D. S. HOODA • VIVEK RAICH

Fuzzy Logic Models and Fuzzy Control

An Introduction

Alpha Science

Fuzzy Logic Models and Fuzzy Control

An Introduction

D. S. Hooda
Vivek Raich

Alpha Science International Ltd.
Oxford, U.K.

Fuzzy Logic Models and Fuzzy Control
An Introduction

408 pgs.

D. S. Hooda
Former PVC, Kurukshetra Univesity
Honorary Professor (Mathematics)
GJ University of Science & Technology Hisar
and adviser (R) to ABV university Bhopal

Vivek Raich
Department of Mathematics
Govt. Holkar Science College
Indore

ALPHA SCIENCE INTERNATIONAL LTD.

7200 The Quorum, Oxford Business Park North
Garsington Road, Oxford OX4 2JZ, U.K.

www.alphasci.com

ISBN 978-1-78332-281-7
E-ISBN 978-1-78332-326-5

Printed from the camera-ready copy provided by the Author.

Preface

Fuzzy control has increased tremendous interest in applications over the past few years and also among control equipment. The present book titled "Fuzzy Logic Models and Fuzzy Control: An Introduction" has been written to meet these inspirations. It consists of total nine chapters: First three chapters are related to fuzzy set theory, fuzzy logic, fuzzy systems and models, while next six chapters deal with design and analysis methodology of fuzzy control. However, it should be clear that such a universal theory does not exist for conventional control engineering either, so we have to proceed from a few isolated spots where we already know exactly how to design a fuzzy control algorithm to clusters of problems and related design methodologies.

Thus, we have tried to structure this book to make it a text book for control engineering covering just the relevant part of the theory and focussing on the principles of fuzzy control rather than particular applications or tools. Starting with relevant mathematics including fuzzy set theory, fuzzy logic and models, we go to design parameters and choices and discuss fuzzy control in control engineering terms such as linear, non-linear, adaptive control and stability criteria.

The book explicitly aimed at readers from B. Tech./ M.Tech engineering students who want to study fuzzy control as elective course. It contains the following nine chapters: Chapter 1 Fuzzy Set and Fuzzy Numbers; Chapter 2 Fuzzy logic and Fuzzy Systems; Chapter 3 Fuzzy System Models; Chapter 4 Fuzzy Control; Chapter 5 Fuzzy Knowledge Base Control (KFBC); Chapter 6 Adaptive Fuzzy Control; Chapter 7 Non-linear Fuzzy Control; Chapter-8 Stability of Fuzzy Control Systems, and Chaper 9 Fuzzy Control System Model.

D. S. Hooda
Vivek Raich

Contents

Chapter-1

Fuzzy Sets and Fuzzy Numbers

1.1 BASIC CONCEPTS AND DEFINITIONS

Only in 20th century, mathematicians defined the concepts of sets and functions to represent problems. In many circumstances the solution using this concept are meaningless. This difficulty was overcome by the fuzzy concept. Almost all mathematical, engineering, medicine etc. concepts have been redefined using fuzzy set. It essentially prepares the teacher as well as student in the use of fuzzy sets. Basic definition of crisp sets properties of α-cuts, representation of fuzzy sets, operations on fuzzy sets. These help in providing a more general appreciation of what has been done with fuzzy sets and applied in problems related to various areas. It also helps in providing an integrated view of extension principle for fuzzy sets. In this chapter we are going to introduce the Basic Definition of Fuzzy set, types of Fuzzy sets and Notation, Operations on Fuzzy sets, α-Cuts and Properties, Interval valued Fuzzy Set and Arithmetic of Fuzzy numbers for more details refer to [Klier and Yuan (1995) and Lee (2005)].

In real world, the complexity generally arises from uncertainty in the form of ambiguity. The probability theory has been an old age and effective tool to handle uncertainty, but it can be applied only to situations whose characteristics are based on random processes, i.e., processes in which the occurrence of events is strictly determined by chance. Uncertainty may arise due to partial information about the problem, or due to information which is not fully reliable, or due to inherent imprecision in the language with which the problem is defined or due to receipt of information from more than one source. Fuzzy set theory is an excellent mathematical tool to handle the uncertainty arising due to vagueness. In 1965, Lotfi A. Zadeh propounded the fuzzy set theory in his paper.

A set A can be equivalently represented by its characteristic function i.e. A mapping χ_A from the universe of discourse (i.e. region of consideration is a large set) containing A to the set $\{0,1\}$; equivalently, $x \in A$ iff $\chi_A(x)$ = 1.

In "fuzzy" case "belonging to" relation $\chi_A(x)$ between x and A is no longer "0 or 1" But it has a degree of "belonging to" i.e. membership degree such as 0.6. Therefore, the range has to extend from set {0, 1} to interval [0, 1].

1.1.1 Crisp Set

The study of sets and their use in the foundation of mathematics was just begun before the turn of the century by Cantor (1845-1918), Russel, Frege and others. The theory of sets developed by Cantor has not only influenced and enriched almost every branch of mathematics but also has helped to clarify the relation between mathematics and philosophy. The basic idea of a set is the origin point in the study of modern algebra.

Well-defined and distinct collection of objects is called Set.

The words aggregate, class or collection are also used in place of the word "set". But the use of the word set is common. In general capital letters like A, B, C, D…….etc are used to denote the sets and lower letters a, b, c, d………to denote the objects or elements belonging to these sets. We express the relation between an object and a set to which it belongs by writing, $a \in A$. There are three basic methods by which sets can be represented.

(i) List method- $A = \{a_1, a_2, a_3, ..., a_n\}$.

(**ii) Set builder form (Rule method**)-

A = {x | P(x)}, where P(x) means "x has the property P"

(iii) A **characteristic function** $\chi_A(x)$ that declares which elements of X are members of the set and which are not and that is represented as follows:

$$\chi_A(x) = \begin{cases} 1, & \text{when } x \in A \\ 0, & \text{when } x \notin A, \end{cases}$$

which is formally expressed by $\chi_A(x) : X \to [0,1]$.

1.1.2 Operations on Crisp Set.

Union – If A and B are two non-empty sets then $A \cup B$ defined as

$A \cup B = \{x \in X \mid x \in A \text{ or } x \in B\}$.

For a family of sets $\{A_i \mid i \in N\}$, this is defined as

$\cup Ai = \{x \mid x \in A_i \text{ for some } i \in N\}$.

Intersection - If A and B are two non-empty sets then $A \cap B$ defined as

$A \cap B = \{x \in X \mid x \in A \text{ and } x \in B\}$.

For a family of sets $\{A_i \mid i \in N\}$ is defined as

$\cap Ai = \{x \mid x \in A_i \text{ for all } i \in N\}$.

Complement of A- It is denoted by A^c and defined as

$A^c = \{x \in X \mid x \notin A\}$.

(i) $(A^c)^c = A$ (ii) $\phi^c = X$ (iii) $X^c = \phi$

1.1.3 Properties of operations on Crisp Set

Basic properties of union, intersection and complements are summarizing in the table-1.1, if A, B, C $\in$ P(X)

Table-1.1 Properties of operations on Crisp set

1.	Involution	$(A^c)^c = A$
2.	Commutative property	$A \cup B = B \cup A$; $A \cap B = B \cap A$
3.	Associative, property	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
4.	Distributive, property	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
5.	Idempotent, property	$A \cup A = A$, $A \cap A = A$
6.	Absorption, property	$A \cup (A \cap B) = A$, $A \cap (A \cup B) = A$
7.	Absorption by, X and ϕ	$A \cup X = X$, $A \cap \phi = \phi$
8.	Identity	$A \cup \phi = A$, $A \cap x = A$
9.	Law of contradiction	$A \cap A^c = \phi$
10.	Law of excluded	$A \cup A^c = X$
11.	De-Morgan's law	$(A \cap B)^c = A^c \cup B^c$, $(A \cup B)^c = A^c \cap B^c$

1.1.4 Convex Set

Definition. The term convex is applicable to a set A in $\Re^n$ (n – dimensional Euclidian vector space) if the followings are satisfied.

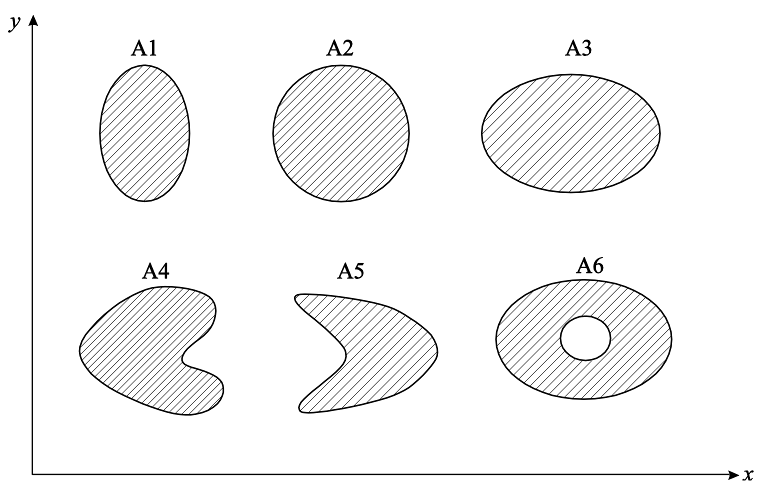


Figure 1.1: Convex sets A_1, A_2, A_3 and non-convex sets A_4, A_5, A_6 in $\Re^2$

(i) Two arbitrary points s and r are defined in A $r=\{r_i : i \in N_n\}$ $s=\{s_i : i \in N_n\}$. (N is a set of positive integers)

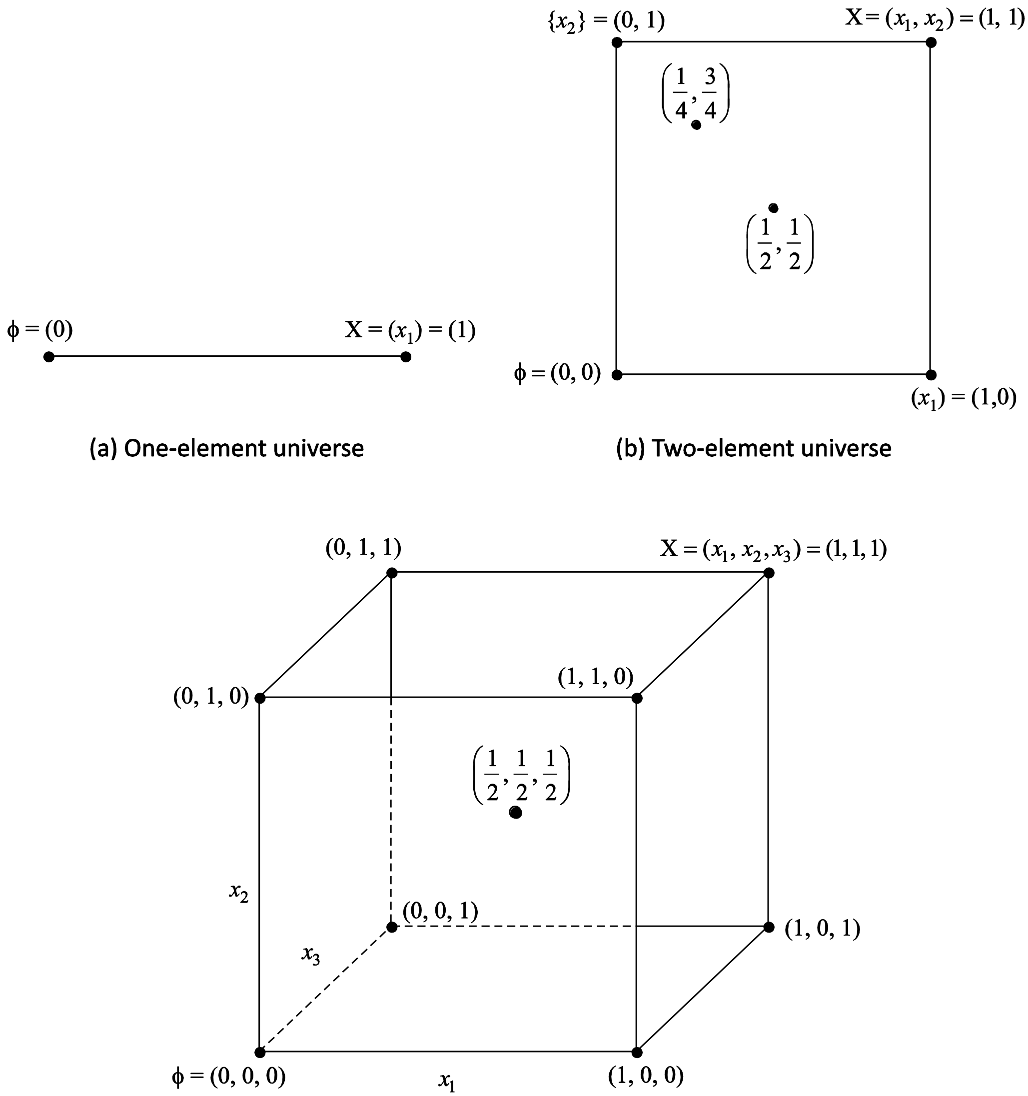
(ii) For arbitrary real number λ between 0 and 1, point t is involved in A, where t is $t=\{\lambda r_i+(1-\lambda)s_i : i \in N_n\}$.

In other words, if every point on the line connecting two point's *s* and *r* in *A* is also in *A*. (Fig.1.1) shows some examples of convex and non-convex sets.

1.1.5 Sets as Points in Hypercubes

There is an interesting geometric analog for illustrating the idea of set membership (Kosko, 1992). Heretofore we have described a fuzzy set A defined on a universe X. For a universe with only one element, the membership function is defined on the unit interval [0, 1]; for a two-element universe, the membership function is defined on the unit square; and for a three-element universe, the membership function is defined on the unit cube. All of these situations are shown in Fig. 1.2. For a universe of *n* elements we define the membership on the unit hypercube, $I^n=[0,1]^n$.

The endpoints on the unit interval in Fig. 1.2(*a*), and the vertices of the unit square and the unit cube in Figs. 1.2(b) and 1.2(c), respectively, represent the possible crisp subsets, or collections, of the elements of the universe in each figure. This collection of possible crisp (non-fuzzy) subsets of elements in a universe constitutes the power set of the universe. For example, in Fig. 1.2(c) the universe comprises three elements, $X=\{x_1,x_2,x_3\}$. The point (0, 0, 1) represents the crisp subset in 3-space, where x_1 and x_2 have no membership and element x_3 has full membership, i.e., the subset { x_2 }; the point (1, 1, 0) is the crisp subset where x_1 and x_2 have full membership and element x_3 has no membership, i.e., the subset { x_1, x_2 }; and so on for the other six vertices in Fig. 1.2(c). In general, there are 2*n* subsets in the power set of a universe with *n* elements; geometrically, this universe is represented by a hypercube in *n*-space, where 2*n* vertices represent the collection of sets constituting the power set. Two points in the diagrams bear special note, as illustrated in Fig. 1.2(c). In this figure the point (1, 1, 1), where all elements in the universe have full membership, is called the whole set, X, and the point (0, 0, 0), where all elements in the universe have no membership, is called the null set, ϕ.



(a) One-element universe (b) Two-element universe

(c) Three-element universe

Figure 1.2: Sets as points

The centroids of each of the diagrams in Fig. 1.2 represent single points where the membership value for each element in the universe equals. For example, the point in Fig. 1.2(b) is in the midpoint of the square. This midpoint in each of the three figures is a special point – it is the set of maximum ''fuzziness.'' A membership value of indicates that the element belongs to the fuzzy set as much as it does not – that is, it holds equal membership in both the fuzzy set and its complement. In a geometric sense, this point is the location in the space that is farthest from any of the vertices and yet equidistant from all of them. In fact, all points interior to the vertices of the spaces represented in below Figure represent fuzzy sets, where the membership value of each variable is a number between 0 and 1. For example, in

Fig. 1.2(b), the point represents a fuzzy set where variable x_1 has a 0.25 degree of membership in the set and variable x_2 has a 0.75 degree of membership in the set. It is obvious by inspection of the diagrams in Figure 1.2 that although the number of subsets in a power set is enumerated by $2n$ vertices, yet the number of fuzzy sets on the universe is infinite as represented by the infinite number of points on the interior of each space.

The concept of a fuzzy set is an extension of the concept of a crisp set. Just as a crisp set on a universal set X is defined by its characteristic function while X to {0, 1}, while a fuzzy set on a domain X is defined by its membership function from X to [0, 1].

Definition1. Let X be a non-empty set (called the universal set or the universe of discourse or simply domain). A fuzzy set A is a subset of X and is defined as

$$A=\{(x_i,\mu_A(x_i)):\mu_A(x_i)\in[0,1];\forall x_i\in X\},$$

where $\mu_A(x_i)$ represents the degree of membership and is defined as

$$\mu_A(x_i)=\begin{cases} 0, if\ x_i \in A\ and\ there\ is\ no\ ambiguity \\ 1\ \ if\ x_i \in A\ and\ there\ is\ no\ ambiguity \\ 0.5, there\ is\ maximum\ ambiguity\ whether\ \ x_i \in A\ or\ \ x_i \in A \end{cases}$$

Important Remark: Generally to distinguish Crisp set and Fuzzy set A is used in place of A. But for the simplicity of typing throughout the text we shall use A in place of Ã to represent fuzzy set.

Definition2. Let X be a domain. The set of all fuzzy sets on X denoted by FS (x) is called F**uzzy power set of X**.

1.1.6 Representation of a Fuzzy Set

A fuzzy set can be represented by the following ways

(A) Fuzzy set A on X can be represented by, set of ordered pair as follow.

$$A = \{(x, \mu_{\tilde{A}}(x)) : x \in X\}$$

(B) In case if the domain is finite fuzzy set $A = \sum \mu_A(x_i) / x_i$.

For example if A (a) = 0, A (b) = 0.7, A (c) = 0.4 and A (d) = 1. Then, fuzzy set A can be written as A= {(a, 0), {(b, 0.7), (c, 0.4), (d, 1)}

or A = 0/a + 0.7/b + 0.4/c + 1/d.

The symbol $\sum$ and the symbol + in the above notation stand for union of all $\mu_A(x) / x$

(C) In case the domain is continuous $A= \int A(X) / x$.

(D) $A=$ "real numbers close to 10"

$$A=\left\{\left(x,\mu_A(x)\right):\mu_A(x)=\left(1+(x-10)^2\right)^{-1}\right\}.$$

As shown in fig. 1.3.

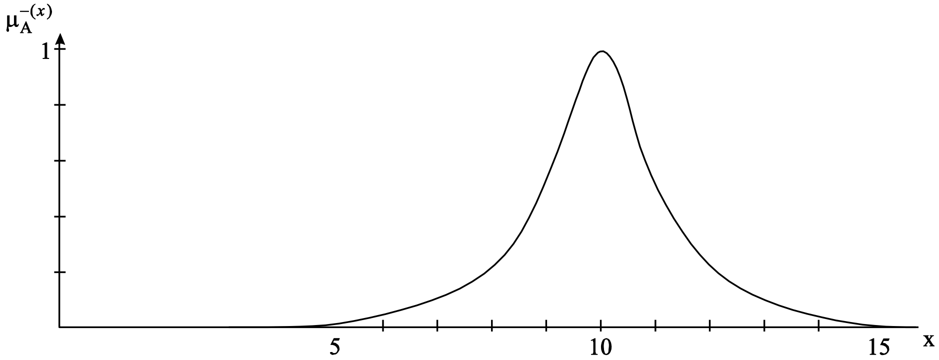


Figure 1.3: Real numbers close to 10

Example 1.1.1 Consider fuzzy set 'two or so'. In this instance, universal set *X* is the positive real numbers.

$$X=\{a,b,c,d,e,f,....\}$$

Membership function for A ='two or so' in this universal set *X* is given as follows:

$$\mu_A(a)=1,\ \mu_A(b)=1,\ \mu_A(c)=0.5,\ \mu_A(d)=0....$$

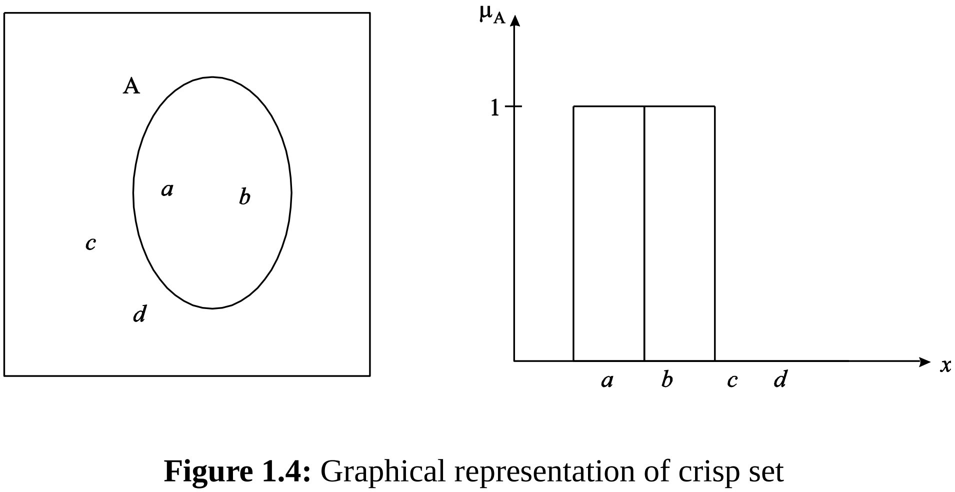


Figure 1.4: Graphical representation of crisp set

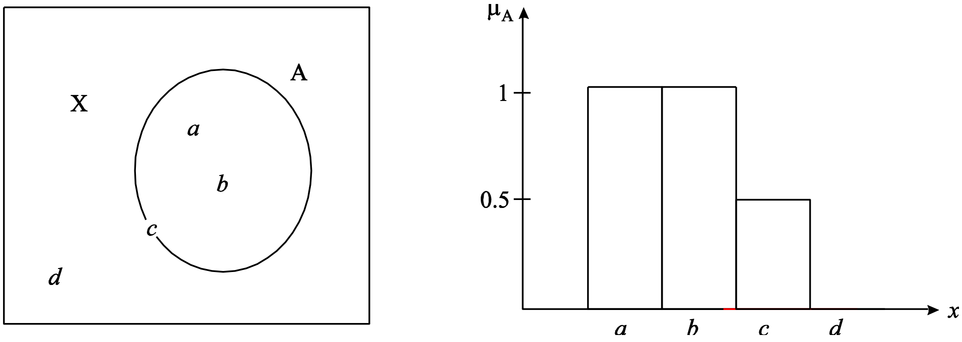


Figure 1.5: Graphical representation of fuzzy set

Usually, if elements are discrete as the above, it is possible to have membership degree or grade as

$$A = \{(a,1.0),\ (b,1.0),\ (c,0.5)\} \text{ or } A = a/1.0 + b/1.0 + c/0.5.$$

Example1.1.2 Let's define a fuzzy set A = {real number near 0}. The boundary for set "real number near 0" is pretty ambiguous. The possibility of real number x to be a member of prescribed set can be defined by the following membership function.

$$\mu_A(x) = \frac{1}{1+x^2}\ .$$

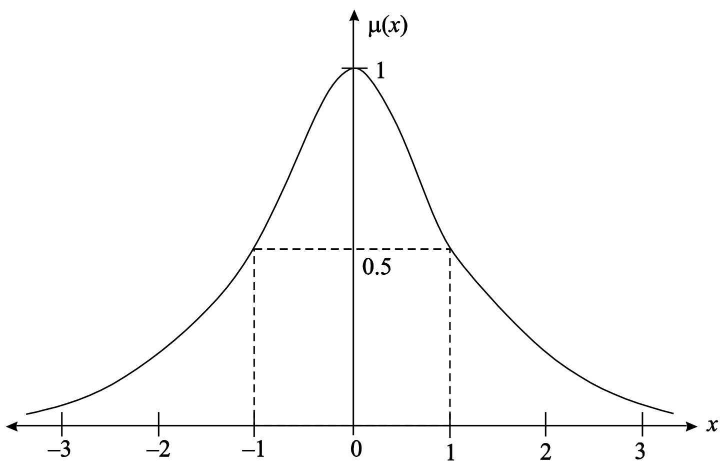


Figure 1.6: Membership function of fuzzy set "real number near 0"

Fig. 1.6 shows this membership function. We can also write the fuzzy set with the function. The membership degree of 1 is 0.5, the possibility of 2 is 0.2 and that of 3 is 0.1.

