tuning self-organizing fuzzy robot control". In: *Robot Control* (Troch I, Desoyer, K and Kopacek, P, eds.). Pergamon Press, pp. 171–176.

Sugeno, M, 1985. *Industrial Applications of Fuzzy Control.* North-Holland.

Sugeno, M and Nishida, M, 1985. "Fuzzy control of model car". *Fuzzy Sets and Systems* **16** 103–113.

Sugeno, M and Murakami, K, 1985. "An experimental study on fuzzy parking control using a model car". In: *Industrial Applications of Fuzzy Control* (M Sugeno, ed.). North-Holland, pp. 125–138.

Suh, Il, Hong, Jong Hyuck Hong, Sang-Rok Oh and Kwang Bae Kim, 1991. "Fuzzy rule-based position/force control of industrial manipulators". In: *Proc. IROS '91 International Workshop on Intelligent Robots and Systems*, Osaka, Japan, pp. 111–116.

Takagi, T and Sugeno, M, 1985. "Fuzzy identification of systems and its applications to modelling and control". *IEEE Trans. Syst., Man Cybern.* **15** 116–132.

Takagi, H, 1990. "Fusion technology of fuzzy theory and neural network-survey and future directions". In: *Proc. 1990 Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 13–26.

Takagi, H and Hayashi, I, 1991. "Artificial neural networks driven fuzzy reasoning". *Int. J. Approximate Reasoning* **5** 191–212.

Takagi, T and Sugeno, M, 1993. "Derivation of fuzzy control rules from human operator's control actions". In: *Proc. IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis* Marseilles, France, pp. 55–60.

Takahashi, H and Minami, H, 1989. "Subjective evaluation modelling using fuzzy logic and a neural network". *Proc. 3rd Int. Congress on Fuzzy Systems Association* Seattle, WA, pp. 520–523.

Takeuchi, T, Nagai, Y and Enomoto, N, 1988. "Fuzzy control of a mobile robot for obstacle avoidance". *Information Sciences* **45**(2) 231–248.

Takeuchi, T, 1991. "An autonomous fuzzy mobile robot". *Advanced Robotics* **5**(2) 215–230.

Tanaka, H, Uejima S and Asai, K, 1982. "Linear regression analysis with fuzzy model". *IEEE Trans. Syst. Man Cybern.* (**12**)(6).

Tanaka, H and Watanabe, J, 1988. "Possibilistic linear systems and their applications to the linear regression model". *Fuzzy Sets and Systems* **27** 275–289.

Tanaka, H and Ishibuchi, H, 1991. "Identification of possibilistic linear systems by quadratic membership functions of fuzzy parameters". *Fuzzy Sets and Systems* **41** 145–160.

Tanscheit, R and Scharf, RJ, 1988. "Experiments with the use of a rule-based self-organizing controller for robotics applications". *Fuzzy Sets and Systems* **26** 195–214.

Teichrow, J, Horskotte, E and Togai, M, 1989. "The fuzzy-C compiler: A software tool for producing portable fuzzy expert systems". In: *Proc. 3rd IFSA Congress, Int. Fuzzy Systems, Association*, pp. 708–711.

Thomason, MG, 1977. "Convergence of powers of a fuzzy matrix". *J. Math Anal. Appl.* **57** 476–480.

Tong, RM, 1976. "Analysis of fuzzy control algorithms using the relational matrix". *Int. J. Man-Machine Studies* **8** 679–686.

Tong, RM, 1977a. "A control engineering review of fuzzy systems". *Automatica* **13** 559–569.

Tong, RM, 1977b. "Analysis and control of fuzzy systems using finite discrete relations". *Int. J. Control* **32**.

Tong, RM, Beck, MB and Latten, A, 1980. "Fuzzy control of the activated sludge wastewater treatment process". *Automatica* **16** 695–701.

Tsukamoto, Y, 1979. "Fuzzy logic based on Lukasiewicz logic and its application to diagnosis and control". PhD Thesis, Tokyo Institute of Technology.

Tzafestas, SG, 1986. "Knowledge engineering approach to system modelling, diagnosis, supervision and control". In: *Proc. IFAC/IMACS Symp. on Simul. of Control Systems* Vienna, Austria. (Also, 1989, *Systems Anal. Modelling Simul.* **6** 1–19.)