1.2 TYPES OF FUZZY SETS

(i) **Interval-Valued Fuzzy Sets**. A membership function based on the approach that does not assign to each element of the universal set to one real number, but a closed interval of real numbers between the identified lower and upper bounds. Fuzzy sets defined by membership function of this type are called interval-valued fuzzy sets. These sets are defined as follows

$$A: X \rightarrow \in ([0, 1]),$$

where $P \in ([\, 0, 1\,])$ denote the family of all closed intervals of real numbers in [0, 1].

(ii) **Fuzzy Sets of Type-2**. We can generalize interval-valued fuzzy sets by allotting their intervals to be fuzzy. Each internal now becomes an ordinary fuzzy set defined within the universal set [0, 1]. The membership grades assigned to elements of the universal set by these generalized fuzzy sets are ordinary fuzzy set. These sets are known as 'fuzzy set of type-2. The membership function of this type of set having the following form

$A: X \rightarrow P([0\ 1])$ where $P([0\ 1])$ denotes the set of all ordinary fuzzy sets that can be defined within the universal set [0, 1].

The set $P([0, 1])$ is also called a fuzzy power set of [0, 1].

(iii) **L-Fuzzy Sets**. When we relax the requirement that the membership grades must be represented by numbers in the unit interval [0, 1] and allow them to be represented by symbols of an arbitrary set L, i.e., at least partially ordered, we obtain L-fuzzy set The membership function of L-fuzzy set have the following form

$$A: X \rightarrow L$$

(iv) **Level-2 Fuzzy Sets.** The generalization of ordinary fuzzy sets involves fuzzy sets defined within a universal set whose elements are ordinary fuzzy sets. These fuzzy sets are known as level-2 fuzzy sets. The membership function of level-2 fuzzy set having the following form

$A: \rightarrow C(X) \rightarrow ([0, 1])$ where $C(X)$ denotes the fuzzy power set of X (The set of all ordinary sets).

Example 1.2.1: Consider set A= “adult”. The membership function of this set maps whole age to “youth”, “manhood” and “senior” (Fig. 1.7). For instance, for any person *x*, *y* and *z*,

$$\mu_A(x) = \text{"youth"}, \mu_A(y) = \text{"manhood"}, \mu_A(z) = \phi.$$

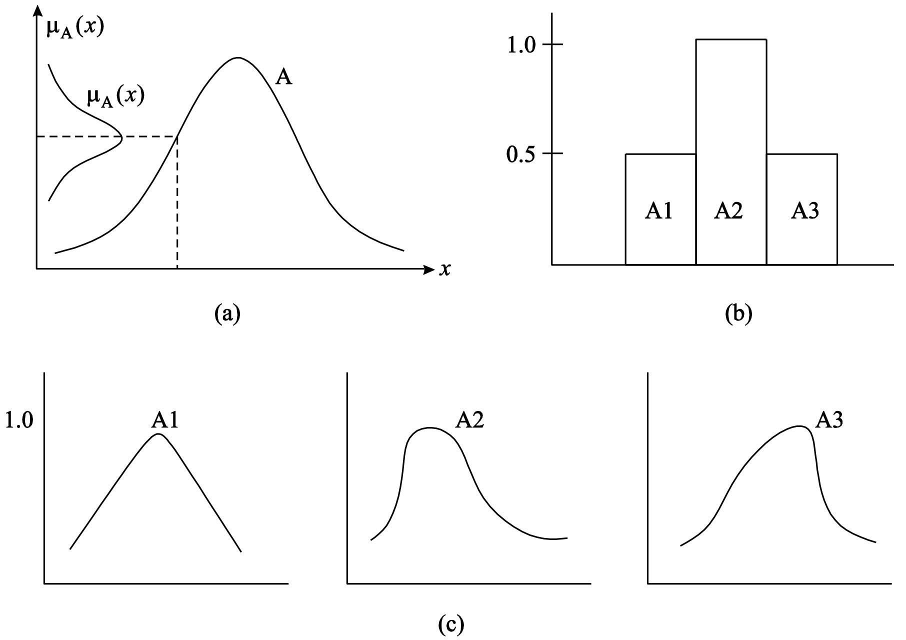


Figure 1.7: Fuzzy set of type - 2(a) and level - 2 fuzyz set {(b), (c)}

The values of membership for "youth" and "manhood" are also fuzzy sets, and thus the set "adult" is a type-2 fuzzy set.

The sets "youth" and "manhood" are type-1 fuzzy sets. In the same manner, if the values of membership function of "youth" and "manhood" are type-2, the set "adult" is type-3.

1.2.1 Support of a Fuzzy Set

Let A be fuzzy set defined on X, then the support of fuzzy set A is a crisp set. It is denoted by Supp (A) and defined as

$$Supp(A) = \{x \in X : \mu_A(x) > 0\}.$$

Example 1.2.2: Consider a universal set *X* which is defined on the age domain

$$X = \{5, 15, 25, 35, 45, 55, 65, 75, 85\}$$

Table 1.2 membership of "infant", "young", "adult" and "senior"

Age	Infant	Young	Adult	Senior
5	0	0	0	0
15	0	0.2	0.1	0
25	0	1	0.9	0
35	0	0.8	1	0
45	0	0.4	1	0.1
55	0	0.1	1	0.2
65	0	0	1	0.6
75	0	0	1	1
85	0	0	1	1

We can define fuzzy sets such as "infant", "young", "adult" and "senior" in *X*. The possibilities of each element of *x* to be in those four fuzzy sets are in Table 1.2.

The support of fuzzy set "young" i.e. $\text{Supp}(\text{young}) = \{15, 25, 35, 45, 55\}$ and it is a crisp set. Certainly, the support of "infant" is empty set.

The maximum value of the membership is called "height". Suppose the "height" of some fuzzy sets is 1, then fuzzy set is "normalized". The sets "young", "adult" and "senior" are normalized as shown in fig. 1.8.

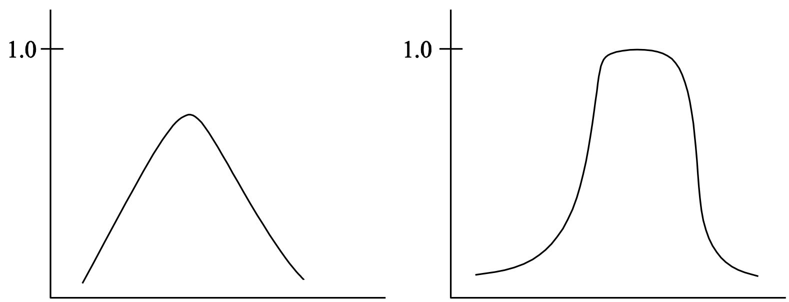


Figure 1.8: Non-normalized and normalized fuzzy set

Let's consider crisp set "teenager". This crisp set is clearly defined having elements only 10-19 in the universal set *X*. As you shall notice this set is a ***restricted set*** comparing with *X*. Similarly fuzzy set "young" is also a restricted set. When we apply a "***fuzzy restriction***" to universal set *X* in certain manner, we get a fuzzy set.

1.2.2 α - Cut Set

Let A be fuzzy set defined on universe of discourse and $\alpha \in [0,1]$ be any number, then α - cut set of A is a crisp set. It is denoted by ${}^{\alpha}\mathrm{A}$ and defined as

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

where strong α - cut set of fuzzy set A is denoted by ${}^{\alpha+}\mathrm{A}$ and defined as

$${}^{\alpha+}A = \{x \in X : \mu_A(x) > \alpha\}.$$

Remark: For any fuzzy set A and pair $\alpha_1, \alpha_2 \in [0, 1]$ of distinct values such that $\alpha_1 < \alpha_2$, we have ${}^{\alpha_1}\mathrm{A} \supseteq {}^{\alpha_2}\mathrm{A}$ and ${}^{\alpha_1+}\mathrm{A} \supseteq {}^{\alpha_2+}\mathrm{A}$.

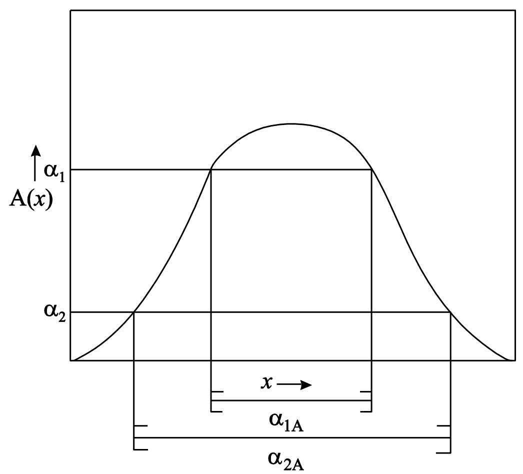


Figure 1.9: Subnormal fuzzy set that is convex

Example 1.2.3: The α -cut set is derived from fuzzy set “young” by giving 0.2 to α this means “the age that we can say young with possibility not less than 0.2”.

If $\alpha = 0.4,\ young_{0.4} = \{25, 35, 45\}$

If $\alpha = 0.8,\ young_{0.8} = \{25, 35\}$.

Remark: The 1- cut, ${}^{1}\mathrm{A}$, is often called the core of fuzzy set A. The height, $h(A)$, of a fuzzy set A is the largest membership grade obtained by any element in that set. Formally, $h(A) = \sup_{x \in X} \mu_A(x)$. A fuzzy set A is called normal when $h(A) = 1$; it is called subnormal when $h(A) < 1$. The height of fuzzy set may also be viewed as the supremum of α for which ${}^{\alpha}\mathrm{A} \neq \phi$.

1.2.3 Level Set

The value α which explicitly shows the value of the membership function, is in the range of [*0, 1*]. The "level set" is obtained by the $\alpha's.$ That is,

$$\Lambda_A = \{\alpha : \mu_A(x) = \alpha, \alpha \geq 0, x \in X\}.$$

The level set of above fuzzy set "young" is,

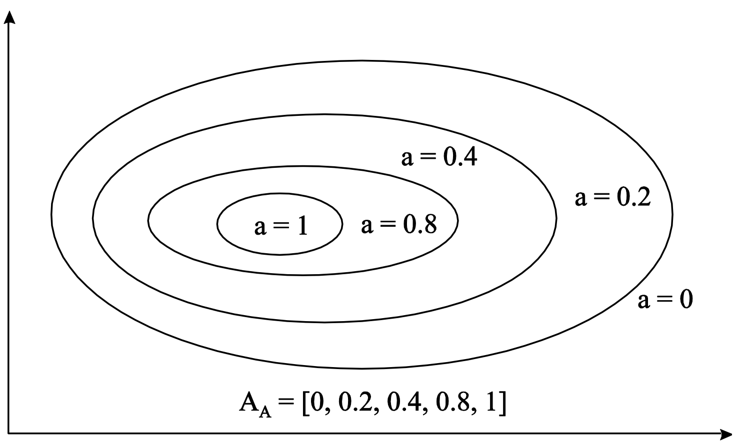
$$\Lambda_A = \{0,\ 0.1,\ 0.2,\ 0.4,\ 0.8,\ 1.0\}.$$

1.2.4 Convex Fuzzy Set

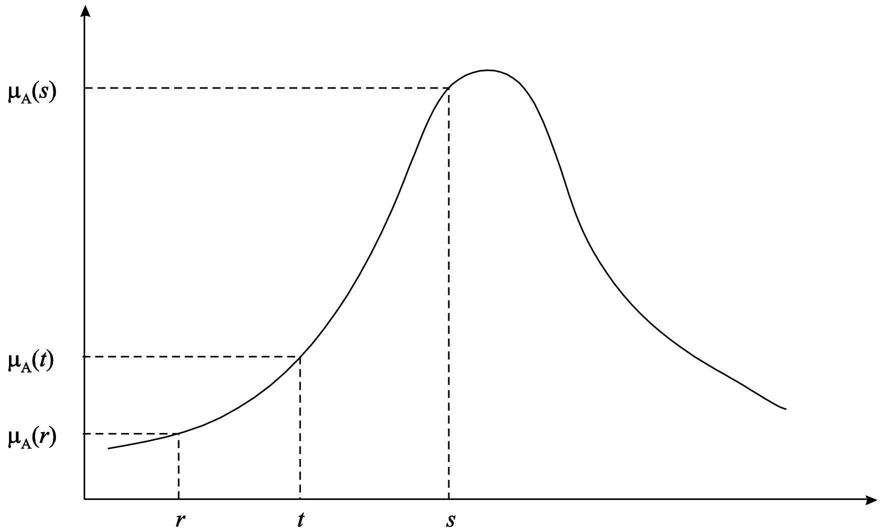
Assuming universal set X is defined in n-dimensional Euclidean Vector space $\Re^n$. If all the α - cut sets are convex, the fuzzy set with these α - cut sets is convex (Fig. 1.10(a)). In other words, if a relation

$$\mu_A(t) \geq \text{Min}\left[\mu_A(r), \mu_A(s)\right],$$

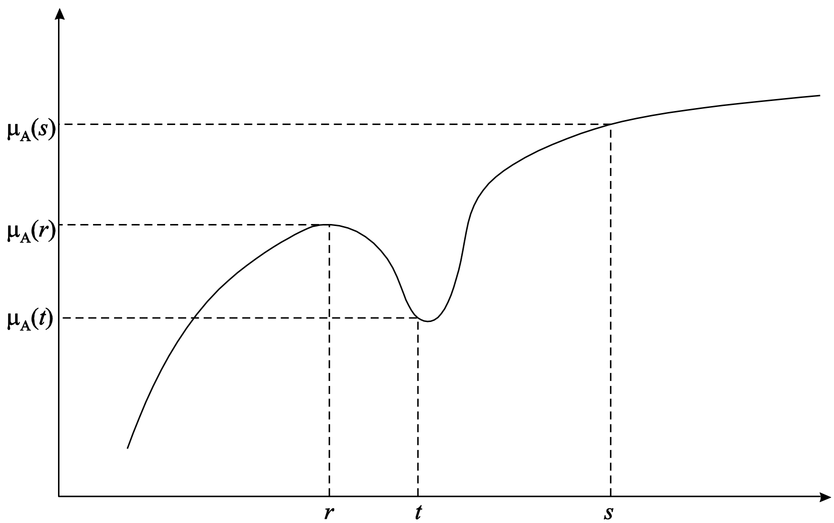
where $t = \lambda r + (1-\lambda)s;\ r, s \in \Re^n, \lambda \in [0,\ 1]$ holds, the fuzzy set is convex. Fig. 1.10(b) shows a convex fuzzy set and Fig. 1.10(c) describes a non-convex set.



(a) Convex Fuzzy Set



(b) Convex Fuzzy Set $\mu_A(t) \geq \mu_A(r)$



(c) Non- Convex Fuzzy Set $\mu_A(t) < \mu_A(r)$

Figure 1.10: Convex and non-convex fuzzy sets

1.2.5 Magnitude of Fuzzy set

There are three ways of measuring the cardinality of fuzzy set. First, we can derive magnitude by summing up the membership degrees. It is "***scalar cardinality***".

$$|A| = \sum_{x \in X} \mu_A(x).$$

Following this method, the magnitude of fuzzy set "senior" is,

$$|\text{senior}| = 0.1 + 0.2 + 0.6 + 1 + 1 = 2.9.$$

Second, comparing the magnitude of fuzzy set A with that of universal set X can be an idea.

$$\|A\| = \frac{|A|}{|X|}.$$

This is called "relative cardinality". In the case of "senior",

$$|\text{senior}| = 2.9 \quad |X| = 9 \text{ and } \|\text{senior}\| = 2.9/9 = 0.32.$$

Third method expresses the cardinality of fuzzy set as explained below:

Let ${}^{\alpha}\mathrm{A}$ be the α -cut of A. The number of elements is $\left|{}^{\alpha}\mathrm{A}\right|$. In other words, the possibility for number of elements in A to be $\left|{}^{\alpha}\mathrm{A}\right|$ is α. Then the membership degree of fuzzy cardinality $|A|$ is defined as

$$\mu_{|A|}\left(\left|{}^{\alpha}\mathrm{A}\right|\right) = \alpha,\ \alpha \in \Lambda_A, \text{ where } \Lambda_A \text{ is a level set.}$$

Example 1.2.5: If we cut fuzzy set "senior" at $\alpha = 0.1$, there are 5 elements in the α -cut. $\text{senior}_{0.1} = \{45, 55, 65, 75, 85\}$, $|\text{senior}_{0.1}| = 5$. In the same manner, there are 4 elements at $\alpha = 0.2$, there are 3 elements $\alpha = 0.6$, there are 2 elements $\alpha = 1.0$. Therefore, the fuzzy cardinality of "senior" is

$$|\text{senior}| = \{(5, 0.1), (4, 0.2), (3, 0.6), (2, 1)\}.$$

1.3 OPERATIONS ON FUZZY SETS

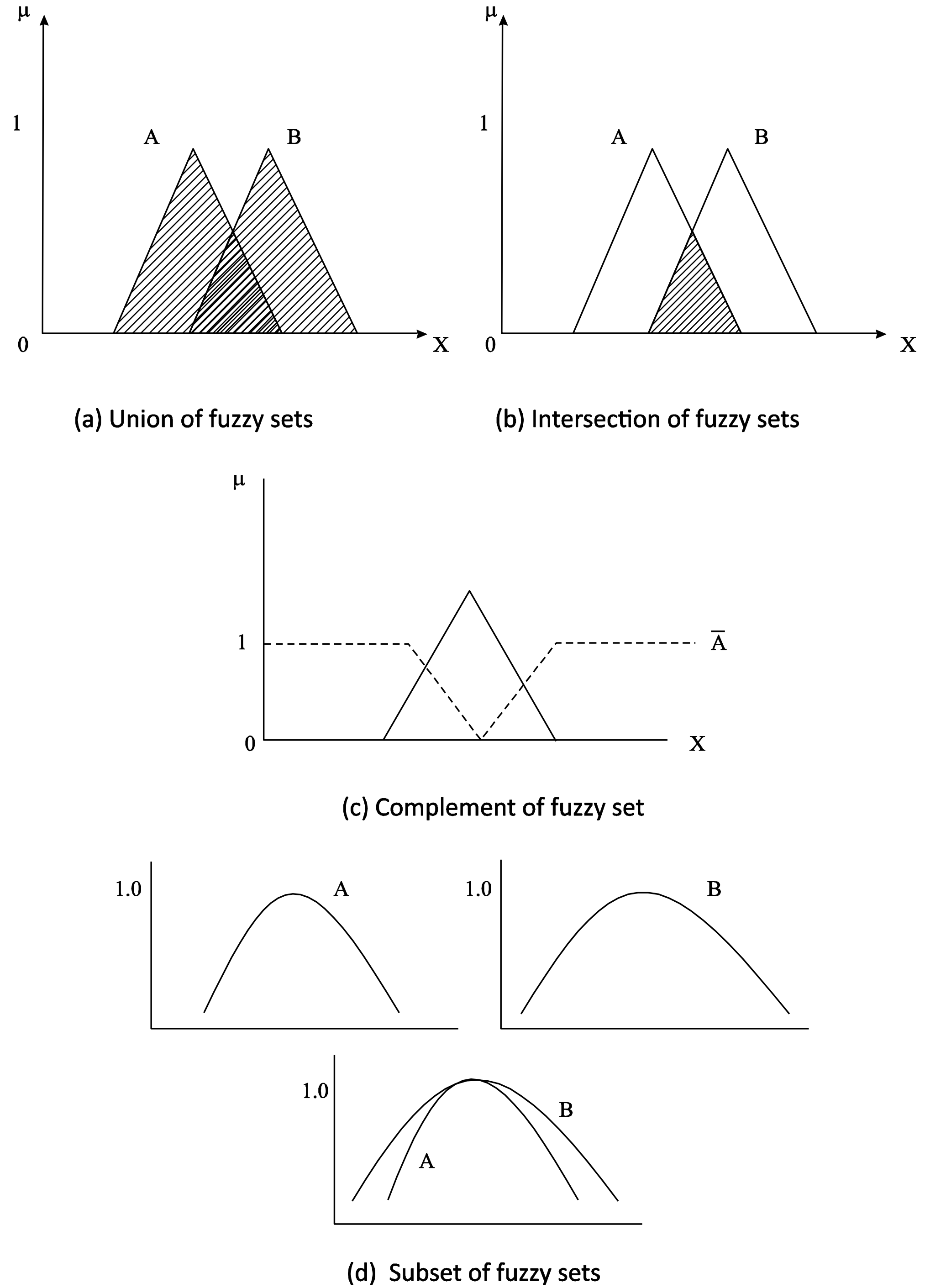
Considering three fuzzy sets A, B and C on the universe X. For a given element x of the universe, the following function theoretic operations for the set theoretic operations unions, intersection and complement are defined for A, B and C on X:

Union: $\mu_{A \cup B}(x) = \mu_A(x) \vee \mu_B(x) = \max\left(\mu_A(x), \mu_B(x)\right).$

Intersection: $\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x) = \min\left(\mu_A(x), \mu_B(x)\right).$

Complement: $\mu_{\bar{A}}(x) = 1 - \mu_A(x).$

Subset: $A \subseteq B \Rightarrow \mu_A(x) \le \mu_B(x)$, for all $x \in X$, $\mu_{\phi}(x) = 0$ and $\mu_X(x) = 1.$



(a) Union of fuzzy sets

(b) Intersection of fuzzy sets

(c) Complement of fuzzy set

(d) Subset of fuzzy sets

Figure 1.11: Venn diagram of operation on fuzzy sets

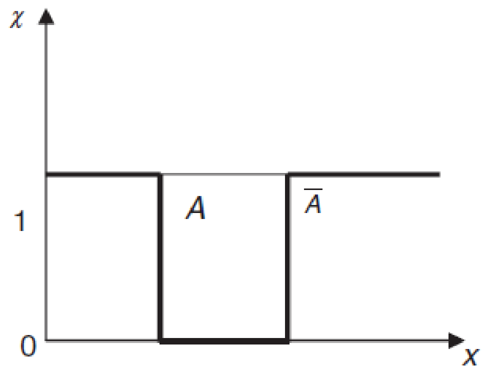
Any fuzzy set A defined on a universe X is a subset of that universe. The membership value of any element *x* in the null set ϕ is 0, and the membership value of any element *x* in the whole set *x* is 1. This statement is given by De Morgan's laws stated for classical sets also hold for fuzzy sets, as denoted by the following expressions:

$\overline{A \cap B} = \overline{A} \cup \overline{B}$ and $\overline{A \cup B} = \overline{A} \cap \overline{B}$.

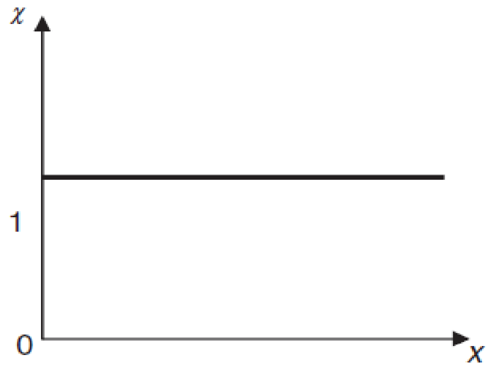
All operations on classical sets also hold for the fuzzy set except for the excluded middle laws. These two laws does not hold good for fuzzy sets. Since fuzzy sets can overlap, a set and its complement also can overlap.