Tzafestas, SG, 1987. "Artificial intelligence techniques in control". In: *AI, Expert Systems and Languages in Modelling and Simulation* (Kulikowski, C and Ferrate, G, eds.). North-Holland, pp. 55–67.

Tzafestas, SG (ed.), 1988. *Knowledge-based System Diagnosis, Supervision and Control.* Plenum.

Tzafestas, SG and Papanikolopoulos, N, 1988. "Intelligent PID control based on fuzzy logic". In: *Proc. IFAC Symp. on Distributed Intelligence Systems Methods and Applications (DIS '88)* Varna, Bulgaria.

Tzafestas, SG, 1990a. "Petri-net and knowledge-based methodologies in Manufacturing systems modelling, simulation and control". *Systems & Control Encyclop.*: Suppl. 1. Pergamon Press.

Tzafestas, SG, 1990b. "AI techniques in computer-aided manufacturing systems". In: *Knowledge Engineering* (H Adeli, ed.). McGraw-Hill, pp. 212.

Tzafestas, SG and Papanikolopoulos, N, 1990. "Incremental fuzzy expert PID control". *IEEE Trans. Industrial Electron.* **37**(5) 365–371.

Tzafestas, SG, 1991. "Adaptive, robust and fuzzy rule-based control of robotic manipulators". In: *Intelligent Robotic Systems* (Tzafestas, SG, ed.). Marcel Dekker, pp. 313–419.

Tzafestas, SG, Stamou, GB and Watanabe, K, 1994. "Fuzzy reasoning through a general neural network model". In: *Fuzzy Reasoning in Information, Decision and Control Systems* (SG Tzafestas and AN Venetsanopoulos, eds.). Kluwer, pp. 145–161.

Tzafestas, SG, Hatzivasiliou, FV and Kaltsounis, SK, 1994. "Fuzzy logic design of a nondestructive robotic fruit collector". In: *Fuzzy Reasoning in Information, Decision and Control Systems* (SG Tzafestas and AN Venetsanopoulos, eds.). Kluwer, pp. 553–561.

Tzafestas, SG, Terzakis, S and Venetsanopoulos, AN, 1994. "Fuzzy parameter and state estimation". In: *Fuzzy Reasoning in Information, Decision and Control Systems* (SG Tzafestas and Venetsanopoulos, eds.). Kluwer, pp. 347–368.

Tzafestas, SG, Palios, L and Cholin, F, 1994. "Diagnostic expert system inference engine based on the certainty factors model". *Knowledge-Based Systems* **7**(1) 17–26.

Tzafestas, SG and Venetsanopoulos, AN, 1994. *Fuzzy Reasoning in Information Decision and Control Systems*. Kluwer.

Tzafestas, SG and Stamou, G, 1994. "A fuzzy path planning algorithm for autonomous robots in an uncertain environment". In: *Proc. EURISCON '94: The 2nd European Robotics and Intelligent Systems Conference* Malaga, Spain.

Uehara, K and Fujise, M, 1990. "Learning of fuzzy-inference criteria with artificial neural network". In: *Proc. Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 193–198.

Umano, M, Mizumoto, M and Tanaka, K, 1978. "FSTDS system: A fuzzy-set manipulation system". *Information Sciences* **15** 115–159.

Umano, M, 1987a. "Fuzzy-Set Prolog". Preprints: *2nd IFSA Congress*, pp. 750–753.

Umano, M, 1987b. "Fuzzy-set manipulation system in Lisp". Preprints: *2nd IFSA Congress*, pp. 840–843.

Umbers, IG and King, PJ, 1980. "An analysis of human decision-making in cement kiln control and the implications for automation". *Int. J. Man-Machine Studies* **12** 11–23.

Uragami, M, Mizumoto, M and Tanaka, K, 1976. "Fuzzy robot controls". *Journal of Cybernetics* **6**(1–2) 39–64.

Vachtsevanos, GJ, Davey, K and Lee, KM, 1987. "Development of a novel intelligent robotic manipulator". *IEEE Control Systems Magazine* **7**(3) 9–15.

Valavanis, KP and Stellakis, KM, 1991. "A general organizer model for robotic assemblies and intelligent robotic systems". *IEEE Trans. on Systems, Man and Cybernetics* **21**(2) 302–317.