The excluded middle law for fuzzy sets is given by

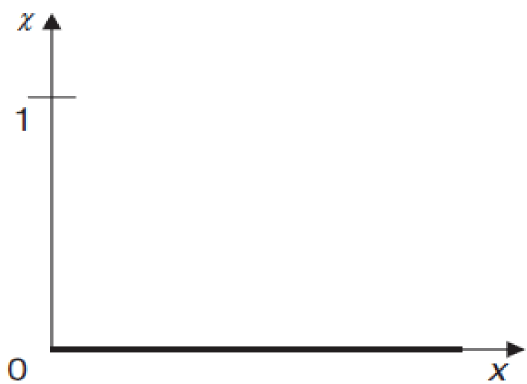
$A \cup \overline{A} \neq X$ and $A \cap \overline{A} \neq \phi$.



(a) Crisp set A and its complement

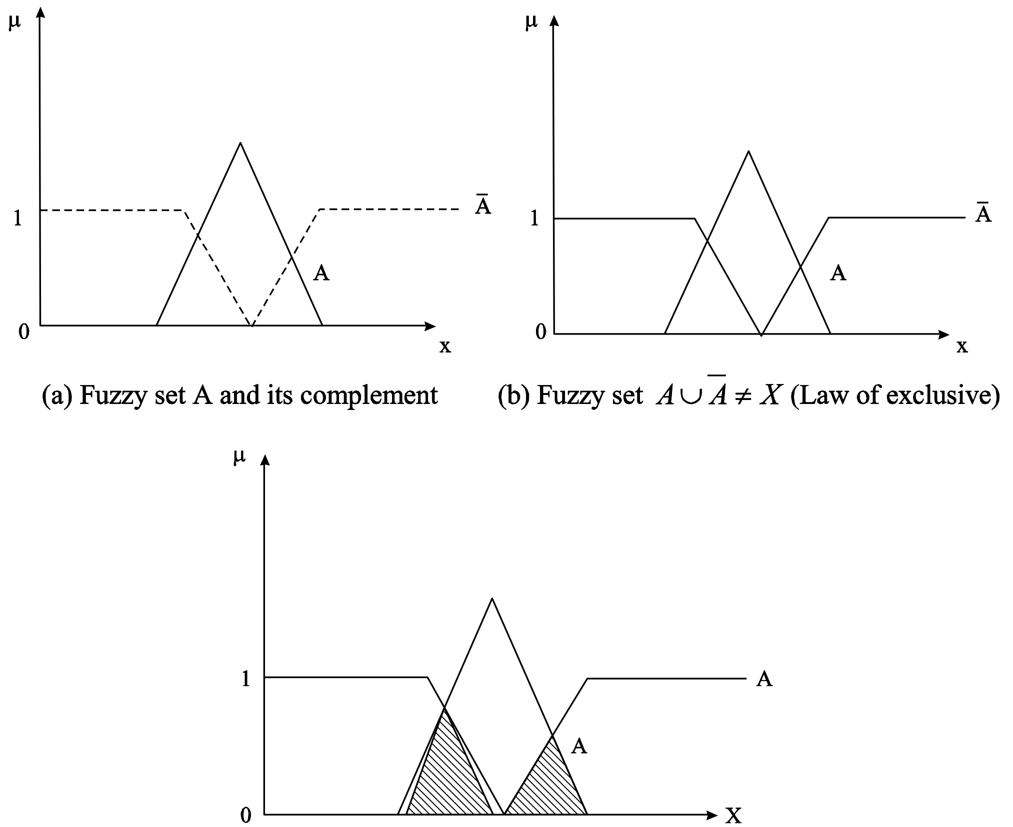


(b) Crisp set $A \cup \overline{A} = X$ (Law of exclusive)



(c) Crisp set $A \cap \overline{A} = \phi$ (Law of contradiction)

Figure 1.12: Excluded middle law for classical sets



(a) Fuzzy set A and its complement (b) Fuzzy set $A \cup \overline{A} \neq X$ (Law of exclusive)

(c) Crisp set $A \cap \overline{A} = \phi$ (Law of contradiction)

Figure 1.13: Excluded middle law for Fuzzy sets

Comparing Venn diagram for classical sets and fuzzy sets for excluded middle law are shown in Fig.1.6 and Fig.1.7 respectively.

1.3.1 Properties of Fuzzy Sets

The properties of the classical set also suits for the properties of the fuzzy sets. The important properties of fuzzy sets are as follows:

Commutativity: $A \cup B = B \cup A$ and $A \cap B = B \cap A$.

Associativity: $A \cup (B \cup C) = (A \cup B) \cup C$ and $A \cap (B \cap C) = (A \cap B) \cap C$.

Ditributivity: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ and $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Idempotency: $A \cup A = A$ and $A \cap A = A$.

Identity: $A \cup \phi = A$ and $A \cap X = A$, $A \cap \phi = \phi$ and $A \cup X = X$.

Transitivity: If $A \subset B \subset C$ then $A \subset C$.

Involution: $\overline{\overline{A}} = A$.

These are the important properties of the fuzzy set.

Example 1.3.1: Consider two fuzzy sets A and B. find Complement, Union, Intersection, Difference, and De Morgan's law.

$$A = \left\{\frac{1}{2} + \frac{0.5}{3} + \frac{0.6}{4} + \frac{0.2}{5} + \frac{0.6}{6}\right\} \text{ and } B = \left\{\frac{0.5}{2} + \frac{0.8}{3} + \frac{0.4}{4} + \frac{0.7}{5} + \frac{0.3}{6}\right\}.$$

Complement

$$\overline{A} = \left\{\frac{0}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.8}{5} + \frac{0.4}{6}\right\} \text{ and } \overline{B} = \left\{\frac{0.5}{2} + \frac{0.2}{3} + \frac{0.6}{4} + \frac{0.3}{5} + \frac{0.7}{6}\right\}.$$

Union

$$A \cup B = \left\{\frac{1}{2} + \frac{0.8}{3} + \frac{0.6}{4} + \frac{0.7}{5} + \frac{0.6}{6}\right\}.$$

Comparing the membership values and writing maximum of the two values determine Union of the fuzzy set.

Intersection

$$A \cap B = \left\{\frac{0.5}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.2}{5} + \frac{0.3}{6}\right\}.$$

Comparing the membership values and writing minimum of the two values determine intersection of the fuzzy set.

Difference

$$A/B = A \cap \overline{B} = \left\{\frac{0.5}{2} + \frac{0.2}{3} + \frac{0.6}{4} + \frac{0.2}{5} + \frac{0.6}{6}\right\},$$

$$B/A = B \cap \overline{A} = \left\{\frac{0}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.7}{5} + \frac{0.3}{6}\right\}.$$

De Morgan's Laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B} = \left\{\frac{0}{2} + \frac{0.2}{3} + \frac{0.4}{4} + \frac{0.3}{5} + \frac{0.4}{6}\right\},$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B} = \left\{\frac{0.5}{2} + \frac{0.5}{3} + \frac{0.6}{4} + \frac{0.8}{5} + \frac{0.7}{6}\right\}.$$

Example 1.3.2: We want to compare two sensors based upon their detection levels and gain settings. The following table 1.3 of gain settings and sensor detection levels with a standard item being monitored provides typical membership values to represent the detection levels for each of the sensors.

Table 1.3

Gain setting	Sensor 1 detection levels	Sensor 2 detection levels
0	0	0
20	0.5	0.35
40	0.65	0.5
60	0.85	0.75
80	1	0.90
100	1	1

The universe of discourse is $X = \{0, 20, 40, 60, 80, 100\}$ Find the membership function for the two sensors: Find the following membership functions using standard set operations:

(a) $\mu_{S_1 \cup S_2}(x)$ (b) $\mu_{S_1 \cup S_2}(x)$ (c) $\mu_{\bar{S}_1}(x)$ (d) $\mu_{\bar{S}_2}(x)$

Solution: It can be solved similar as previous example.

1.3.2 Fuzzy Complement

Complement set $\bar{A}$ of set *A* carries the sense of negation. Complement set may be defined by the following function *C*.

$C : [0, 1] \to [0, 1]$

Complement function *C* is designed to map membership function $\mu_A(x)$ of fuzzy set *A* to [0,1] and the mapped value is written as $C(\mu_A(x))$. To be a fuzzy complement function, the following two axioms should be satisfied:

(C_1) $C(0) = 1, C(1) = 0$ (boundary condition).

(C_2) If $a, b \in [0, 1]$ and $a < b$, then $C(a) \geq C(b)$ (monotonic non increasing), where *a* and *b* stand for membership value of member *x* in *A*. For example, if $\mu_A(x) = a$, $\mu_A(y) = b; x, y \in X$, then $\mu_A(x) < \mu_A(y), C(\mu_A(x)) \geq C(\mu_A(y))$.

(C_1) and (C_2) are fundamental requisites to be a complement function. These two axioms are called "axiomatic skeleton". For particular purposes, we can insert the following additional requirements:

(C_3) C is a continuous function.

(C_4) C is a involutive. *i.e.*, $C(C(a)) = a$ for all $a \in [0, 1]$.

Example 1.3.3: Above four axioms hold in standard complement operator

$$C(\mu_A(x)) = 1 - \mu_A(x) \text{ or } \mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

this standard function is shown in Fig. 1.14(a), and it's visual representation is given in Fig 1.14(b).

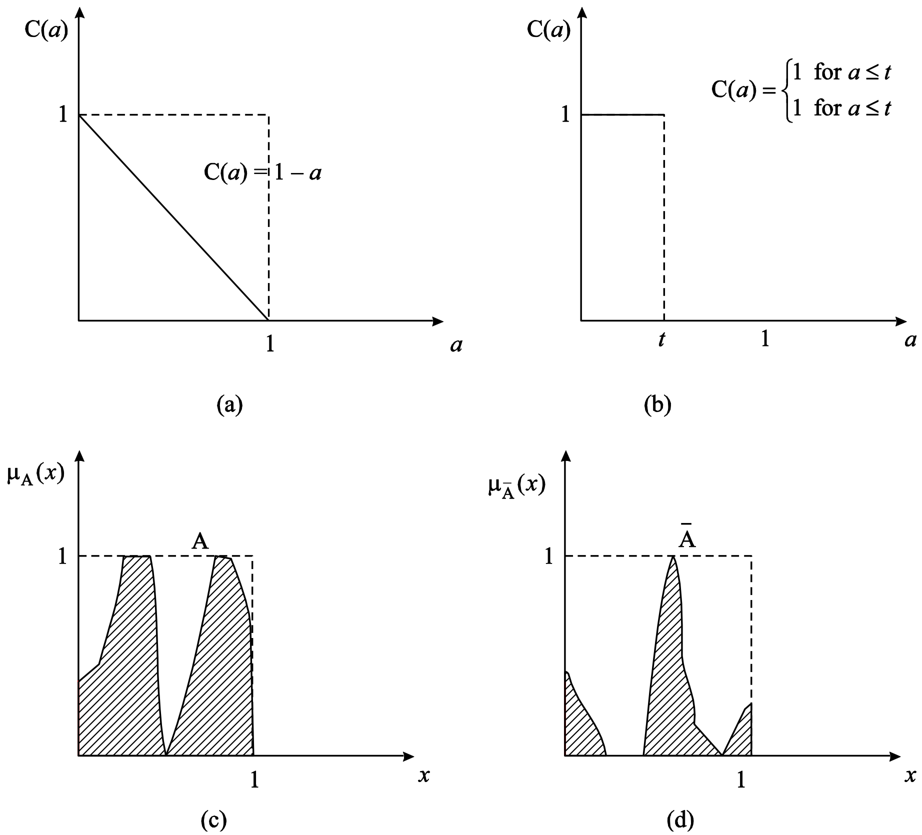


Fig. 1.14 Axioms in standard complement operator

The following is a complement function satisfying only the axiomatic skeleton (Fig 1.14(b)):

$$C(a) = \begin{cases} 1, \text{ for } a \le t \\ 0, \text{ for } a > t. \end{cases}$$

Note that it does not hold in fig.1.14(c) and fig.1.14 (d).

Again, the following as shown in Fig 1.15:

$$C(a) = 0.5(1 + \cos \pi a),$$

when $a = 0.33$, $C(0.33) = 0.75$ is continuous. However, when $C(0.75) = 0.15 \neq 0.33$, $C(a)$ is not continuous. One of the popular complement functions is Yager's function defined as

$$C_w(a) = (1 - a^w)^{1/w}, \text{ where } w \in]-1, \infty[.$$

The shape of the function is dependent on parameter (Fig 1.16). When $w = 1$, the Yager's function becomes the standard complement function $C(a) = 1 - a$.

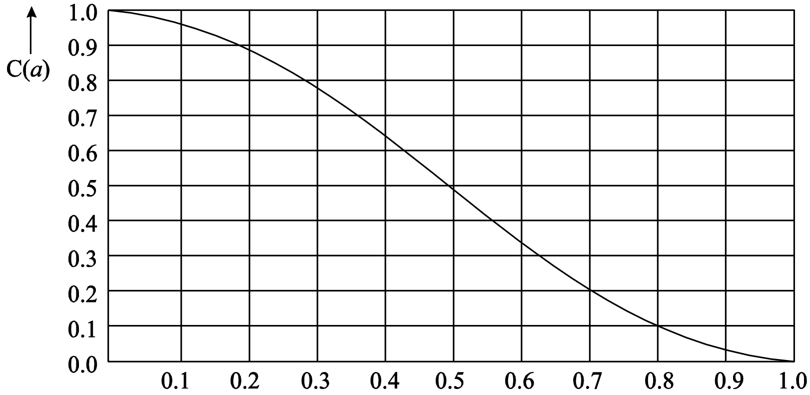


Figure 1.15: Continuous fuzzy complement function $C(a) = 0.5(1 + \cos \pi a)$

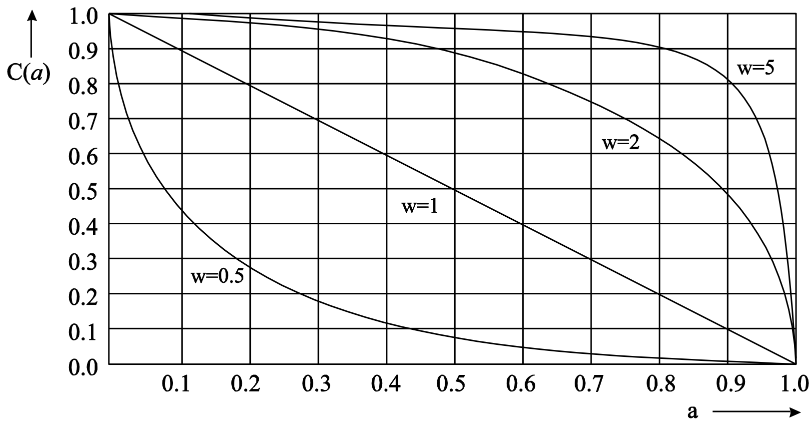


Figure 1.16: Yager's complement function

1.3.3 Fuzzy Union Function

In general sense, union of *A* and *B* is specified by a function of the form:

$$U : [0, 1] \times [0, 1] \rightarrow [0, 1].$$

This union function calculates the membership degree of union $A \cup B$ from those of *A* and *B*,

$$\mu_{A \cup B}(x) = U[\mu_A(x), \mu_B(x)]$$

this union function should obey the following axioms:

(U_1) $U(0,\ 0) = 0, U(0,\ 1) = 1,\ U(1,\ 0) = 1,\ U(1,\ 1) = 1$ (Boundary condition)

(U_2) $U(a,b) = U(b,a)$ (Commutative)

(U_3) If $a \le a'$ and $b \le b'$, $U(a, b) \le U(a', b')$. Function U is monotonic function.

(U_4) $U\left(U(a,b),c\right) = U\left(a, U(b, c)\right)$ (Associative)

(U_5) Function U is continuous.

(U_6) $U(a, a) = a$ (Idempotent)

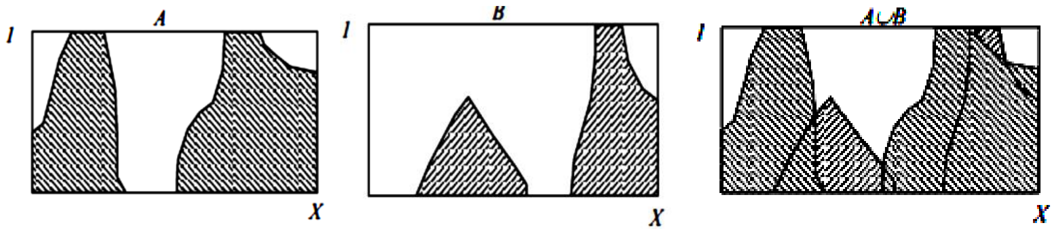


Figure 1.17: Visualization of standard union operation

The standard operator Max is treading on those six axioms.

$$U[\mu_A(x), \mu_B(x)] = \max\left[\mu_A(x), \mu_B(x)\right] \text{ or } \mu_{A\cup B}(x) = \max\left[\mu_A(x), \mu_B(x)\right]$$

visualizing the standard union operation is illustrated in figure 1.17.

Yager's union function satisfies all axioms except (U_6).

$$U_w(a, b) = \min\left[1, (a^w + b^w)^{1/w}\right], \quad \textit{where } w \in]0, \infty[.$$

The shape of Yager function varies with parameter *w*. For instance,

$w = 1$ leads to $U_1(a, b) = \min\left[1, (a+b)\right]$.

$w = 2$ leads to $U_2(a, b) = \min\left[1, \sqrt{a^2 + b^2}\right]$.

Supposing $w \to \infty$, Yager union function is transformed into the standard union function $\lim\limits_{w\to\infty} \min\left[1, (a^w + b^w)^{1/w}\right] = \max(a, b)$.

There are some examples of Yager function for *w* =1, 2 and ∞ *in* Tables 1.4. We know that the union operation of crisp sets is identical to OR logic. It is easy to see thet the relation is also preserved here. For example, if set *A* be "young" and *B* "senior", the union of *A* and *B* is "young or senior". In the sense of meaning, the union and OR logic are completely identical.

Table 1.4 Yager's union function

(a) $U_1(a,b) = \min[1, (a+b)]$

a \ b	0	0.25	0.5
1	1	1	1
0.75	0.75	1	1
0.25	0.25	0.5	0.75

(b) $U_2(a,b) = \min\left[1, \sqrt{a^2+b^2}\right]$

a \ b	0	0.25	0.5
1	1	1	1
0.75	0.75	0.79	0.9
0.25	0.25	0.35	0.55

(c) $U_\infty = \max(a,b)$

a \ b	0	0.25	0.5
1	1	1	1
0.75	0.75	0.75	0.75
0.25	0.25	0.25	0.5

1.3.4 Fuzzy Intersection Function

In general sense, intersection of *A* and *B* is specified by a function of the form

$I:[0, 1]\times[0, 1]\rightarrow[0, 1].$

This intersection function calculates the membership degree of union $A\cap B$ from those of *A* and *B*,

$$\mu_{A\cup B}(x) = I[\mu_A(x), \mu_B(x)]$$

this intersection function should obey the followings:

(I_1) $I(0, 0)=0, I(0, 1)=0, I(1, 0)=0, I(1, 1)=1$ (boundary condition).

(I_2) $I(a,b)=I(b,a)$ (commutativity).

(I_3) If $a\le a'$ and $b\le b'$, $I(a, b)\le I(a', b')$. Function I is monotonic function.

(I_4) $I\left(I(a,b),c\right)=I\left(a,I(b, c)\right)$ (associativity).

(I_5) Function I is continuous.

(I_6) $I(a, a)=a$ (idempotency).

The standard fuzzy intersection satisfies the above six axioms with

$$I[\mu_A(x), \mu_B(x)] = \min\left[\mu_A(x), \mu_B(x)\right] \text{ or } \mu_{A\cap B}(x) = \min\left[\mu_A(x), \mu_B(x)\right].$$

Visualizing the standard intersection union operation leads to Fig. 1.18.

Yager's function as constructed in previous section satisfies all the axioms except (I_6), where

$$I_w(a, b) = 1-\min\left[1, ((1-a)^w+(1-b)^w)^{1/w}\right], \quad where\ w\in]0, \infty[.$$

However, the shape varies with parameter *w*. For instance,

$w=1$ leads to $I_1(a,\, b)=1-\min\left[1,\, (2-a-b)\right]$.

$w=2$ leads to $I_2(a,\, b)=1-\min\left[1,\, \sqrt{(1-a)^2+(1-b)^2}\,\right]$.

Supposing $w \to \infty$, Yager function converges to the standard intersection function

$$\lim_{w\to\infty}\left(1-\min\left[1,\, ((1-a)^w+(1-b)^w)^{1/w}\right]\right)=\min(a,\, b).$$

Note that intersection and *AND* logic are equivalent. For instance, consider two fuzzy sets "young" and "senior". Intersection for these is "person who is at once young and senior".

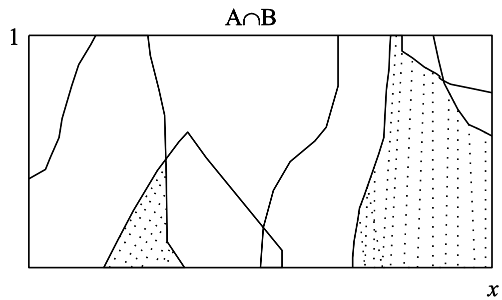


Figure 1.18: Visualization of standard fuzzy intersection set

Example 1.3.4: Take Yager function at *w = 1* for example. Taking *a* = 0.4 and *b* = 0.6, we have

$$I_1(a,\, b)=1-\min\left[1,\, (2-a-b)\right]=1-\min\left[1,\, 2-1\right]=1-1=0.$$

This time let *a = 0.5* and *b = 0.6*, then *a+b=1.1*

$$I_1(a,\, b)=1-\min\left[1,\, 2-1.1\right]=1-\min[1,\, 2-1.1]=0.1.$$

Next, take *a = 0.3* and *b = 0.6* for an example. If $w \to \infty$, the intersection is reduced to,

$$I_\infty(a,\, b)=\min[0.3,\, 0.6]=0.3.$$

There are some more examples of Yager function in Tables 1.5.

Tables 1.5 Yager's intersection function

(a) $I_1(a,b) = 1 - \min\left[1, (2-a-b)\right]$ (b) $I_2(a,b) = 1 - \min\left[1, \sqrt{(1-a)^2 + (1-b)^2}\right]$ (c) $I_\infty(a,b) = \min(a,b)$

a \ b	0	0.25	0.5
1	0	0.25	0.5
0.75	0	0	0.25
0.25	0	0	0

a \ b	0	0.25	0.5
1	0	0.25	0.5
0.75	0	0.21	0.44
0.25	0	0	0.1

a \ b	0	0.25	0.5
1	0	0.25	0.5
0.75	0	0.25	0.5
0.25	0	0.25	0.25

1.3.5 Some Other Operations

(i) Simple disjunctive sum

By means of fuzzy union and fuzzy intersection, definition of the disjunctive sum in fuzzy set is defined just like in crisp set as given below:

$$A \oplus B = (A \cap \overline{B}) \cup (\overline{A} \cap B) \text{ (crisp set)}$$

and $A \oplus B = \max\left\{\min[\mu_A(x), 1-\mu_B(x)], \min[1-\mu_A(x), \mu_B(x)]\right\}.$

(ii) Disjoint sum

The key idea of "exclusive OR" is elimination of common area from the union of A and B. With this idea, we can define an operator Δ for the exclusive OR disjoint sum as follows

$$\mu_{A\Delta B}(x) = \left|\mu_A(x) - \mu_B(x)\right|.$$

Example 1.3.5: Here we discuss procedures obtaining disjunctive sum and disjoint sum of *A* and *B* given Figures 1.19 (a) and 1.19 (b), respectively.

$$A = \left\{(x_1, 0.2), (x_2, 0.7), (x_3, 1), (x_4, 0)\right\},\ B = \left\{(x_1, 0.5), (x_2, 0.3), (x_3, 1), (x_4, 0.1)\right\}$$

$$A = \left\{(x_1, 0.8), (x_2, 0.3), (x_3, 0), (x_4, 1)\right\},\ B = \left\{(x_1, 0.5), (x_2, 0.7), (x_3, 0), (x_4, 0.9)\right\}$$

$$A \cap \overline{B} = \left\{(x_1, 0.2), (x_2, 0.7), (x_3, 0), (x_4, 0)\right\},\ \overline{A} \cap B = \left\{(x_1, 0.5), (x_2, 0.3), (x_3, 0), (x_4, 0.1)\right\}$$

and finally, we have

$$A \oplus B = (A \cap \overline{B}) \cup (\overline{A} \cap B) = \{(x_1, 0.5), (x_2, 0.7), (x_3, 0), (x_4, 0.1)\}$$

$$A \Delta B = \{(x_1, 0.3), (x_2, 0.4), (x_3, 0), (x_4, 0.1)\}.$$

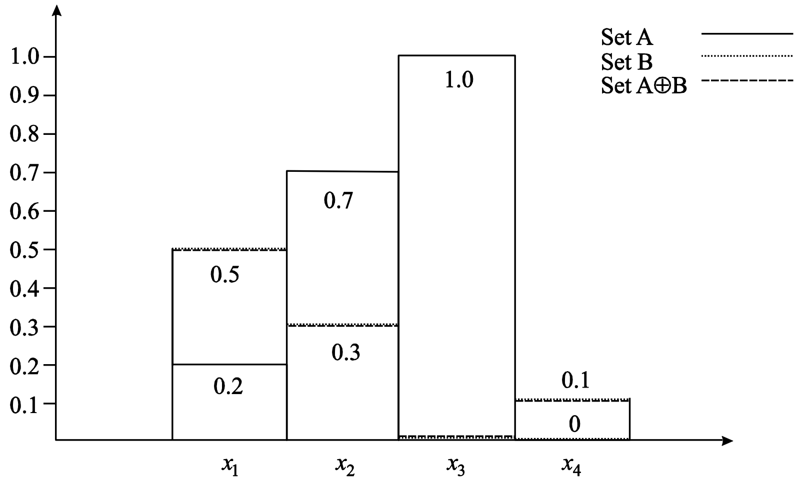


Figure 1.19(a): Example of simple disjunctive sum

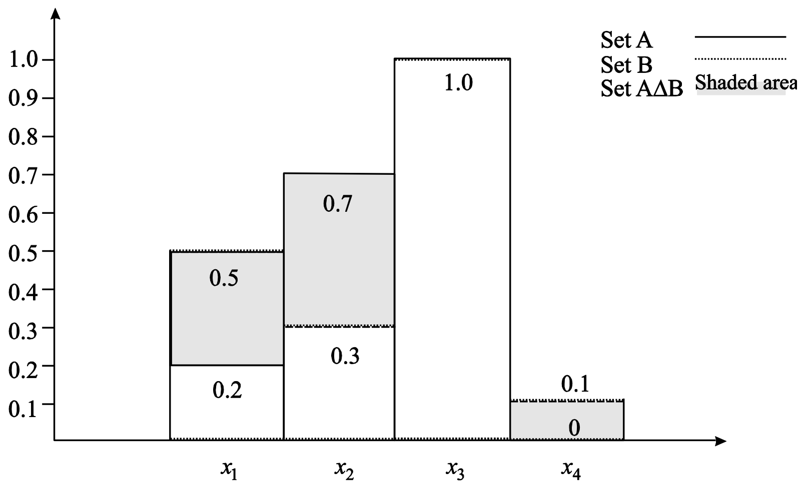


Figure 1.19(b): Example of disjoint sum

(i) Simple difference in fuzzy set

The difference in crisp set is defined as follows

$$A - B = A \cap \overline{B}.$$

By using standard complement and intersection operations, the difference operation in fuzzy set would be simple.

(ii) Bounded difference

For novice-operator θ, we define the membership function as

$$\mu_{A\theta B}(x) = \max[0, \mu_A(x) - \mu_B(x)].$$

Example 1.3.6: If we reconsider the previews example, $A - B$ and $A\theta B$ (Figures 1.20 and 1.21).

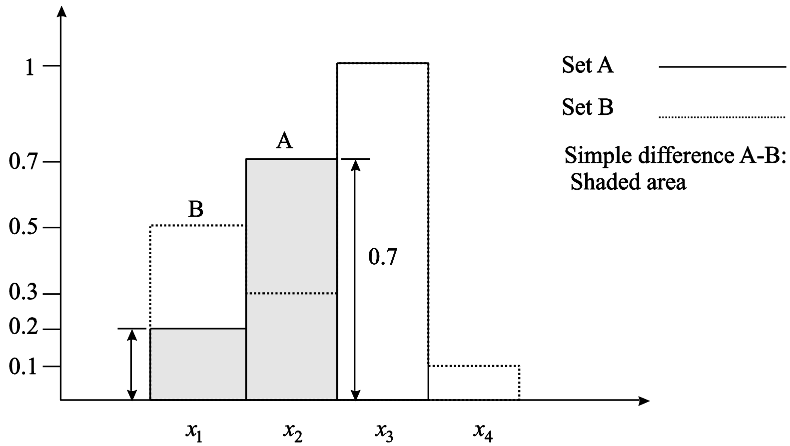


Figure 1.20: Simple difference $A - B$

$A = \{(x_1, 0.2), (x_2, 0.7), (x_3, 1), (x_4, 0)\}$, $B = \{(x_1, 0.5), (x_2, 0.3), (x_3, 1), (x_4, 0.1)\}$

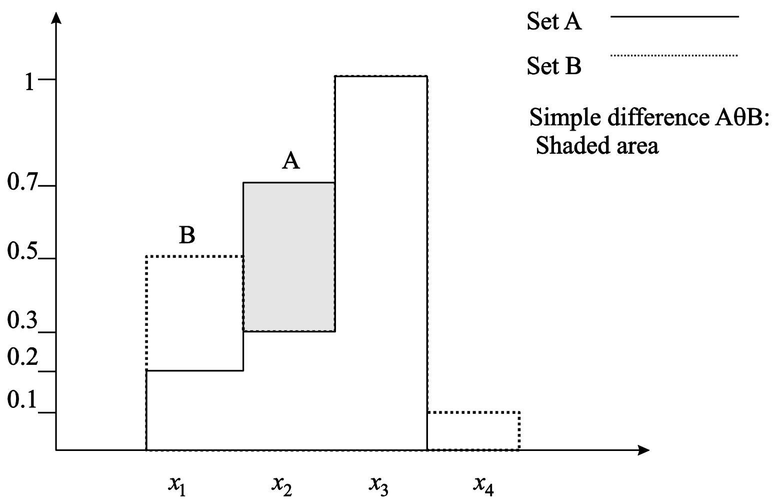


Figure 1.21: Bounded difference $A\theta B$

Simple difference is

$$A - B = \{(x_1, 0.2), (x_2, 0.7), (x_3, 0), (x_4, 0)\}.$$

By this definition, bounded difference of preceding two fuzzy sets is as follows

$$A\theta B=\{(x_1,0),(x_2,0.4),(x_3,0),(x_4,0)\}.$$

1.3.6 t-norms and t-conorms

There are two types of operators in fuzzy sets: t-norm and t-conorm. These are often called as triangular-norm and triangular-conorm respectively.

Definition (t-norm): The fuzzy intersection/t-norms of two fuzzy sets A and B is specified in general by a binary operation on the unit interval. In other words a function $T:[0,1]\times[0,1]\to[0,1]$ which satisfies the following axioms: for all $x,\ y,\ x',\ y',\ z\in[0,1]$:

(T_1) $T(x,0)=0, T(x,1)=x$ (boundary condition).

(T_2) $T(x,y)=T(y,x)$ (commutativity).

(T_3) If $x\le x', y\le y' \Rightarrow T(x,y)\le T(x',y')$ (monotonicity).

(T_4) $T(T(x,y),z)=T(x,T(y,z))$ (associativity).

Now we can easily recognize that the following operators hold conditions for t-norm.

(a) Intersection operator $(\cap)$

(b) Algebraic product operator $(\bullet)$

(c) Bounded product operator $(\odot)$

(d) Drastic product operator $(\ddot{\cap})$

Definition (t-conorm (s-norm)): The fuuzy union/t-conorm/s-norm of two fuzzy sets A and B is specified in general by a binary operation on the unit interval. In other words it is a function $S:[0,1]\times[0,1]\to[0,1]$ satisfies the following axioms: for all $x,\ y,\ x',\ y',\ z\in[0,1]$:

(T_1) $S(x,0)=x, S(x,1)=1$ (boundary condition).

(T_2) $S(x,y)=S(y,x)$ (commutativity).

(T_3) If $x\le x', y\le y' \Rightarrow S(x,y)\le S(x',y')$ (monotonicity).

(T_4) $S(S(x,y),z)=S(x,S(y,z))$ (associativity).

There are examples of t-conorm/s-norm operators.

(a) Union operator $(\cup)$

(b) Algebraic sum operator $(\hat{+})$

(c) Bounded sum operator $(\oplus)$

(d) Drastic sum operator $(\dot{\cup})$

(e) Disjoint sum operator (Δ)

1.4 FUZZY NUMBERS

Among the various types of fuzzy sets, of special significance are fuzzy sets that are defined on the set *R* of real numbers. Membership functions of these sets as

$A : R \rightarrow [0, 1].$

These clearly have a quantitative meaning and may, under certain conditions, be viewed as fuzzy numbers or fuzzy intervals. To view them in this way, they should capture our intuitive conceptions of approximate numbers or intervals, such as "numbers that are close to a given real number" or "numbers that are around a given interval of real numbers." Such concepts are essential for characterizing states of fuzzy variables and, consequently, play an important role in many applications, including fuzzy control, decision making, approximate reasoning, optimization, and statistics with imprecise probabilities.

To qualify as a *fuzzy number,* a fuzzy set A on R must possess at least the following three properties:

(i) A must be normal fuzzy set;

(ii) ${}^{\alpha}$A must be closed interval for every $\alpha \in [0, 1]$;

(iii) Support of A, ${}^{0+}$A, must be bounded

Since α -cuts of any fuzzy number are required to be closed intervals for all $\alpha \in]0, 1],$ every fuzzy number is a convex fuzzy set. The inverse, however, is not necessarily true, since α -cuts of some convex fuzzy sets may be open or half-open intervals.

Special cases of fuzzy numbers include ordinary real numbers and intervals of real numbers. As illustrated in Figure 1.22 (a) is an ordinary real number, (b) is an ordinary (crisp) closed interval [1.25, 1.35], (c) is a fuzzy number expressing the proposition "close to 1.3;" and (d) is a fuzzy number with a flat region (a fuzzy interval).

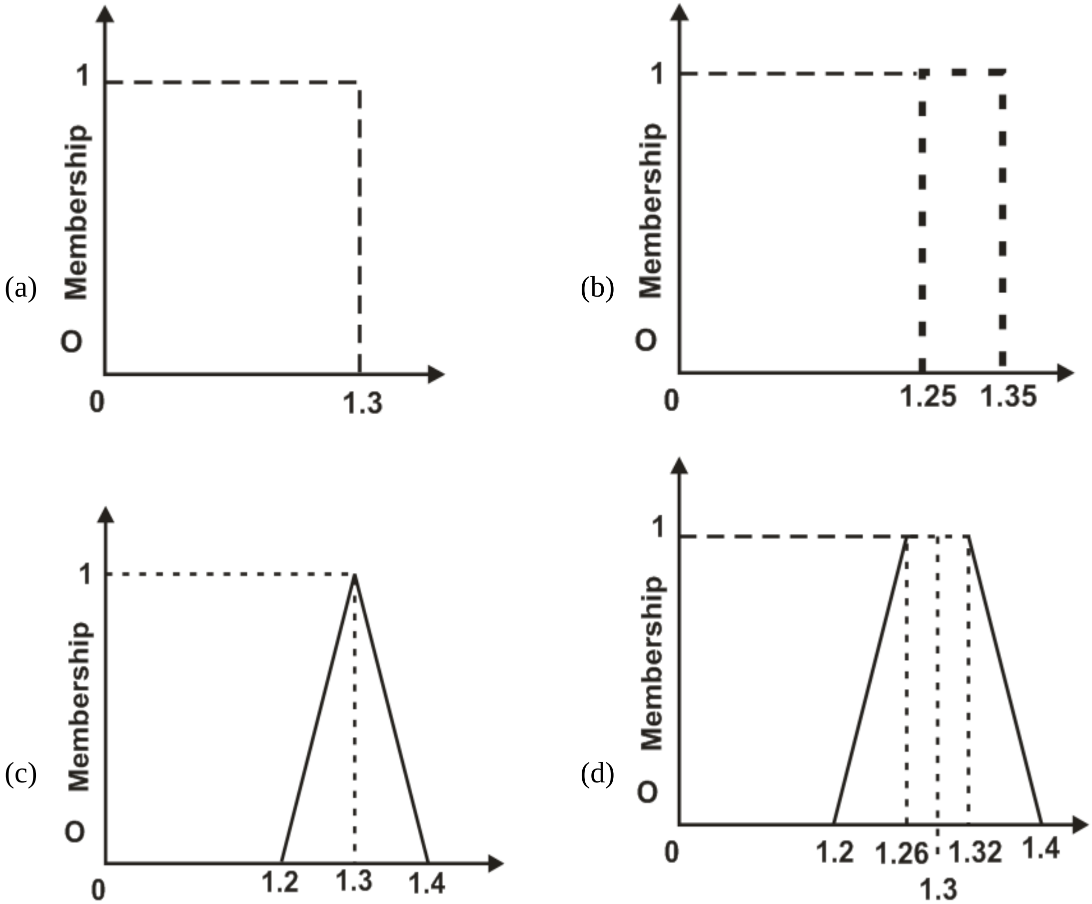
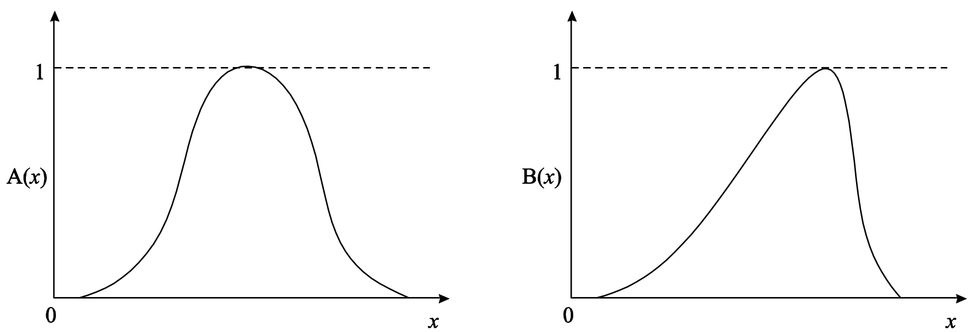


Figure 1.22: A comparison of a real number and a crisp interval with a fuzzy number and a fuzzy interval, respectively



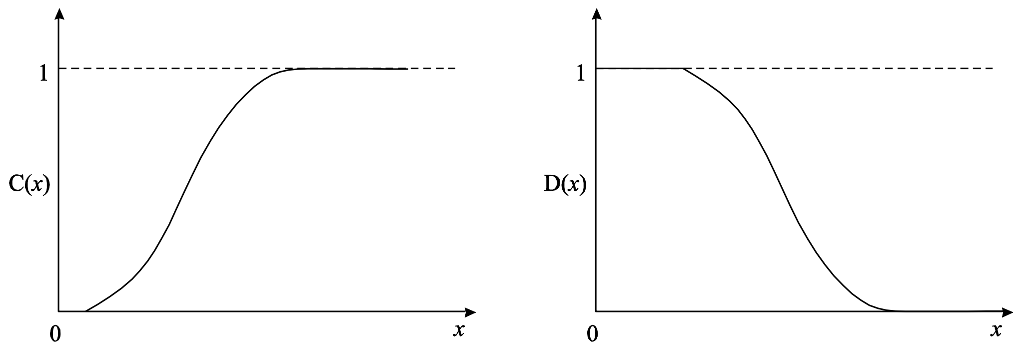


Figure 1.23: Basic types of fuzzy numbers

Although the triangular and trapezoidal shapes of membership functions shown in Figure 1.22 are used most often for representing fuzzy numbers, other shapes may be preferable in some applications. Furthermore, membership functions of fuzzy numbers need not be symmetric as shown in Figure 1.23. Fairly typical are so-called "bell-shaped" membership functions, as exemplified by the functions in Figure 1.22 (a) is symmetric and figure 1.22 (b) is asymmetric. It may also be membership functions which only increase (Figure 1.23 (c)) or only decrease (Figure 1.23 (d)) also qualify as fuzzy numbers. They capture our conception of a *large number* or a *small number* in the context of each particular application.

The following proposition shows that membership functions of fuzzy numbers may be, in general, piecewise-defined functions.

Proposition 1.4.1: Let $A \in F(R)$. Then, A is a fuzzy number if and only if there exists a closed interval $[a, b] \neq \phi$ such that

$$A(x) = \begin{cases} 1, & \text{for} \quad x \in [a, b] \\ l(x), & \text{for} \quad x \in]-\infty, a[\\ r(x), & \text{for} \quad x \in]b, \infty[, \end{cases}$$

where $l(x)$ is a function from $]-\infty, a[$ to [0,1] that is monotonic increasing and continuous from the right such that $l(x) = 0$ for $x \in]-\infty, \omega_1[$; *r* is a function from $]b, \infty[$ to [0,1] that is monotonic decreasing and continuous from the left such that $r(x) = 0$ for $x \in]\omega_2, \infty[$.

The implication of proposition 1.4.1 is that every fuzzy number can be represented in the form of $A(x)$. In general this form allows us to define fuzzy numbers in a piecewise manner, as illustrated in Figure 1.24.

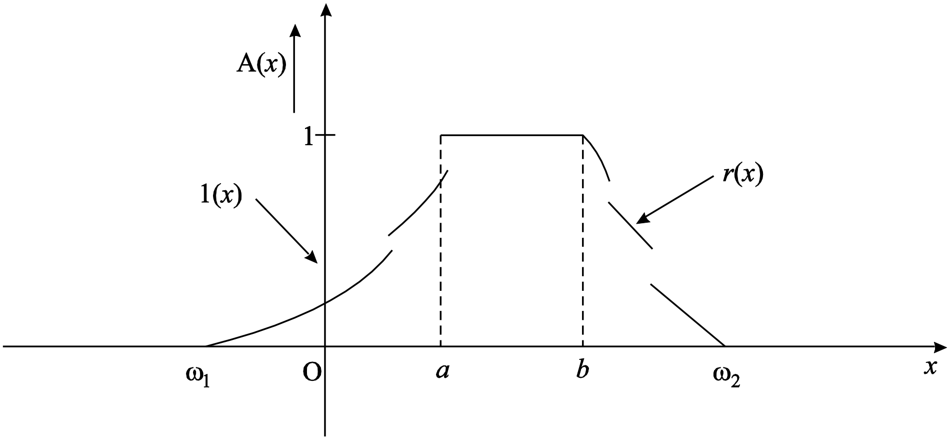


Figure 1.24: General fuzzy number

As an example, let us define the four fuzzy numbers in Figure 1.24 in terms of $A(x)$:

(a) $\omega_1 = a = b = \omega_2 = 1.3, l(x) = 0,$ for all $x \in]-\infty, 1.3[,\ r(x) = 0$ for all $x \in]1.3, \infty[.$

(b) $\omega_1 = a = 1.25,\ b = \omega_2 = 1.35, l(x) = 0,$ for all $x \in]-\infty, 1.25[,\ r(x) = 0,$ for all $x \in]1.35, \infty[.$

(c) $\omega_1 = 1.2,\ a = b = 1.3,\ \omega_2 = 1.4,$

$$l(x) = \begin{cases} 0, & \text{for } x \in]-\infty, 1.2[\\ 10(x - 1.3) + 1, & \text{for } x \in [1.2, 1.3[. \end{cases}$$

$$r(x) = \begin{cases} 10(1.3 - x) + 1, & \text{for } x \in]1.3, 1.4] \\ 0, & \text{for } x \in]1.4, \infty[. \end{cases}$$

(d) $\omega_1 = 1.2,\ a = 1.28,\ b = 1.32,\ \omega_2 = 1.4,$

$$l(x) = \begin{cases} 0, & \text{for } x \in]-\infty, 1.2[\\ 12.5(x - 1.28) + 1, & \text{for } x \in [1.2, 1.28[. \end{cases}$$

$$r(x) = \begin{cases} 12.5(1.32 - x) + 1, & \text{for } x \in]1.32, 1.4] \\ 0, & \text{for } x \in]1.4, \infty[. \end{cases}$$

1.4.1 Triangular Fuzzy Number

It is a fuzzy number represented with three points as follows

$A = (a_1,\ a_2,\ a_3).$

This representation is interpreted as membership function defined below:

$$\mu_{(A)}(x)=\begin{cases}0, & x<a_1\\ \dfrac{x-a_1}{a_2-a_1}, & a_1\le x\le a_2\\ \dfrac{a_3-x}{a_3-a_2}, & a_2\le x\le a_3\\ 0, & x>a_3\end{cases}$$

Now if you get crisp interval by α -cut operation, interval ${}^{\alpha}\mathrm{A}$ shall be obtained as follows $\forall\alpha\in[0,\,1]$

$$\frac{a_1^{(\alpha)}-a_1}{a_2-a_1}=\alpha,\ \frac{a_3-a_3^{(\alpha)}}{a_3-a_2}=\alpha.$$

We get

$$a_1^{(\alpha)}=(a_2-a_1)\alpha+a_1 \ \textit{and}\ \ a_3^{(\alpha)}=-(a_3-a_2)\alpha+a_3.$$

Thus

$${}^{\alpha}A=\left[a_1^{(\alpha)},\ a_3^{(\alpha)}\right]=\left[(a_2-a_1)\alpha+a_1,\,-(a_3-a_2)\alpha+a_3\right].$$

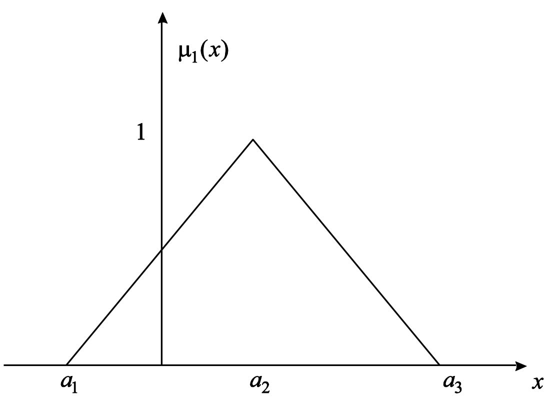


Figure 1.25: Triangular fuzzy number $A=(a_1,\,a_2,\,a_3)$

Example 1.4.1: In the case of the triangular fuzzy number $A=(-5,\,-1,\,1)$ (Figure 1.25), the membership function value will be

$$\mu_{(A)}(x) = \begin{cases} 0, & x < -5 \\ \dfrac{x+5}{4}, & -5 \le x \le -1 \\ \dfrac{1-x}{2}, & -1 \le x \le 1 \\ 0, & x > 1 \end{cases}$$

α - cut interval of this fuzzy number is

$$\frac{x+5}{4} = \alpha \Rightarrow x = 4\alpha - 5$$

$$\frac{1-x}{2} = \alpha \Rightarrow x = -2\alpha + 1$$

$$^{\alpha}A = \left[a_1^{(\alpha)},\ a_3^{(\alpha)}\right] = [4\alpha - 5,\ -2\alpha + 1].$$

If $\alpha = 0.5$, substituting 0.5 for α , we get

$$^{0.5}\mathrm{A} = \left[a_1^{(0.5)},\ a_3^{(0.5)}\right] = [-3,\ 0].$$

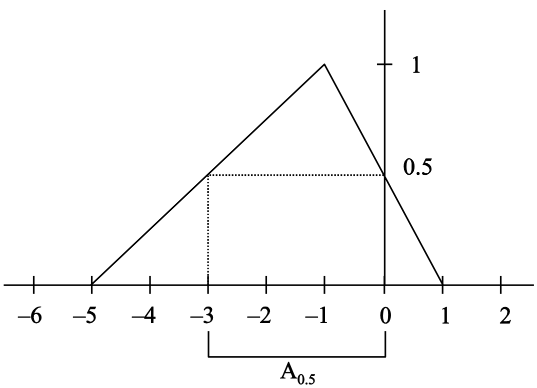


Figure 1.26: $\alpha = 0.5$ cut of triangular fuzzy number $A = (-5, -1, 1)$

1.4.2 Trapezoidal Fuzzy Number

We can define trapezoidal fuzzy number *A* as (Figure 1.27),

$A = (a_1,\ a_2,\ a_3,\ a_4).$

The membership function of this fuzzy number will be interpreted as follows

$$\mu_A(x) = \begin{cases} 0, & x < a_1 \\ \dfrac{x-a_1}{a_2-a_1}, & a_1 \le x \le a_2 \\ 1, & a_2 \le x \le a_3 \\ \dfrac{a_4-x}{a_4}, & a_3 \le x \le a_4 \\ 0, & x > a_4 \end{cases}$$

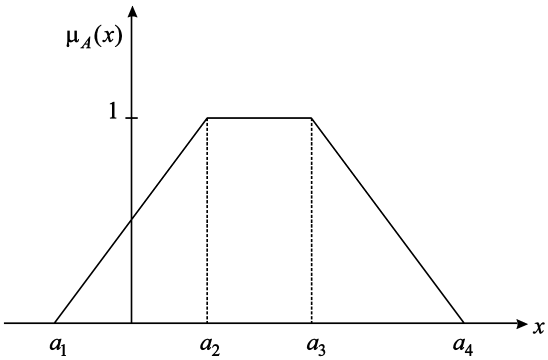


Figure 1.27: Trapezoidal fuzzy number $A = (a_1,\ a_2,\ a_3,\ a_4)$

α -cut interval for this shape is $\forall \alpha \in [0, 1]$

$${}^{\alpha}A = \left[(a_2 - a_1)\alpha + a_1,\ -(a_4 - a_3)\alpha + a_4\right].$$

When $a_2 = a_3$ the trapezoidal fuzzy number coincides with triangular one.