Van Nauta Lemke, HR and Dezhao, W, 1985. "Fuzzy PID supervisor". In: *Proc. 24th IEEE CDC* Paper WP9-5:15, pp. 602–608.

Van Amerongen, J, Van Nauta Lemke, HR and Van der Veen, JCT. 1977. "An autopilot for ships designed with fuzzy sets". In: *Proc. 5th IFAC/IFIP Int. Conf. on Digital Comp. Appl. to Process Control.*

Wakileh, BAN and Gille, KS, 1988. "Use of fuzzy logic in robotics". *Computers in Industry* **10**(1) 35–46.

Watanabe, K, Jin, S and Tzafestas, SG, 1994. "Fuzzy control for robot manipulators with artificial rubber muscles". In: *Fuzzy Reasoning in Information, Decision and Control Systems*. Kluwer, pp. 493–510.

Wenstop, F, 1976. "Deductive verbal models of organizations". *Int. J. Man-Machine Studies* **8** 293–311.

Witsenhausen, H, 1968. "Set of possible states for linear systems given perturbed observations". *IEEE Trans. Auto. Contr.* **13**(5) 556—558.

Xu, Y, Paul, RP and Shun, HY, 1991. "Fuzzy control of robot and compliance wrist systems". In: *Proc. IEEE Industry Applications Society Annual Meeting* pp. 1431–1437.

Yager, RR, 1992. "Implementing fuzzy logic controllers using a neural network framework". *Fuzzy Sets and Systems* **48** 53–64.

Yamakawa, T, 1990. "Pattern recognition hardware system employing a fuzzy neuron". In: *Proc. Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 943–948.

Yamamoto, S, 1994. "Software representation of fuzzy sets and logic". In: *Fuzzy Reasoning in Information, Decision and Control Systems* (SG Tzafestas and AN Venetsanopoulos, eds.). Kluwer, pp. 52–68.

Yasunobu, S and Hasegawa, T, 1986. "Evaluation of an automatic container crane operation system based on predictive fuzzy control". *Control-Theory and Advanced Technology* **2** 419–432.

Yen, J, 1990. The Role of Fuzzy Logic in the Control of Neural Networks. In: *Proc. Int. Conf. on Fuzzy Logic and Neural Networks.* Iizuka, Japan, pp. 771–774.

Zadeh, LA, 1965. "Fuzzy Sets". *Inform. and Control* **8** 338–353.

Zadeh, LA, 1969. "Fuzzy algorithms". *Inform. and Control* **11** 323–339.

Zadeh, LA, 1970. "Theory of approximate reasoning". In: *Machine Intelligence 9* (JE Hayes, D Michie and LI Mikulich, eds.). Ellis Horwood, pp. 149–194.

Zadeh, LA, 1973. "Outline of a new approach to the analysis of complex systems and decision processes". *IEEE Trans. Syst, Man. Cybern*, **3** 28–44.

Zadeh, LA, 1975. "Fuzzy logic and approximate reasoning. (In Memory of Grigore Moisil)". *Synthèse* **30** 407–428.

Zadeh, LA, 1978. "PRUF-A meaning representation language for natural languages". *Int. J. Man-Machine Studies* **10** 395–460.

Zadeh, LA, 1983a. "A computational approach to fuzzy quantifiers in natural languages". *Comp. & Maths.* **9** 149–184.

Zadeh, LA, 1983b. "Commonsense knowledge representation based on fuzzy logic". *Computer* October, 63–65.

Zahzah, EH, Desachy, J and Zehana, M, 1992. "A fuzzy connectionist approach for a knowledge based image interpretation system". In: *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 1135–1138.

Zhou, MC and Leu, MC, 1991. "Petri net modeling of a flexible assembly station for printed circuit boards". In: *Proc. 1991 IEEE Int. Conf. on Robotics and Automation* Sacramento, pp. 2530–2535.

Zhuang, WP, Qiao, WZ and Heng, TH, 1990. "The truth-valued flow inference network". In: *Proc. 1990 Int Conf. on Fuzzy Logic and Neural Networks* Iizuka, Japan, pp. 267–281.

Zimmermann, HJ, 1992. "Approximate reasoning in manufacturing". In: *Intelligent Design and Manufacturing* (A Kusiak, ed.). Wiley, pp. 701–722.