1.4.3 Bell Shape Fuzzy Number

Bell shape fuzzy number is often used in practical applications and its function is defined as follows

$$\mu_f(x) = \exp\left\{\frac{-(x-m_f)^2}{2\delta_f^2}\right\}$$

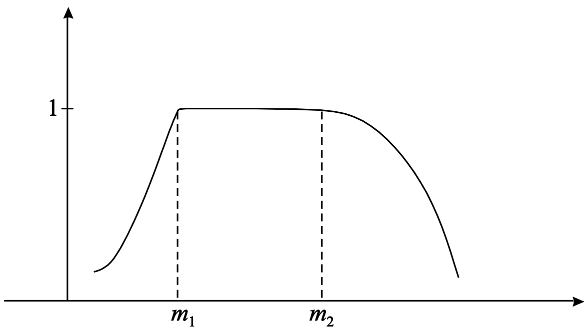


Figure 1.28: Flat fuzzy number

Here $\mu_f(x)$ is the mean of the function and δ_f is the standard deviation.

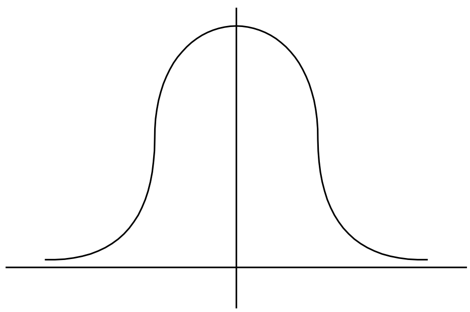


Figure 1.29: Bell shape fuzzy number

1.5 ARITHMETIC OPERATIONS ON INTERVALS AND FUZZY NUMBERS

Fuzzy arithmetic is based on two properties of fuzzy numbers: (1) each fuzzy set, and thus also each fuzzy number, can fully and uniquely be represented by its α -cuts and (2) α -cuts of each fuzzy number are closed intervals of real numbers for all $\alpha \in]0, 1],$ These properties enable us to define arithmetic operations on fuzzy numbers in terms of arithmetic operations on their a-cuts (i.e., arithmetic operations on closed intervals). The latter operations are a subject of *interval analysis*, a well-established area of classical mathematics; we overview them in this section to facilitate our presentation of fuzzy arithmetic in the next section.

1.5.1 Arithmetic Operations on Intervals

Let * denote any of the four arithmetic operations on closed intervals: *addition* (+), *subtraction* (-), *multiplication* (•) , and *division* (/). Then,

$$[a, b]*[c, d]=\{ f * g : a \le f \le b, c \le g \le d\},$$

is a general property of all arithmetic operations on closed intervals, except that $[a, b]/[c, d]$ is not defined when $0 \in [c, d]$, i. e., the resultant of an arithmetic operation on closed intervals is again a closed interval.

The four arithmetic operations on closed intervals are defined as follows:

Addition

$$[a, b]+[c, d]=[a+c, b+d],$$

Subtraction

$$[a, b]-[c, d]=[a-d, b-c],$$

Multiplication

$$[a, b].[c, d]=[\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$$

or $[a, b].[c, d]=[(a.c \wedge a.d \wedge b.c \wedge b.d), (a.c \vee a.d \vee b.c \vee b.d)]$,

and, provided that $0 \notin [c, d]$.

Division

$$[a, b]/[c, d]=[\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]$$

or

$$[a, b]/[c, d]=[(a/c \wedge a/d \wedge b/c \wedge b/d), (a/c \vee a/d \vee b/c \vee b/d)],$$

Inverse Interval

$$[a, b]^{-1}=[\min(1/a, 1/b), \ \max(1/a, 1/b)],$$

or

$$[a, b]^{-1}=[(1/a \wedge 1/b), \ (1/a \vee 1/b)].$$

Example 1.5.1: Let A and B be

$A = [3, 5]$, $B = [-2, 7]$, thus

Addition

$$[3, 5]+[-2, 7]=[3-2, 5+7]=[1, 12],$$

Subtraction

$$[3, 5]+[-2, 7]=[3-7, 5-(-2)]=[-4, 7],$$

Multiplication

$[3, 5]\cdot[-2, 7] = [\min(3\cdot(-2), 3\cdot 7, 5\cdot(-2), 5\cdot 7), \max(3\cdot(-2), 3\cdot 7, 5\cdot(-2), 5\cdot 7)] = [-10, 35],$

Division

$[3, 5]/[-2, 7] = [\min(3/(-2), 3/7, 5/(-2), 5/7), \max(3/(-2), 3/7, 5/(-2), 5/7)] = [-2.5, 5/7],$

Inverse Interval

$$[-2, 7]^{-1} = [\min(1/(-2), 1/7), \max(1/(-2), 1/7)] = [-1/2, 1/7].$$

1.5.2 Operations on α-cut Intervals

We referred to α -cut interval of fuzzy number $A = [a_1, a_2]$ as crisp set

$$^{\alpha}\mathrm{A} = [a_1^{(\alpha)}, a_2^{(\alpha)}], \forall \alpha \in [0, 1], a_1, a_2, a_1^{(\alpha)}, a_2^{(\alpha)} \in \Re$$

So $^{\alpha}\mathrm{A}$ is a crisp interval. As a result, the operations of interval reviewed in the previous section can be applied to the α -cut interval $^{\alpha}\mathrm{A}$.

If α -cut interval $^{\alpha}B$ of fuzzy number $B = [b_1, b_2]$ is given

$$^{\alpha}B = [b_1^{(\alpha)}, b_2^{(\alpha)}], \forall \alpha \in [0, 1], b_1, b_2, b_1^{(\alpha)}, b_2^{(\alpha)} \in \Re,$$

operations between $^{\alpha}\mathrm{A}$ and $^{\alpha}B$ can be described as follows

$$[a_1^{(\alpha)}, a_2^{(\alpha)}](+)[b_1^{(\alpha)}, b_2^{(\alpha)}] = \left[a_1^{(\alpha)} + b_1^{(\alpha)}, a_2^{(\alpha)} + b_2^{(\alpha)}\right], \forall \alpha \in [0, 1],$$

and $$[a_1^{(\alpha)}, a_2^{(\alpha)}](-)[b_1^{(\alpha)}, b_2^{(\alpha)}] = \left[a_1^{(\alpha)} - b_2^{(\alpha)}, a_2^{(\alpha)} - b_1^{(\alpha)}\right], \forall \alpha \in [0, 1].$$

These operations can be also applicable to multiplication and division in the same manner.

1.5.3 Arithmetic Operations on Fuzzy Numbers

Let A and B denote fuzzy numbers and let $*$ denote any of the four basic arithmetic operations. Then, we define $A*B$ a fuzzy set on R by defining its α -cut, $^{\alpha}(\mathrm{A*B})$ as

$$^{\alpha}(\mathrm{A*B}) = {}^{\alpha}\mathrm{A} * {}^{\alpha}\mathrm{B}$$

for any $\alpha \in]0, 1].$ When $*$ is $/,$ it is required that $0 \in {}^{\alpha}\mathrm{B}$ for all $\alpha \in]0, 1].$ Now $A*B$ can be expressed as

$$A*B = \bigcup_{\alpha \in]0, 1]} {}^{\alpha}(A*B).$$

Since ${}^{\alpha}\left(A*B\right)$ is closed interval for each $\alpha\in]0,1]$ and A, B are fuzzy numbers, $A*B$ is also a fuzzy number.

For example consider two triangular-shape fuzzy numbers A and B defined as follows:

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

$$B(x)=\begin{cases}0, & \text{for } x\le 1 \text{ and } x>3\\ (x-1)/2, & \text{for } 1<x\le 3\\ (5-x)/2, & \text{for } 3<x\le 5.\end{cases}$$

Their α -cuts are:

${}^{\alpha}\mathrm{A}=[2\alpha-1,\,3-2\alpha]$ and ${}^{\alpha}\mathrm{B}=[2\alpha+1,\,5-2\alpha]$.

Using arithmetic operation, we obtain

$${}^{\alpha}(A+B)=[4\alpha,\,8-4\alpha] \text{ for } \alpha\in]0,1],$$

$${}^{\alpha}(A-B)=[4\alpha-6,\,2-4\alpha] \text{ for } \alpha\in]0,1],$$

$${}^{\alpha}(A\cdot B)=\begin{cases}[-4\alpha^2+12\alpha-5,\,4\alpha^2-16\alpha+15] & \text{for } \alpha\in]0,0.5]\\ [4\alpha^2-1,\,4\alpha^2-16\alpha+15] & \text{for } \alpha\in]0.5,1],\end{cases}$$

$${}^{\alpha}(A/B)=\begin{cases}\left[\left(2\alpha-1\right)/\left(2\alpha+1\right),\,\left(3-2\alpha\right)/\left(2\alpha+1\right)\right] & \text{for } \alpha\in]0,0.5]\\ \left[\left(2\alpha-1\right)/\left(5-2\alpha\right),\,\left(3-2\alpha\right)/\left(2\alpha+1\right)\right] & \text{for } \alpha\in]0.5,1].\end{cases}$$

The resulting fuzzy numbers are then:

$$\left(A+B\right)(x)=\begin{cases}0, & \text{for } x\le 0 \text{ and } x>8\\ x/4, & \text{for } 0<x\le 4\\ (8-x)/4, & \text{for } 4<x\le 8,\end{cases}$$

$$\left(A-B\right)(x)=\begin{cases}0, & \text{for } x\le -6 \text{ and } x>2\\ (x+6)/4, & \text{for } -6<x\le -2\\ (2-x)/4, & \text{for } -2<x\le 2,\end{cases}$$

$$(A \cdot B)(x) = \begin{cases} 0, & \text{for } x < -5 \text{ and } x \geq 15 \\ \left[3 - (4-x)^{1/2}\right]/4, & \text{for } -5 \leq x < 0 \\ (1+x)^{1/2}/2, & \text{for } 0 \leq x < 3 \\ \left[4 - (1+x)^{1/2}\right]/2, & \text{for } 3 \leq x < 15, \end{cases}$$

$$(A / B)(x) = \begin{cases} 0, & \text{for } x < -1 \text{ and } x \geq 3 \\ (x+1)/(2-2x), & \text{for } -1 \leq x < 0 \\ (5x+1)/(2x+2), & \text{for } 0 \leq x < 1/3 \\ (3-x)/(2x+2), & \text{for } 1/3 \leq x < 3. \end{cases}$$

We now proceed to the second method for developing fuzzy arithmetic, which is based on the extension principle. Employing this principle, standard arithmetic operations on real numbers are extended to fuzzy numbers.

Let A and B denote fuzzy numbers and let $*$ denote any of the four basic arithmetic operations. Then, we define a fuzzy set $A * B$ on R by the equation

$$(A * B)(z) = \sup_{z = x * y} \left[A(x) \wedge B(y)\right] \text{ for all } z \in R.$$

More specifically, we define for all $z \in R$ as given below:

$$(A + B)(z) = \sup_{z = x + y} \left[A(x) \wedge B(y)\right],$$

$$(A - B)(z) = \sup_{z = x - y} \left[A(x) \wedge B(y)\right],$$

$$(A \cdot B)(z) = \sup_{z = x \cdot y} \left[A(x) \wedge B(y)\right],$$

$$(A / B)(z) = \sup_{z = x / y} \left[A(x) \wedge B(y)\right].$$

Example 1.5.2: Let A and B be such that

$$A(x) = [-5, 1] \text{ and } B(y) = [-5, 12]$$

with associate membership functions shown in Fig. 1.30.

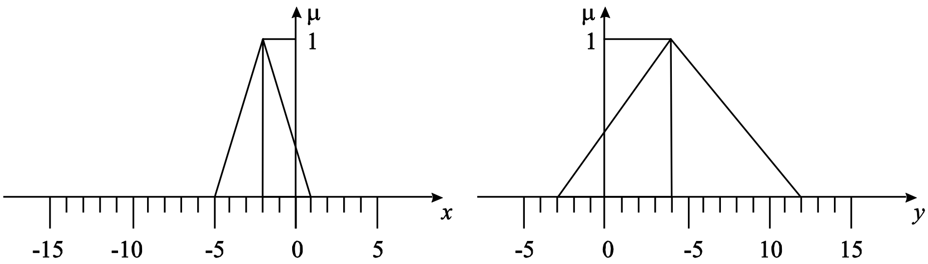


Figure 1.30: Two membership functions

$$\mu_A(x) = \begin{cases} \dfrac{x}{3} + \dfrac{5}{3}, & \text{for } -5 \le x \le -2 \\ \dfrac{-x}{3} + \dfrac{1}{3}, & \text{for } -2 \le x \le 1, \end{cases}$$

$$\mu_B(y) = \begin{cases} 0, & \text{for } -5 \le x \le -3 \\ \dfrac{y}{7} + \dfrac{3}{7}, & \text{for } -3 \le x \le 4 \\ \dfrac{-y}{8} + \dfrac{12}{8}, & \text{for } 4 \le x \le 12. \end{cases}$$

Then, by the general operation rule we have

$$C(z) = A(x) + B(y) = [-5, 1] + [-5, 12] = [-10, 13]$$

and, by comparing $\mu_A(x)$ and $\mu_B(x)$ point wise, we obtain

$$\mu_C(z) = \sup_{z=x+y} \{\mu_A(x), \ \mu_B(y)\}$$

$$\mu_C(z) = \begin{cases} 0, & \text{for } -10 \le z \le -8 \\ \dfrac{z}{10} + \dfrac{8}{10}, & \text{for } -8 \le z \le 2 \\ \dfrac{-z}{11} + \dfrac{13}{11}, & \text{for } \ 2 \le z \le 13. \end{cases}$$

Here, it is clear that the general "sup" rule does not yield the explicit formulas easily. In contrast, this resulting explicit formula of $\mu_C(z)$ can be easily obtained by using the equivalent α -cut operation as follows.

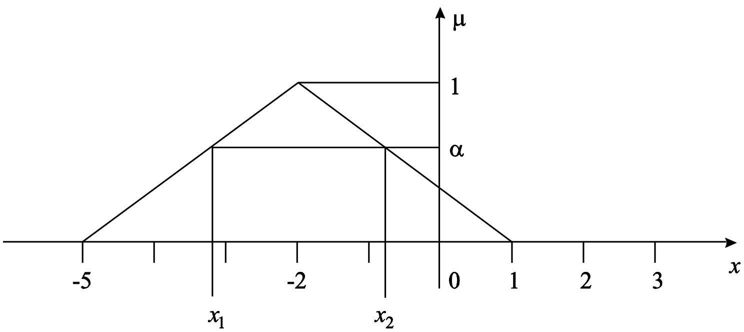


Figure 1.31: Membership function

In the α-cut notation, for any α value, the α-cut of $A(x)$ is obtained by letting α = $x/3 + 5/3$ and α = $-x/3 + 1/3$, respectively, which give $x_1 = 3\alpha - 5$ and $x_2 = -3\alpha + 1$, as shown in (the enlarged) Fig. 1.31. Hence, the projection interval is

$$[A(x)]_\alpha = [x_1,\ x_2] = [3\alpha - 5,\ -3\alpha + 1].$$

Similarly, $[B(y)]_\alpha = [x_1,\ x_2] = [7\alpha - 3,\ -8\alpha + 12],$

so that

$$[C(z)]_\alpha = [A(x)]_\alpha + [B(y)]_\alpha = [10\alpha - 8,\ -11\alpha + 13].$$

Setting $z_1 = 10\alpha - 8$ and $z_2 = -11\alpha + 13$ gives $\alpha = z_1/10 + 8/10$ and $\alpha = -z_2/11 + 12/11$, which yield the membership function shown in Fig. 1.32.

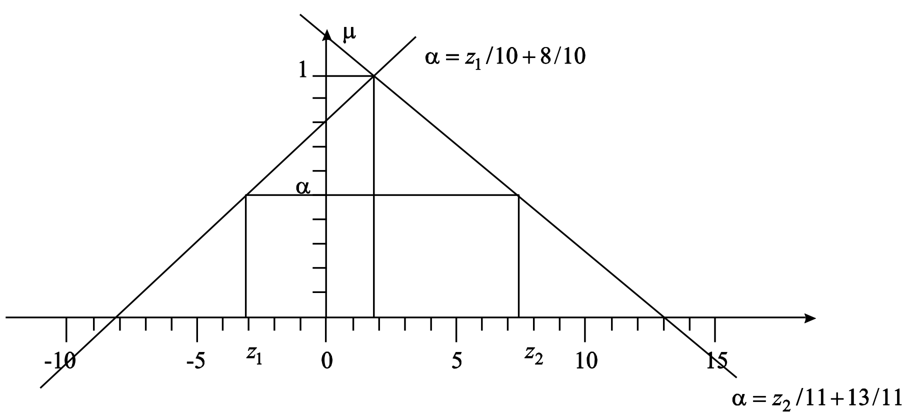


Figure 1.32: The resulting membership function

Taking $C(z) = [-10, 13]$ into account, we finally arrive at

$$\mu_C(z) = \begin{cases} 0, & \text{for } -10 \le z \le -8 \\ \dfrac{z}{10} + \dfrac{8}{10}, & \text{for } -8 \le z \le 2 \\ \dfrac{-z}{11} + \dfrac{13}{11}, & \text{for } 2 \le z \le 13. \end{cases}$$

Example 1.5.3: Let $A(x)$ and $B(y)$ be such that

$$A(x) = [0,\ 20] \text{ and } B(y) = [0,\ 10],$$

With the membership functions

$$\mu_A(x) = \begin{cases} 0, & \text{for } 0 \le x \le 7 \\ \dfrac{y}{7} + \dfrac{3}{7}, & \text{for } 7 \le x \le 14 \\ \dfrac{-y}{8} + \dfrac{12}{8}, & \text{for } 14 \le x \le 19 \\ 0, & \text{for } 19 \le x \le 20, \end{cases}$$

and

$$\mu_B(y) = \begin{cases} 0, & \text{for } 0 \le x \le -3 \\ \dfrac{y}{2} - \dfrac{3}{2}, & \text{for } 3 \le x \le 5 \\ \dfrac{-y}{5} + 2, & \text{for } 5 \le x \le 10. \end{cases}$$

Then we have, via the interval arithmetic

$$C(z) = A(x) - B(y) = [-10,\ 20], \text{ with}$$

$$\mu_C(z) = \begin{cases} 0, & \text{for } -10 \le x \le -3 \\ \dfrac{z}{12} + \dfrac{3}{12}, & \text{for } -3 \le x \le 9 \\ \dfrac{-z}{7} + \dfrac{16}{7}, & \text{for } 9 \le x \le 16 \\ 0, & \text{for } 16 \le x \le 20. \end{cases}$$

Example 1.5.4: Let $A(x)$ and $B(y)$ be such that

$$A(x) = [2,\ 5] \text{ and } B(y) = [3,\ 6],$$

With the membership functions

$$\mu_A(x) = \begin{cases} x-2, & \text{for } 2 \le x \le 3 \\ \dfrac{-x}{2} + \dfrac{5}{2}, & \text{for } 3 \le x \le 5, \end{cases}$$

$$\mu_B(y) = \begin{cases} \dfrac{y}{2} - \dfrac{3}{2}, & \text{for } 3 \le x \le 5 \\ -y+6, & \text{for } 5 \le x \le 6, \end{cases}$$

as shown in figure 1.33.

In the α -cut notation, for any α value, letting

$$\left[A(x)\right]_\alpha = \left[\alpha+2,\ -2\alpha+5\right] \text{ and } \left[B(y)\right]_\alpha = \left[2\alpha+3,\ -\alpha+6\right].$$

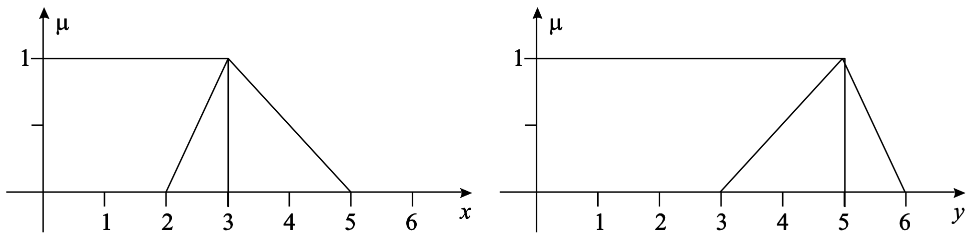


Figure 1.33: Two membership functions

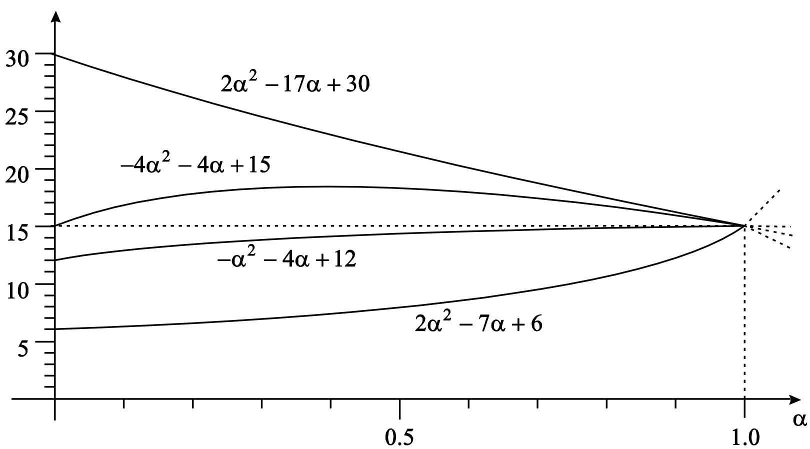


Figure 1.34: The intermediate membership functions

It then follows that

$C(z) = [6,\ 30]$ and

$$[C(z)]_\alpha = [A(x)]_\alpha \cdot [B(y)]_\alpha = [\alpha+2,\ -2\alpha+5]\cdot[2\alpha+3,\ -\alpha+6] = [\underline{p}(\alpha),\ \overline{p}(\alpha)]$$

where

$$\underline{p}(\alpha) = \min\{2\alpha^2+7\alpha+6, -\alpha^2+4\alpha+12, -4\alpha^2+4\alpha+15, 2\alpha^2-17\alpha+30\},$$

$$\overline{p}(\alpha) = \max\{2\alpha^2+7\alpha+6, -\alpha^2+4\alpha+12, -4\alpha^2+4\alpha+15, 2\alpha^2-17\alpha+30\},$$

with the curves shown in figure 1.34.

Hence $\underline{p}(\alpha) = 2\alpha^2+7\alpha+6$ and $\overline{p}(\alpha) = 2\alpha^2-17\alpha+30$ so that

$$C(z) = [\underline{p}(\alpha),\ \overline{p}(\alpha)] = [2\alpha^2+7\alpha+6, 2\alpha^2-17\alpha+30].$$

Let, moreover

$$z_1 = 2\alpha^2+7\alpha+6,\ z_2 = 2\alpha^2-17\alpha+30.$$

We solve them for α, subject to $0 \le \alpha \le 1$, and obtain

$$\alpha = \frac{-7+\sqrt{1+8z_1}}{4} \quad \text{or} \quad \alpha = \frac{17+\sqrt{49+8z_2}}{4}.$$

And consequently

$$\mu_C(z) = \begin{cases} \dfrac{-7+\sqrt{1+8z}}{4}, & \text{for } 6 \le z \le 15 \\ \dfrac{17+\sqrt{49+8z}}{4}, & \text{for } 15 \le z \le 30, \end{cases}$$

As shown in figure 1.35.

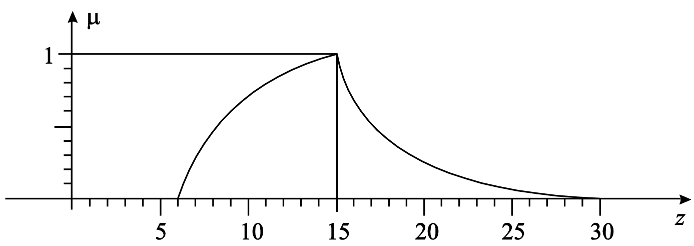


Figure 1.35: The resulting membership function

Example 1.6.5: Let $A(x)$ and $B(y)$ be such that

$A(x) = [18, 33]$ and $B(y) = [5, 8]$,

With the membership functions

$$\mu_A(x) = \begin{cases} \frac{x}{4} - \frac{18}{4}, \text{ for } 18 \le x \le 22 \\ \frac{-x}{11} + 3, \text{ for } 22 \le x \le 33, \end{cases}$$

$$\mu_B(y) = \begin{cases} y - 5, & \text{for } 5 \le x \le 6 \\ -\frac{y}{2} + 4, & \text{for } 6 \le x \le 8, \end{cases}$$

as shown in figure 1.36.

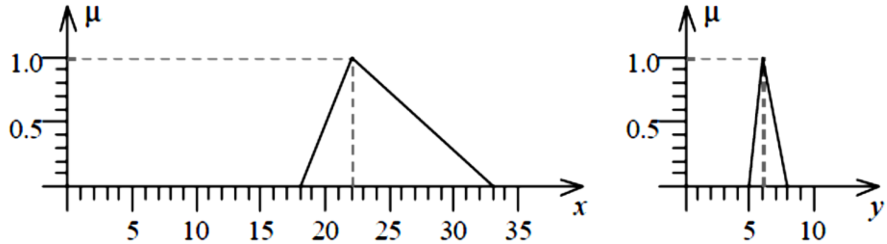


Figure 1.36: Two membership functions

In the α -cut notation, for any α value, letting

$[A(x)]_\alpha = [4\alpha + 18, -11\alpha + 33]$ and $[B(y)]_\alpha = [\alpha + 5, -2\alpha + 8]$.

$$[C(z)]_\alpha = \frac{[A(x)]_\alpha}{[B(y)]_\alpha} = \frac{[4\alpha + 18, -11\alpha + 33]}{[\alpha + 5, -2\alpha + 8]} = \left[\frac{4\alpha + 18}{-2\alpha + 8}, \frac{-11\alpha + 33}{\alpha + 5}\right].$$

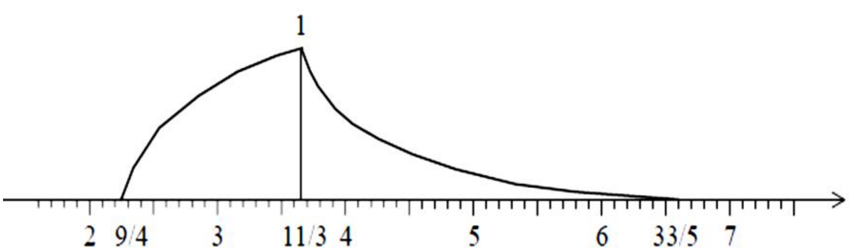


Figure 1.37: The resulting membership functions

Let $z_1 = \dfrac{4\alpha + 18}{-2\alpha + 8}$, $z_2 = \dfrac{-11\alpha + 33}{\alpha + 5}$ gives

$\alpha = \dfrac{8z_1 - 18}{2z_1 + 4}$ and $\alpha = \dfrac{-5z_2 + 33}{z_2 + 11}$, such that

$$\mu_C(z) = \begin{cases} \dfrac{8z - 18}{2z + 4}, & \text{for } \dfrac{9}{4} \le z \le \dfrac{11}{3} \\ \dfrac{-5z + 33}{z + 11}, & \text{for } \dfrac{11}{3} \le z \le \dfrac{11}{3}, \end{cases}$$

As shown in figure 1.37

Exercise - 1

1. Let A be the given fuzzy set with membership function

$$A(x)=\begin{cases} x-5, & for\ 5\le x\le 6 \\ -x+7, & for\ 6<x\le 7 \\ 0, & \text{otherwise,} \end{cases}$$

Sketch the graph of the function. What is its type?

2. Consider two fuzzy sets A and B as shown

$$A=\left\{\frac{0}{1}+\frac{0.5}{2}+\frac{0.3}{3}+\frac{0.7}{4}+\frac{0.9}{5}\right\},\ B=\left\{\frac{0.2}{1}+\frac{0.4}{2}+\frac{0.6}{3}+\frac{0.9}{4}+\frac{0.4}{5}\right\}.$$

Find $(a)\ A\cup B,\ (b)\ A\cap B, (c)\ \overline{A\cup B}, (d)\ \overline{A\cap B}.$

3. Consider the fuzzy sets *A*, *B*, and *C* defined on the interval *X* = [0,10] of real numbers by the membership grade functions

$$A(x)=\frac{x}{x+2},\ B(x)=2^{-x},\ C(x)=\frac{1}{1+10(x-2)^2}.$$

Determine mathematical formulas and graphs of the membership grade functions of each of the following sets:

(a) $\overline{A},\ \overline{B},\ \overline{C};$

(b) $A\cup B, A\cup C,\ B\cup C;$

(c) $A\cap B, A\cap C,\ B\cap C;$

(d) $A\cup B\cup C,\ A\cap B\cap C;$

(e) $\overline{A\cup C}, A\cap\overline{C},\ \overline{B\cup C}.$

4. Let A be a fuzzy set defined by

$$A(x)=\frac{0.5}{x_1}+\frac{0.4}{x_2}+\frac{0.7}{x_3}+\frac{0.8}{x_4}+\frac{1}{x_5}.$$

List all α -cuts and strong α -cuts of *A*.

5. Discuss the reflexivity properties of the following fuzzy relation.

R	Xl	X2	X3
X_1	1.0	0.7	0.3
X_2	0.4	0.5	0.8
X_3	0.7	0.5	1.0

6. For the fuzzy relation

$$R = \begin{bmatrix} 1.0 & 0.4 & 0.8 & 0.3 & 0.0 \\ 0.5 & 1.0 & 0.6 & 0.7 & 1.0 \\ 0.9 & 1.0 & 0.0 & 0.6 & 0.8 \\ 1.0 & 0.5 & 0.2 & 0.0 & 0.9 \\ 0.3 & 0.5 & 0.3 & 0.1 & 1.0 \end{bmatrix}$$

Compute its projections and cylindrical extensions.

7. Show the two fuzzy sets satisfy the De Morgan's Law

$$A(x) = \frac{1}{1+(x-10)}, \quad B(x) = \frac{1}{1+x^2}.$$

8. Define equilibrium point. Show that every fuzzy complement has at most one equilibrium.

9. Compute the complements, intersection and union of the following fuzzy relations *R* and *S*.

$$R = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1.0 & 0.2 & 0.4 & 0.0 \\ 0.0 & 0.1 & 0.0 & 0.9 \\ 0.1 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.4 & 0.0 & 1.0 \end{bmatrix} \end{array} \text{ and } S = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1.0 & 0.0 & 0.0 & 0.4 \\ 0.0 & 0.0 & 0.4 & 0.9 \\ 0.4 & 0.0 & 0.1 & 0.0 \\ 0.5 & 1.0 & 0.0 & 0.0 \end{bmatrix} \end{array}$$

10. Determine whether the following fuzzy relation is an equivalence relation.

$$\begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 1.0 & 0.8 & 0.4 & 0.1 \\ 0.8 & 1.0 & 0.0 & 0.0 \\ 0.4 & 0.0 & 1.0 & 0.5 \\ 0.1 & 0.0 & 0.5 & 1.0 \end{bmatrix} \end{array}$$

11. Discuss the properties of the following morphism *h*.

$h: a, b \to \alpha;\ c \to \beta;\ d, e \to \gamma$, where

$A = \{a, b, c, d, e\}, B = \{\alpha, \beta, \gamma\}, R \subseteq A \times A, S \subseteq B \times B.$

$$R = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \!\!\begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1.0 & 0.0 & 0.0 & 1.0 & 0.1 \\ 1.0 & 0.4 & 0.0 & 0.0 & 1.0 \\ 0.4 & 0.0 & 0.0 & 0.9 & 0.0 \\ 0.8 & 0.0 & 0.6 & 0.8 & 0.0 \\ 0.0 & 0.6 & 0.8 & 0.4 & 1.0 \end{bmatrix} \end{array} \quad \text{and} \quad S = \begin{array}{c} \\ \alpha \\ \beta \\ \gamma \end{array} \!\!\begin{array}{c} \begin{array}{ccc} \alpha & \beta & \gamma \end{array} \\ \begin{bmatrix} 1.0 & 0.0 & 1.0 \\ 0.4 & 0.0 & 0.9 \\ 0.8 & 0.8 & 1.0 \end{bmatrix} \end{array}$$

12. For all $a, b \in [0,1]$, prove that $i_{\min}(a, b) \le i(a, b) \le \min(a, b)$,

where $i_{\min}$ denotes the drastic intersection.

13. For all $a, b \in [0,1]$, show that $\max(a, b) \le u(a, b) \le u_{\max}(a, b)$,

where $u_{\max}$ denotes the drastic union.

14. Let A, *B* be two fuzzy numbers whose membership functions are given by

$$A(x) = \begin{cases} (x+2)/2, & for\ -2 < x \le 0 \\ (2-x)/2, & for\ \ 0 < x < 2 \\ 0, & \text{otherwise,} \end{cases} \quad \text{and} \quad A(x) = \begin{cases} (x-2)/2, & for\ 2 < x \le 4 \\ (2-x)/2, & for\ \ 0 < x \le 6 \\ 0, & \text{otherwise.} \end{cases}$$

Calculate the fuzzy numbers $A+B,\ A-B,\ B-A,\ A \cdot B,\ A/B$.

15. Let $S_x = [1, 5]$ and $S_y = [2, 6]$ with fuzzy membership functions

$$\mu_{S_x} = \begin{cases} (x-1), & for\ 1 \le x \le 2 \\ (5-x)/3, & for\ \ 2 \le x \le 5, \end{cases} \quad \text{and} \quad \mu_{S_y} = \begin{cases} (y-2)/2, & for\ 2 \le x \le 4 \\ (6-y)/2, & for\ \ 4 \le x \le 6, \end{cases}$$

Compute the following six operations to obtain S_z and μ_{S_z} :

$$z = F(x, y) = x * y \ \text{ for } \ * \in \{+, -, \cdot, /\}.$$

BIBLIOGRAPHY

1. Kosko, B., Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.
2. Zadeh, L. A. (1965), Fuzzy sets, Inform. Contr., vol. 8, pp. 338-353.
3. Lee, K. H., First Course on Fuzzy Theory and Applications, Springer, Berlin Heidelberg, NewYork, 2005.
4. Klir, G. J. and Yuan, B., Fuzzy Sets and Fuzzy Logic-Theory and Applications, Prentice Hall, New Jersey, 1995.

Chapter-2

Fuzzy Relations and Fuzzy Logic

2.1 INTRODUCTION

In the first chapter we have defined fuzzy set and its properties, operations on fuzzy sets and different fuzzy numbers. In the present chapter important topics of fuzzy set theory like Membership function; fuzzification and membership values assignments are discussed. Fuzzy relations and their graphs are studied in details. However, our main emphasis is on fuzzy logic which is a method to formalize the human capacity of imprecise reasoning or approximate reasoning. Such reasoning represents the human ability to reason approximately and judge ambiguous uncertainty. In fuzzy logic all truths are partial or approximate. In this sense this reasoning has also been termed interpolative reasoning and the process of interpolating between the binary extremes of true and false is represented by the ability of fuzzy logic to encapsulate partial truths. In later part of the chapter the concepts of classical and fuzzy logic and approximating reasoning are explained in detail.

2.2. FEATURES OF MEMBERSHIP FUNCTION

The feature of the membership function is defined by three properties:

(i) Core

(ii) Support

(iii) Boundary

The Fig. 2.1 shown below defines the properties listed above.

(i) Core

If the region of universe is characterized by full membership 1 in the set *A* then this gives the core of the membership function of fuzzy at *A*. The elements which have the membership function as 1 are the elements of the core, i.e., $\mu_A(x) = 1.$

(ii) Support

If the region of universe is characterized by nonzero membership in the set *A*, this defines the support of a membership function for fuzzy set *A*.

The support has the elements whose membership is greater than zero, *i.e.*, $\mu_A(x) > 0.$

(iii) Boundary

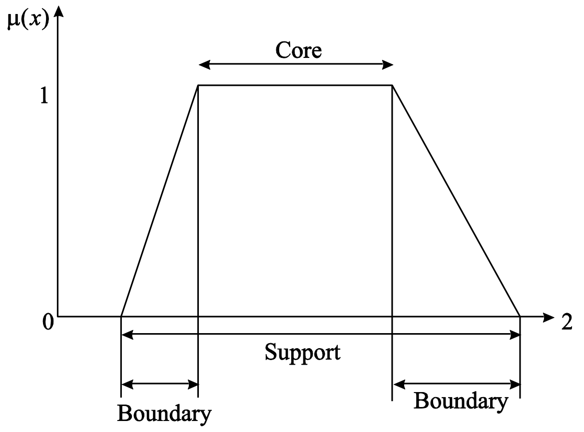


Figure 2.1: Features of membership function

If the region of universe has a nonzero membership but not full membership, this defines the boundary of a membership function; this defines the boundary of a membership function for fuzzy set *A*.

The boundaries have the elements whose memberships are between 0 and 1, *i.e.*, $0 < \mu_A(x) < 1$.

These are the standard regions defined in the membership functions. Next, we define two important terms.

Crossover point

The crossover point of a membership function is the element in universe whose membership value equal to 0.5, *i.e.*, $\mu_A(x) = 0.5$.

Height

The height of the fuzzy set *A* is the maximum value of the membership function, *i. e.*, $\max\left(\mu_A(x)\right)$.

The membership functions can be symmetrical or asymmetrical.

2.3 FUZZIFICATION

Fuzzification is an important concept in the fuzzy logic theory. Fuzzification is the process where the crisp quantities are converted to fuzzy (crisp to fuzzy). By identifying some of the uncertainties present in the crisp values, we form the fuzzy values. The conversion of fuzzy values is represented by the membership functions.

In any practical applications in industries, measurement of voltage, current, temperature, etc., there might be a negligible error. This causes imprecision in the data. This imprecision can be represented by the membership functions. Thus fuzzification is performed.

Hence, fuzzification process may involve assigning membership values for the given crisp quantities.

Membership values Assignments

There are various methods to assign the membership values or the membership functions to fuzzy variables. The assignment can be just done by intuition or by using some algorithms or logical procedures. The methods for assigning the membership values are listed as follows:

(i) Intuition

(ii) Inference

(iii) Rank ordering

(iv) Angular fuzzy sets

(v) Neural networks

(vi) Genetic algorithms

(vii) Inductive seasoning

(i) Intuition

Intuition is based on the human's own intelligence and understanding to develop the membership functions. The thorough knowledge of the problem and the linguistic variable should also be known. Fig. 2.2 (a) shows membership function for imprecision in crisp temperature reading.

For example, consider the speed of a dc-motor. The shape of the universe of speed given in rpm is shown in Fig. 2.2 (b).

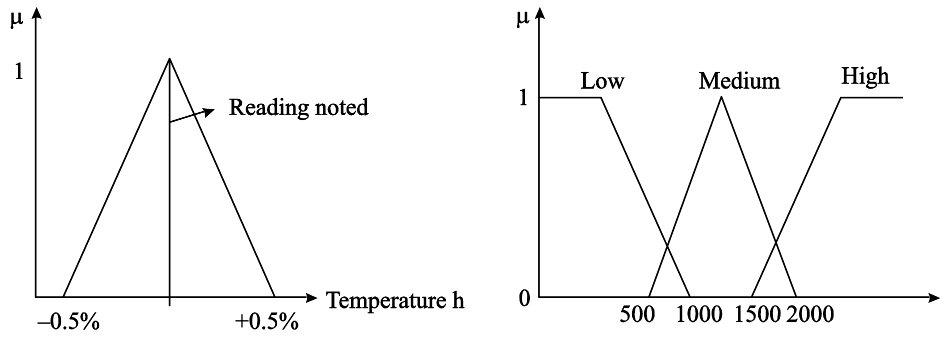


Figure 2.2: Membership function (a) temperature (b) speed

The curves represent membership function corresponding to various fuzzy variables. The range of speed is splitted into low, medium and high. The curves differentiate the ranges set by humans. The placement of curves is approximate over the universe of discourse; the number of curves and the overlapping of curves is an important criteria to be considered while defining membership functions.

(ii) Inference

This method involves the knowledge to perform deductive reasoning. The membership function is formed from the known facts or knowledge set.

Let us use inference method for the identification of the triangle. Let U be universe of triangles and A,B, and C be the inner angles of the triangles. Also $A \geq B \geq C \geq 0$. Therefore the universe is given by:

$$U = \{(A,\ B,\ C),\ A \geq B \geq C \geq 0,\ A + B + C = 180\}.$$

There are various types of triangles, for identifying, we define three types of triangles:

I- Appropriate isosceles triangle, R- Appropriate right triangle and O- Other triangles. The membership vales can be inferred to all of these triangle types through the method of inference, as we know the knowledge about the geometry of the triangles.

The membership for the approximate isosceles triangle, for the given conditions $A \geq B \geq C \geq 0$ and $A + B + C = 180$, is given as,

$$\mu_I(A,B,C) = 1 - \frac{1}{60^\circ}\min(A-B, B-C).$$

The membership for the appropriate right triangle, for the same conditions, is

$$\mu_R(A,B,C) = 1 - \frac{1}{90^\circ}(A - 90^\circ).$$

The membership for the other triangles can be given as the complement of the logical union of the two already defined membership functions

$$\mu_O(A,B,C) = \overline{I \cup R} \text{ or by using De Morgan's law, it is}$$

$$\mu_O(A,B,C) = I \cap R = \min\begin{Bmatrix}1-\mu_I(A,B,C)\\1-\mu_R(A,B,C)\end{Bmatrix}.$$

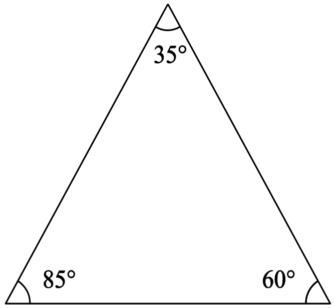


Figure 2.3 : A given triangle

Example 2.3.1: Define the triangle for the figure shown in Fig. 2.3 with the three given angles.

Solution. The condition is

$$A \geq B \geq C \geq 0 \text{ and } A+B+C=180^\circ.$$

Here $U = \{A=85^\circ \geq B=60^\circ \geq C=35^\circ \geq 0,\ A+B+C=180\}$.

The membership values for the triangle shown in Fig. 2.3 for each type of triangles are:

$$\mu_I(x) = 1 - \frac{1}{60^\circ}\min(A-B, B-C).$$

$$\mu_I(x) = 1 - \frac{1}{60^\circ}\min(85^\circ - 60^\circ, 60^\circ - 35^\circ) = 1 - \frac{1}{60^\circ}\min(25^\circ, 25^\circ)$$

$$\mu_I(x) = 0.583.$$

Now, $\mu_R(x) = 1 - \frac{1}{90^\circ}(A - 90^\circ).$

$$\mu_R(x) = 1 - \frac{1}{90^\circ}(85^\circ - 90^\circ) = 0.944.$$

And $\mu_O(x) = \min x\{1-\mu_I(x), 1-\mu_R(x)\} = \min\{1-0.583, 1-0.944\}.$

$$\mu_O(x) = 0.055.$$

Hence there is highest membership for $\mu_R(x)$. Thus inference method can be used to calculate the membership values.

(iii) Rank ordering

The polling concept is used to assign membership values by rank ordering process. Preferences are above for pair wise comparisons and from this the ordering of the membership is done.

Example 2.2: Suppose 1000 people respond to a questionnaire about the pair wise+ preference among five cars of universe set X={Palio, Siena, Astra, Easter, Baleno}. Define a fuzzy set as A on the universe of cars, "best cars".

Table 2.1: Pair wise preferences among five cars between 1000 people

	Number who preferred							
	Palio	Siena	Astra	Easter	Baleno	Total	Percentage	Rank order
Palio	-	515	545	523	671	2.254	22.5	2
Siena	481	-	475	845	580	2.381	23.8	1
Astra	469	624	-	141	536	1.770	17.7	4
Easter	457	530	470	-	649	2.114	21.1	3
Baleno	265	425	402	389	-	1.481	14.8	5
Total						10,000		

The pair wise comparison is made among 1,000 people and their views are summarized in Table 2.1.

From the table, it is clear that 515 preferred Siena compared to Palio, 545 Astra to Palio, etc. The table forms an anti symmetric matrix. There are about ten comparisons made which gives a ground total of 10,000. Based on preferences, the percentage is calculated. The ordering is then performed. It is found that Siena is selected as the best car.

Fig. 2.4 shows the membership function for this example.

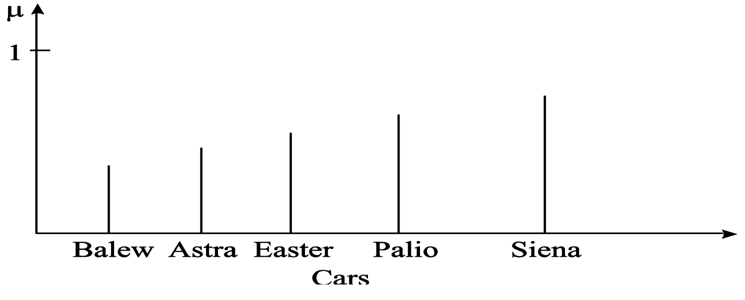


Figure 2.4: Membership for best car

(iv) Angular Fuzzy Set

The angular fuzzy sets are different from the standard fuzzy sets in their coordinate description. These sets are defined on the universe of angles, hence are repeating shapes after every 2π cycle. Angular fuzzy sets are applied in quantitative description of linguistic variables known truth-values. When membership of value 1 is true and that of 0 is false, then in between '0' and '1' is partially true or partially false.

The linguistic values are formed to vary with θ, the angle defined on the unit circle and their membership values are on $\mu_t(\theta)$ The membership of this linguistic term can be obtained from $\mu_t(\theta) = t \tan\theta$, where t is the horizontal projection of the radial vector and is given as cos θ, i.e., $t = \cos\theta$. When the coordinates are in polar form, angular fuzzy sets can be used.

Example 2.3.2: Consider a motor which is used in computer peripheral applications. From the membership function based on its rotation using angular fuzzy sets

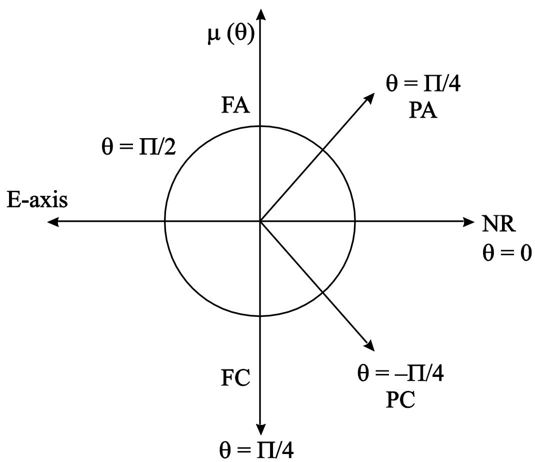


Figure 2.5: Angular fuzzy set

The linguistic terms relating to the direction of motion of the motor is given as

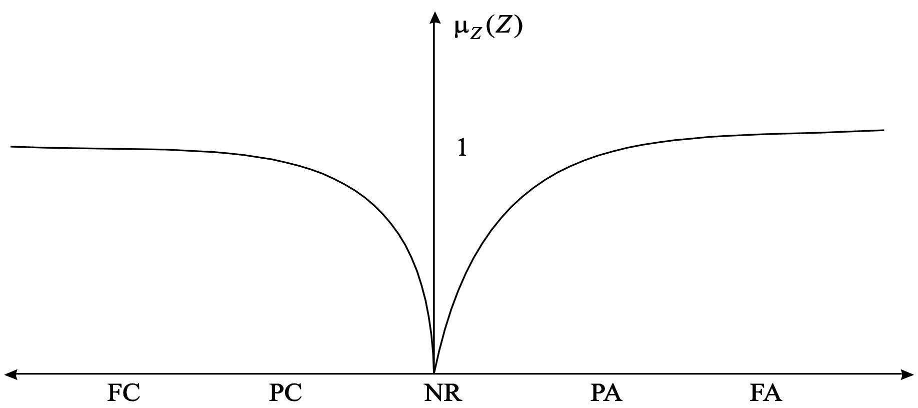


Figure 2.6: Angular fuzzy membership function

Fully anticlockwise (FA) – $\theta = \Pi/2$

Partially anticlockwise (PA) – $\theta = \Pi/4$

No rotation (NR) – $\theta = 0$

Partially clockwise (PC) – $\theta = -\Pi/4$

Fully clockwise (FC) – $\theta = -\Pi/2$

The angular fuzzy set for this is shown in Fig. 2.5. The membership function is shown in Fig. 2.6. The values for membership functions used in Fig. 2.5 is obtained as follows

$$\mu_t(Z) = Z \tan\theta, \text{ where } Z = \cos\theta.$$

Therefore, the angular fuzzy membership values are shown in Table 2.2. Hence, angular fuzzy sets can be used to obtain fuzzy membership values.

Table 2.2: Angular fuzzy membership values

θ	tan θ	Z = cos θ	$\mu_t(Z) = (Z \tan\theta)$
II/2(90°)	∝	0	1
II/4(45°)	1	0.707	0.707
0	0	2	0

(v) Neural Networks

Neural networks are used to simulate the working network of the neurons in the human brain. The concept of the human brain is used to perform computation on computers.

In this case, the fuzzy membership function may be created for fuzzy classes of an input data set. The procedure is the numbers of input data values are selected. Then it is divided into training data set and testing data set. The training data set may be used to train the network.

The generations of membership function from neural network are shown in Fig. 2.7.

Fig. 2.7 (a) shows the training data set. This is passed through a neural network shown in Fig. 2.7 (b). The data points of Fig. 2.7 (a) are divided into three regions as R^1, R^2 and R^3 as in Fig. 2.7 (c). Depending upon the data points, the regions are classified. If the data point is in region 1, then we assign full membership in region 1 and zero membership in regions 2 and 3. Similarly if the data points are in region 2, it will have full membership in region and zero membership in regions so on.

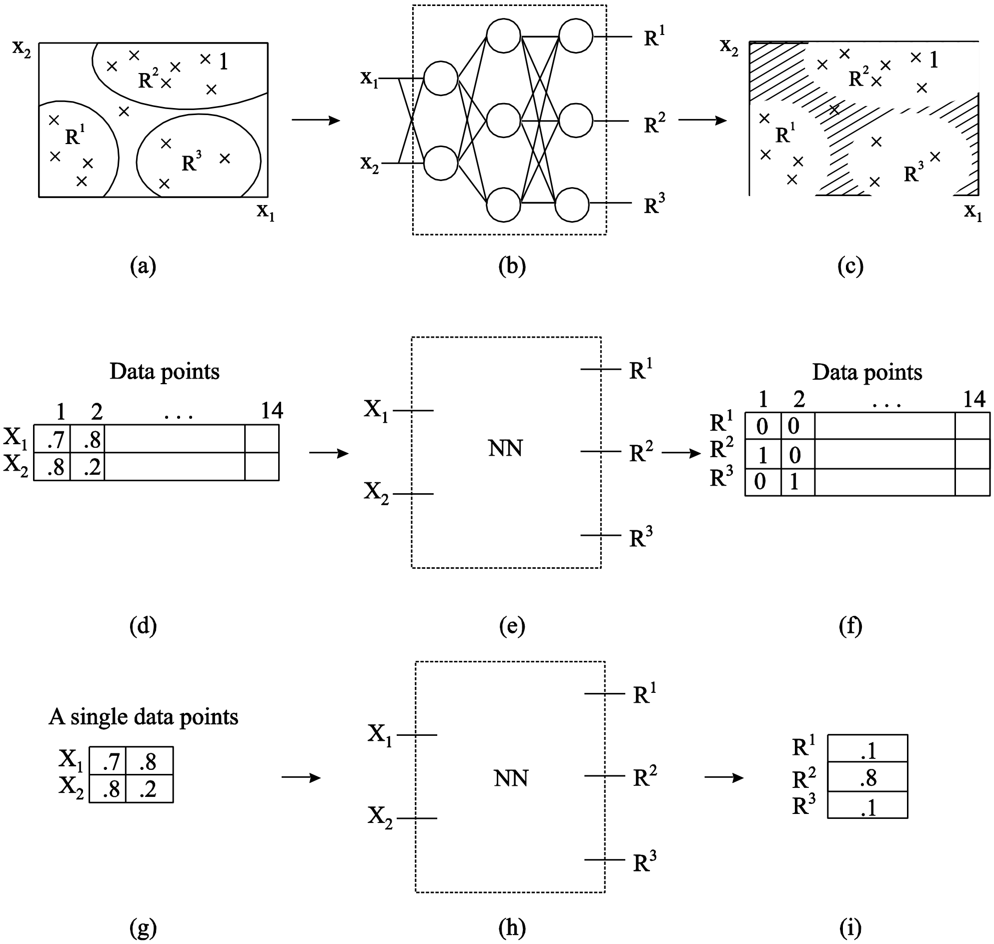


Figure 2.7: Generation of membership function using neural networks

The neural network is then created, from which the training is done between corresponding membership values in different classes, to simulate the relationship between the coordinate locations and membership values. The neural network uses the set of data value and membership values to train itself as shown in Figure 2.7 (d). This training process is continued until the neural network can simulate for the given entire set of input and output value.

After the net is trained, its performance can be checked by the testing data. After full training and testing process is completed, the neural network is ready and it can be used to determine the membership values of any input data in the different regions. These are all shown in Figure 2.7 (g)–(i).

The complete mapping of the membership of different data points in different fuzzy classes can be determined by using neural network approach.

(vi) Genetic Algorithms

Genetic algorithm (GA) uses the concept of Darwin's theory of evolution. Darwin's theory is based on the rule, "survival of the fittest." Darwin also postulated that the new classes of living things came into existence through the process of reproduction, crossover, and mutation among existing organisms.

The steps involved in computing membership functions using GA are:

(a) For the given functional mapping of a system, some membership functions and their shapes are assumed for various fuzzy variables to be defined.

(b) These membership functions are then coded as bit strings.

(c) These bit strings are then concatenated (joined).

(d) Similar to activation function in neural networks, GA has a fitness function.

(e) This fitness function is used to evaluate the fitness of each set of membership functions.

(f) These membership functions are the parameters that define the functional mapping of the system.

Thus, GA can be used to determine the membership functions.

(vii) Inductive Reasoning

The membership can also be generated by the characteristics of inductive reasoning. The induction is performed by the entropy minimization principle, which clusters the parameters corresponding to the output classes. For inductive reasoning method, there should be a well-defined database for the input–output relationships. This method can be suited for complex systems where the data are abundant and static. When the data' are dynamic, this method is not suited, since the membership functions continually change with time.

There are three laws of induction.

(1) Given a set of irreducible outcomes of an experiment, the induced probabilities are those probabilities consistent with all available information that maximize the entropy of the set.

(2) The induced probability of a set of independent observations is proportional to the probability density of the induced probability of a single observation.

(3) The induced rule is that of rule consistent with all available information of which the entropy is minimum.

The third law stated here is the mostly used for membership function development.

The steps involved in generating membership functions using inductive reasoning are as follows:

(a) It is necessary to establish a fuzzy threshold between classes of data.

(b) First, determine the threshold line with an entropy minimization screening method.

(c) After this, start the segmentation process.

(d) The segmentation process first results into two classes.

(e) Further partitioning the first two classes one more time, there is three different classes.

(f) The partitioning is repeated with threshold value calculations, which lead us to partition the data set into a number of classes or fuzzy sets.

(g) Then based on shape, membership function is determined.

Thus the generation of membership function is based on partitioning or analog screening concept. This draws a threshold line between two classes of sample data. The main concept behind drawing the threshold line is to classify the samples when minimizing the entropy for optimum partitioning.

2.4 FUZZY RELATIONS AND THEIR GRAPHS

2.4.1 Crisp Relation

Definition (Cartesian product): Let A and B be two non-empty sets, the product set or Cartesian product $A \times B$ is defined as follows

$$A \times B = \{(a,b) : a \in A, b \in B\}.$$

The concept of Cartesian product can be extended to n sets. For an arbitrary number of sets $A_1, A_2, ..., A_n$, the set of all n-tuples $(a_1, a_2, ..., a_n)$ such that $a_1 \in A_1$, $a_2 \in A_2$, ..., $a_n \in A_n$ is called the Cartesian product and is written as

$$A_1 \times A_2 \times ...... \times A_n \text{ or } \prod_{i=1}^{n} A_i.$$

The product is used for the "composition" of sets and relations in the later sections. For example, a relation is a product space obtained from two sets A and B. $\Re^3 = \Re \times \Re \times \Re$ denotes the 3-dimensional space of real numbers.

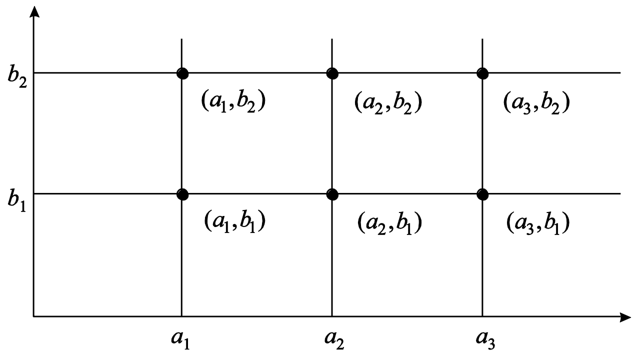


Figure 2.8: Cartesian product $A \times B$

Definition (Binary relation): If *A* and *B* are two sets and there is a specific property between elements *x* of *A* and *y* of *B*, this property can be described using the ordered pair (*x*, *y*). A set of such (*x*, *y*) pairs, $x \in A$ and $y \in B$, is called a relation *R*.

$$R = \{(x,\ y) : x \in A,\ y \in B\}.$$

R is a binary relation and a subset of $A \times B$.

The term "*x* is in relation *R* with *y*" is denoted as $(x,\ y) \in R$ or $x\,R\,y$ with $R \subseteq A \times B$.

Representation Methods of Relations

There are four methods of expressing the relation between sets *A* and *B*.

(i) Bipartigraph

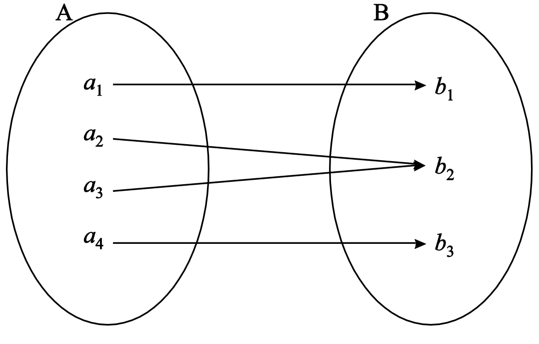


Figure 2.9: Bipartigraph

The first is by illustrating *A* and *B* in a figure and representing the relation by drawing arcs or edges (Fig. 2.9).

(ii) Coordinate diagram

The second is to use a coordinate diagram by plotting members of *A* on *x* axis and that of B on *y* axis, and then the members of $A \times B$ lie on the space. Fig 2.10 shows this type of representation for the relation *R*, namely $x^2 + y^2 = 4$ where $x \in A$ and $y \in B$.

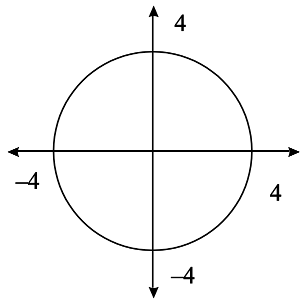


Figure 2.10: Relation of $x^2 + y^2 = 4$

(iii) Matrix

The third method is by manipulating relation matrix. Let *A* and *B* be finite sets having *m* and *n* elements respectively. Assuming *R* is a relation between *A* and *B*, we may represent the relation by matrix $M_R = \left(m_{ij}\right)$ which is defined as follows

$$M_R = (m_{ij}),\ \text{where}\ m_{ij} = \begin{cases} 1,\ (a_i, b_j) \in R;\ i = 1(1)m,\ j = 1(1)n \\ 0,\ (a_i, b_j) \notin R \end{cases}$$

Such matrix is called a relation matrix.

(iv) Digraph

The fourth method is the directed graph or digraph method. Elements are represented as nodes, and relations between elements as directed edges.

A = {1, 2, 3, 4} and *R* ={(1, 1), (1, 2), (2, 1), (2, 2), (1, 3), (2, 4), (4, 1)} for instance.

R	b_1	b_2	b_3
a_1	1	0	0
a_2	0	1	0
a_3	0	1	0
a_4	0	0	1

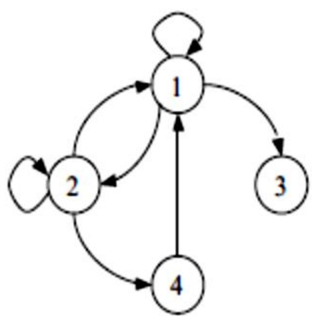


Figure 2.11: Directed graph

Fig 2.11 shows the directed graph corresponding to this relation. When a relation is symmetric, an undirected graph can be used instead of the directed graph.

Table2.4: Comparison of relations

Property Relation	Reflexive	Anti-reflexive	Symmetric	Anti-symmetric	Transitive
Equivalence	✓		✓		✓
Compatibility	✓		✓		
Pre-order	✓				✓
Order	✓			✓	✓
Strict-order		✓		✓	✓

2.4.2 Fuzzy Relation

If a crisp relation *R* represents that of from sets *A* to *B*, for $x \in A$ and $y \in B$, its membership function $\mu_A(x,\ y)$ is

$$\mu_A(x,\ y) = \begin{cases} 1, \text{ iff } (x,\ y) \in R \\ 0, \text{ iff } (x, y) \notin R. \end{cases}$$

This membership function maps $A \times B$ to set {0, 1}, *i.e.*, $\mu_R : A \times B \to \{0,\ 1\}$.

Definition (Fuzzy relation): Fuzzy relation has degree of membership whose value lies in [0, 1].

$$\mu_R : A \times B \to [0,\ 1] \text{ and } R = \Big\{\big((x, y), \mu_R(x, y)\big) : \mu_R(x, y) \geq 0,\ x \in A,\ y \in B\Big\}.$$

Here $\mu_R(x, y)$ is interpreted as strength of relation between *x* and *y*.

Operations on Fuzzy Relations

Let *R* and *T* be fuzzy relation on Cartesian space $X \times Y$. Then the following operations apply for the membership values for various set operations:

Union: The union of two fuzzy relations *R* and *T* is defined by

$$\mu_{R \cup T}(x,\ y) = \max\big(\mu_R(x,\ y),\ \mu_T(x,\ y)\big).$$

Intersection: The union of two fuzzy relations *R* and *T* is defined by

$$\mu_{R \cap T}(x,\ y) = \min\big(\mu_R(x,\ y),\ \mu_T(x,\ y)\big).$$

Complement: The complement of fuzzy relation *R* is given by

$$\mu_{\overline{R}}(x,\ y) = 1 - \mu_R(x,\ y).$$

Containment: The containment of two fuzzy relations *R* and *T* is given by

$$R \subset T \Rightarrow \mu_R(x,\ y) \le \mu_T(x,\ y),\ \forall\ x,\ y.$$

2.4.3 Fuzzy Cartesian product and Composition

Let *A* be a fuzzy set on universe *X* and *B* be a fuzzy set on universe *Y*, then the Cartesian product between fuzzy sets *A* and *B* will result in a fuzzy relation *R* which is contained with the full Cartesian product space or $A \times B = R \subset X \times Y$, where the fuzzy relation *R* has membership function

$$\mu_R(x, y) = \mu_{A\times B}(x, y) = \min\left(\mu_A(x, y),\ \mu_B(x, y)\right).$$

Example 2.4.1: Consider two fuzzy sets *A* and *B*. *A* represents universe of three discrete temperatures $x = \{x_1, x_2, x_3\}$ and *B* represents universe of two discrete flow $y = \{y_1, y_2\}$. Find the fuzzy Cartesian product between them

$$A = \frac{0.4}{x_1} + \frac{0.7}{x_2} + \frac{0.1}{x_3} \quad \text{and} \quad B = \frac{0.5}{y_1} + \frac{0.8}{y_2}.$$

Solution. A represents column vector of size 3 × 1 and B represents column vector of size 1×2. The fuzzy Cartesian product results in a fuzzy relation R of size 3 × 2:

$$A \times B = R = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \end{array} \begin{array}{c} \begin{array}{cc} y_1 & y_2 \end{array} \\ \begin{bmatrix} 0.4 & 0.4 \\ 0.5 & 0.7 \\ 0.1 & 0.1 \end{bmatrix} \end{array}.$$

Composition of Fuzzy Relation

Let R be a fuzzy relation on the Cartesian space X ×Y, S be a fuzzy relation on Y × Z, and T be a fuzzy relation on X × Z, then the fuzzy set max–min composition is defined as:

$T = R \circ S$ (Set-theoretic notation)

$$= \left\{ (x, z), \max_y \left\{ \min\left(\mu_R(x,\ y), \mu_S(y,\ z)\right) \right\} : x \in X, y \in Y, z \in Z \right\}.$$

In function-theoretic form

$$\mu_T(x,\ z) = \bigvee_{y\in Y} \left(\mu_R(x,\ y) \wedge \mu_S(y,\ z)\right).$$

In fuzzy max–product composition is defined in terms of set-theoretic

$$T = R \circ S = \left\{ (x,\ z), \max_y \left(\mu_R(x,\ y) * \mu_S(y,\ z) \right) : x \in X,\ y \in Y,\ z \in Z \right\}$$

$$\mu_T(x,\ z) = \bigvee_{y \in Y} \left(\mu_R(x,\ y) \bullet \mu_S(y,\ z) \right).$$

Max-Average Composition

The max–average composition $S \underset{\sim org \sim}{\circ} R$ is defined as follows:

$$S \underset{\sim org \sim}{\circ} R(x,\ z) = \left\{ (x,\ z), \frac{1}{2} \max_y \left(\mu_R(x,\ y) + \mu_S(y,\ z) \right) : x \in X,\ y \in Y,\ z \in Z \right\}.$$

Example 2.4.2: Consider fuzzy relations

$$R = \begin{array}{c|cc} & y_1 & y_2 \\ x_1 & 0.7 & 0.6 \\ x_2 & 0.8 & 0.3 \end{array}, \quad S = \begin{array}{c|ccc} & z_1 & z_2 & z_3 \\ y_1 & 0.8 & 0.5 & 0.4 \\ y_2 & 0.1 & 0.6 & 0.7 \end{array}.$$

Find the relation $T = R \circ S$ using max–min and max–product composition.

Solution. Max–Min Composition

$T = R \circ S$

$\mu_T(x_1, z_1)$ = max[min(0.7, 0.8), min(0.6, 0.1)]
= max[0.7, 0.1]
= 0.7,

$\mu_T(x_1, z_2)$ = max[min(0.7, 0.5), min(0.6, 0.6)]
= max[0.5, 0.6]
= 0.6,

$\mu_T(x_1, z_3)$ = max[min(0.7, 0.4), min(0.6, 0.7)]
= max[0.4, 0.7]
= 0.7,

$\mu_T(x_2, z_1)$ = max[min(0.8, 0.8), min(0.3, 0.1)]
= max[0.8, 0.1]
= 0.8,

$\mu_T(x_2, z_2)$ = max[min(0.8, 0.5), min(0.3, 0.6)]

$= \max[0.5, 0.3]$

$= 0.5,$

$\mu_T(x_2, z_3) = \max[\min(0.8, 0.4), \min(0.3, 0.7)]$

$= 0.4$

$$T = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.7 & 0.6 & 0.7 \\ 0.8 & 0.5 & 0.4 \end{bmatrix} \end{array}.$$

Max–Product Composition

$$\mu_T(x_1, z_1) = \max\left[(0.7\times 0.8),\ (0.6\times 0.1)\right]$$
$$= \max\left[0.56,\ 0.06\right]$$
$$= 0.56$$

$$\mu_T(x_1, z_2) = \max\left[(0.7\times 0.5),\ (0.6\times 0.6)\right]$$
$$= \max\left[0.35,\ 0.36\right]$$
$$= 0.36$$

$$\mu_T(x_1, z_3) = \max\left[(0.7\times 0.4),\ (0.6\times 0.7)\right]$$
$$= \max\left[0.28,\ 0.42\right]$$
$$= 0.42$$

and so on

$$T = \begin{array}{c} \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.56 & 0.36 & 0.42 \\ 0.64 & 0.40 & 0.32 \end{bmatrix} \end{array}.$$

Example 2.4.3: In the field of computer networking there is an imprecise relationship between the level of use of a network communication bandwidth and the latency experienced in peer-to-peer communication. Let X be a fuzzy set of use levels (in terms of the percentage of full bandwidth used) and Y be a fuzzy set of latencies (in milliseconds) with the following membership function

$$X = \left\{\frac{0.2}{10} + \frac{0.5}{20} + \frac{0.8}{40} + \frac{1.0}{60} + \frac{0.6}{80} + \frac{0.1}{100}\right\},\ Y = \left\{\frac{0.3}{0.5} + \frac{0.6}{1} + \frac{0.9}{1.5} + \frac{1.0}{4} + \frac{0.6}{8} + \frac{0.3}{20}\right\}.$$

Find the Cartesian product represented by the relation $R = X \times Y$.

Reflexivity: Let R be the fuzzy relation on $X \times X$, then R is called reflexive

if $\mu_R(x, x) = 1, \ \forall \ x \in X$.

Example 2.4.4: Let $X = \{x_1, x_2, x_3, x_4\}$, then the following relation is reflexive.

$$R: \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{bmatrix} 1 & 0 & 0.2 & 0.3 \\ 0 & 1 & 0.1 & 1 \\ .2 & 0.7 & 1 & 0.4 \\ 0 & 1 & 0.4 & 1 \end{bmatrix} \end{array}$$

If R_1 and R_2 are reflexive fuzzy relation, then max-min composition $R_1 \circ R_2$ is also reflexive.

Symmetry: Let R be the fuzzy relation on $X \times X$.

(i) R is called symmetric if $\mu_R(x, y) = \mu_R(y, x) \ \forall \ x, y \in X$.

(ii) R is called anti-symmetric if for $x \neq y$

$$\left.\begin{array}{ll} \text{either} & \mu_R(x, y) \neq \mu_R(y, x) \\ \text{or} & \mu_R(x, y) = \mu_R(y, x) = 0 \end{array}\right\} \forall \ x, y \in X.$$

(iii) R is called perfectly symmetric if for $x \neq y$ whenever

$\mu_R(x, y) > 0$ then $\mu_R(y, x) = 0 \ \forall \ x, y \in X$.

Example 2.4.5:

$$R_1: \begin{array}{c|c|c|c|c|} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & .4 & 0 & .1 & .8 \\ \hline x_2 & .8 & 1 & 0 & 0 \\ \hline x_3 & 0 & .6 & .7 & 0 \\ \hline x_4 & 0 & .2 & 0 & 0 \\ \hline \end{array} \quad R_2: \begin{array}{c|c|c|c|c|} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & .4 & 0 & .7 & 0 \\ \hline x_2 & 0 & 1 & .9 & .6 \\ \hline x_3 & .8 & .4 & .7 & .4 \\ \hline x_4 & 0 & .1 & 0 & 0 \\ \hline \end{array} \quad R_2: \begin{array}{c|c|c|c|c|} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & .4 & .8 & .1 & .8 \\ \hline x_2 & .8 & 1 & .0 & .2 \\ \hline x_3 & .1 & .6 & .7 & .1 \\ \hline x_4 & 0 & .2 & 0 & 0 \\ \hline \end{array}$$

R_1 is a perfectly anti symmetric relation, while R_2 is an anti symmetric relation, but not a perfectly anti symmetric relation. R_3 is non symmetric relation, that is, there exist $x, y \in X$ with $\mu_R(x, y) \neq \mu_R(y, x)$, which is not anti symmetric and therefore also not perfectly anti symmetric.

One could certainly define other concepts, such as an α -anti symmetry

$$\left|\mu_R(x,\ y) - \mu_R(y,\ x)\right| \geq \alpha\ \forall\ x, y \in X.$$

These concepts would probably be more in line with the basic ideas of fuzzy set theory. Since we will not need this type of definition for our further considerations, we will abstain from any further definition in the direction.

Example2.4.6: Let $X = \{x_1, x_2, x_3, x_4\}$ and $Y = \{y_1, y_2, y_3, y_4\}$.

The following relation is symmetric relation.

R(x,y):

	y_1	y_2	y_3	y_4
x_1	0	.1	0	.1
x_2	.1	1	.2	.3
x_3	0	.2	.8	.8
x_4	.1	.3	.8	1

Transitivity: A fuzzy relation R is called (max-min) transitive if $R \circ R \subseteq R.$

In other words, a fuzzy relation $R(X, X)$ is transitive (max-min transitive) if

$$\mu_R(x,\ z)\ \geq\ \max_{y \in Y} \min\left[\mu_R(x, y),\ \mu_R(y,\ z)\right]$$

is satisfied for each pair of $(x,\ z) \in X \times X$. A relation failing to satisfy this inequality for some member of X is called non transitive, and if

$$\mu_R(x,\ z)\ <\ \max_{y \in Y} \min\left[\mu_R(x, y),\ \mu_R(y,\ z)\right],$$

for all $(x,\ z) \in X \times X,$ then the relation is called anti transitive.

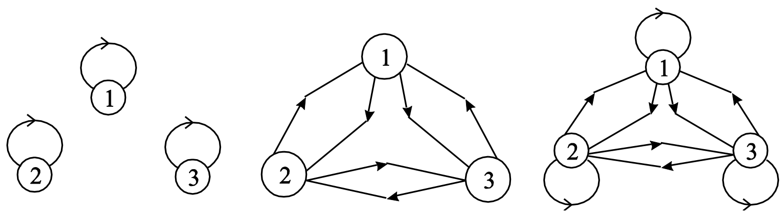


Figure 2.12: Fuzzy relation

For max-min composition, the following properties hold:

(i) R_1 is reflexive and R_2 is an arbitrary fuzzy relation, then $R_1 \circ R_2 \supseteq R_2$ and $R_2 \circ R_1 \supseteq R_2$.

(ii) If R is reflexive, then $R \subseteq R \circ R.$

(iii) R_1 and R_2 are reflexive, then $R_1 \circ R_2$ is reflexive.

(iv) R_1 and R_2 are symmetric, then $R_1 \circ R_2$ is symmetric if $R_1 \circ R_2 = R_2 \circ R_1$.

(v) If R is symmetric, so is each power of R.

Example 2.4.7: Let R be the fuzzy relation defined as

$$R: \begin{array}{c|cccc|} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & .2 & 1 & .4 & .4 \\ x_2 & 0 & .6 & .3 & 0 \\ x_3 & 0 & 1 & .3 & 0 \\ x_4 & .1 & 1 & 1 & .1 \\ \hline \end{array}$$

Then $R \circ R$ is

$$R: \begin{array}{c|cccc|} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & .2 & .6 & .4 & .2 \\ x_2 & 0 & .6 & .3 & 0 \\ x_3 & 0 & .6 & .3 & 0 \\ x_4 & .1 & 1 & .3 & .1 \\ \hline \end{array}$$

Now one can easily see that $\mu_{R \circ R}(x, y) \le \mu_R(x, y)$ holds for all $x, y \in X$.

Combinations of the above properties give some interesting results for max-min composition as given below:

(i) If R is symmetric and transitive, then $\mu_{R \circ R}(x, y) \le \mu_R(x, x)$ for all $x, y \in X$.

(ii) If R is reflexive and transitive, then $R \circ R = R$.

(iii) If R_1 and R_2 are transitive and $R_1 \circ R_2 = R_2 \circ R_1$, then $R_1 \circ R_2$ is transitive.

Similarity or Equivalence Relation

A similarity relation is a fuzzy relation $\mu_R(x, x)$ that is reflexive, symmetric and max-min transitive.

Example 2.4.8: Let's consider a fuzzy relation expressed in the following matrix. Since this relation is reflexive, symmetric and transitive, we see that it is a fuzzy equivalence relation (Fig 1.4.6).

	a	*b*	*c*	*d*
a	1.0	0.8	0.7	1.0
b	0.8	1.0	0.7	0.8
c	0.7	0.7	1.0	0.7
d	1.0	0.8	0.7	1.0

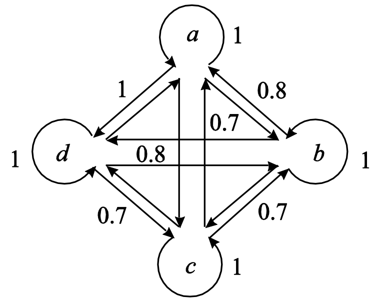


Figure 2.13: Graph of fuzzy equivalence relation

Example 2.4.9: The following relation is a similarity relation:

$$R:\quad \begin{array}{c|cccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \hline x_1 & 1 & .2 & 1 & .6 & .2 & .6 \\ x_2 & .2 & 1 & .2 & .2 & .8 & .2 \\ x_3 & 1 & .2 & 1 & .6 & .2 & .6 \\ x_4 & .6 & .2 & .6 & 1 & .2 & .8 \\ x_5 & .2 & .8 & .2 & .2 & 1 & .2 \\ x_6 & .6 & .2 & .6 & .8 & .2 & 1 \end{array}$$

A similarity relation of a finite number of elements can also be represented by a similarity tree, similar to a dendogram. In this tree, each level represents an α -cut (α -level set) of the similarity relation. For the above similarity relation, the similarity tree is shown below. The sets of elements on specific α -levels can be considered as similarity classes of α -level.

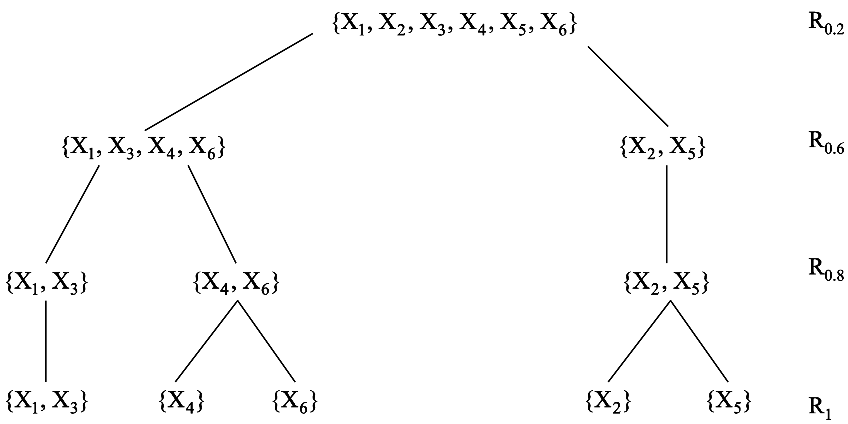


Figure 2.14: Dendogram

We shall now turn to fuzzy order relations: As already mentioned, similarity relations and order relations are primarily distinguished by their degree of symmetry. Roughly speaking, similarity relations are fuzzy relations that are reflexive, (max-min) transitive and symmetrical; order relations, however, are not symmetrical. To be more precise, even different kinds of fuzzy order relations differ by their degree of similarity.

Pre-order Relation

A fuzzy relation that is (max-min) transitive and reflexive is called a fuzzy pre-order relation.

Order Relation

A fuzzy relation that is (max-min) transitive, reflexive and anti symmetric is called a fuzzy order relation. If the relation is perfectly anti symmetrical, it is called a perfect fuzzy order relation. It is also called a fuzzy partial order relation.

Total Order Relation

A order relation or a fuzzy linear ordering is a fuzzy total order relation such that $\forall\ x, y \in X;\ x \neq y$ either $\mu_R(x,\ y) > 0$ or $\mu_R(y,\ x) > 0$.

Any α -cut of a fuzzy linear order is a crisp linear order.

Example 2.4.10: The following relation is fuzzy order relation:

R:

	y_1	y_2	y_3	y_4
x_1	.7	.4	.8	.8
x_2	0	1	0	.2
x_3	0	.6	0	.4
x_4	0	0	0	.7

2.4.4 Fuzzy Graphs

A graph G is defined as follows: $G = (V,\ E)$, where

$V:$ set of vertices. A vertex is also called a node or element.

$E:$ set of edges. An edge is pair $(x,\ y)$ of vertices in V.

A graph is a data structure expressing relation $R \subseteq V \times V$. When order in pair $(x,\ y)$ is defined, the pair is called edge with direction and we call such graph directed graph. When order is not allowed, we call it undirected graph.

When sets A and B are given (including the case $A = B$), let's define a crisp relation $R \subseteq A \times B$. For $x \in A, y \in B,$ if $(x, y) \in R,$ there exists an edge between x and y. In other words,

$$\forall (x,\ y) \in R \Leftrightarrow \mu_R(x,\ y) = \mu_G(x,\ y) = 1.$$

Here given the relation R is a fuzzy relation, and the membership function $\mu_R(x,\ y)$ enables $\mu_G(x,\ y)$ value to be between 0 and 1. Such graph is called a fuzzy graph.

Definition: Let E be the (crisp) set of nodes. A fuzzy graph is then defined by

$$G(x_i, x_j) = \left\{\left((x_i, x_j), \mu_G(x_i, x_j)\right) : (x_i, x_j) \in E \times E\right\}.$$

Example 2.4.11: Figure shows an example of fuzzy graph represented as fuzzy

relation matrix M_G.

M_G	b_1	b_2
a_1	0.8	0.2
a_2	0.3	0.0
a_3	0.7	0.4

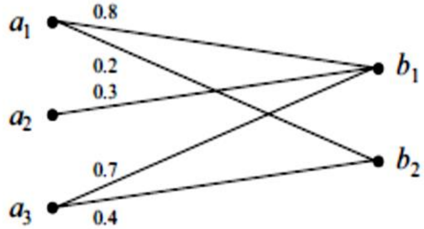


Figure 2.15: Fuzzy graph

Example 2.4.12: There is a set $A \subset \Re$ where $\Re$ is the set of real numbers and $x \in \Re,\ A = \{x : x \text{ close to } 2k\pi, k = 0,1,2,...\}$. The membership function of the set A is formally defined as $\mu_A(x) = \max[0,\ \cos x]$.

(i) Show the graphical representation of A.

(ii) Show the α -cut set of A at α =0.5.

(iii) Show the relation defined as follows:

$$R = \{(x,\ y) : y = \cos x \geq 0,\ x \in A\}$$

(iv) Show the α -cut relation of R at α =0.5.

(v) Show the set defined by $\mu_B(y) = \cos x,\ x \in A.$ This set B is a set induced by R and A.

(vi) Show the relation defined as follows: $\mu_R(x,\ y) = \max[0,\ \sin x],\ x \in A$

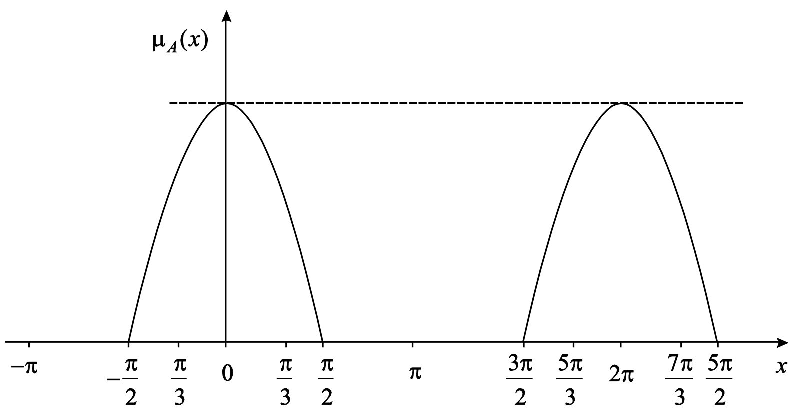


Figure 2.16: Set $\mu_A(x) = \cos x \geq 0$

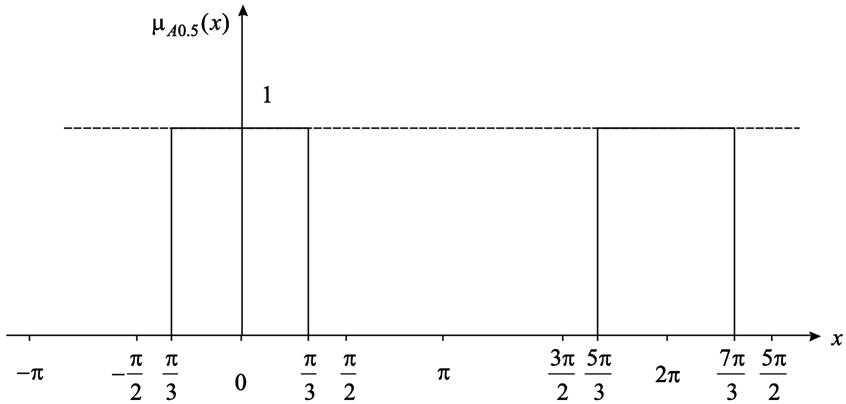


Figure 2.17: $\alpha -$cut set ${}^{0.5}$A

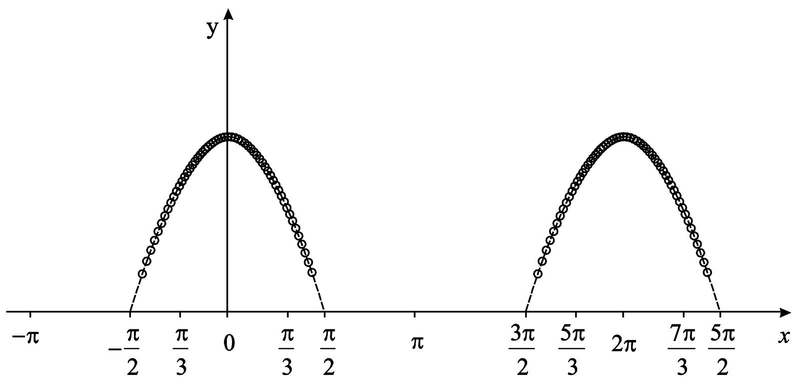


Figure 2.18: Relation $\mu_R(x,\ y) = \cos x$

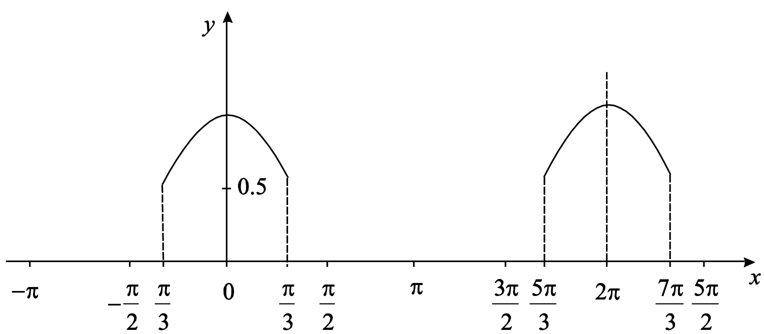


Figure 2.19: $\alpha -$cut relation ${}^{0.5}R$

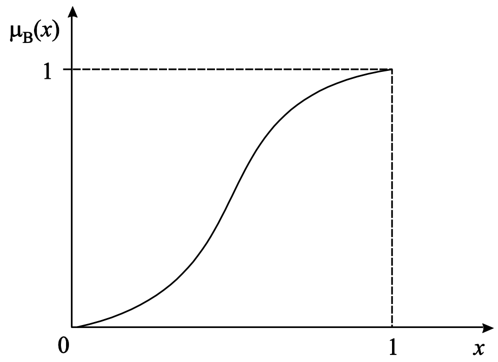


Figure 2.20: $\mu_B(y) = \cos x,\ x \in A$

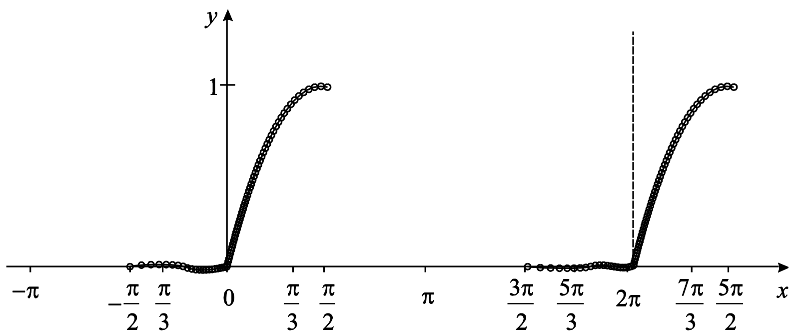


Figure 2.21: $\mu_R(x,\ y) = \max\left[0,\ \sin x\right]$

Sub graph

$H(x_i,\ x_j)$ is a fuzzy sub graph of $G(x_i,\ x_j)$ if

$\mu_H(x_i,\ x_j) \le \mu_G(x_i,\ x_j),\ \forall\ (x_i,\ x_j) \in E \times E.$

$H(x_i,\ x_j)$ spans graph $G(x_i,\ x_j)$ if the node sets of $H(x_i,\ x_j)$ and $G(x_i,\ x_j)$ are equal, that is, if they differ only in their arc weights.

2.4.5 Fuzzy Morphism

Definition (Homomorphism): Given multiple crisp relations $R \subseteq A \times A$ and $S \subseteq B \times B$ homomorphism from (*A*, *R*) to (*B*, *S*) is for the function $h: A \to B$ having the characteristics as for $x_1, x_2 \in A$, $(x_1, x_2) \in R \Rightarrow \left(h(x_1), h(x_2)\right) \in S$

In other words, if two elements x_1 and x_2 are related by *R*, their images $h(x_1)$ and $h(x_2)$ are also related by *S*.

Definition (Strong Homomorphism): Given two crisp relations $R \subseteq A \times A$ and $S \subseteq B \times B$, if the function $h : A \rightarrow B$ satisfies the following, it is called strong homomorphism from (*A*, *R*) to (*B*, *S*).

(i) for $x_1, x_2 \in A$, $(x_1, x_2) \in R \Rightarrow \left(h(x_1), h(x_2)\right) \in S$

(ii) For all $y_1, y_2 \in B$, if

$x_1 \in h^{-1}(y_1),\ x_2 \in h^{-1}(y_2)$ then $(y_1,\ y_2) \in S \Rightarrow (x_1,\ x_2) \in R.$

Definition (Fuzzy Homomorphism): If the relations $R \subseteq A \times A$ and $S \subseteq B \times B$ are fuzzy relations, the above Morphism is extended to a fuzzy homomorphism as follows

For all $x_1, x_2 \in A$ and their images $h(x_1), h(x_2) \in B$,

$$\mu_R(x_1, x_2) \le \mu_S\left(h(x_1), h(x_2)\right).$$

in other words, the strength of the relation *S* for $\left(h(x_1), h(x_2)\right)$ is stronger than or equal to the that of *R* for (x_1, x_2).

If a homomorphism exists between fuzzy relations (*A*, *R*) and (*B*, *S*), the homomorphism *h* partitions *A* into subsets $A_1, A_2, \ldots, A_n$ because it is a many-to-one mapping $\forall x_i \in A_j,\ i = 1, 2, \ldots, n,\ \ h(x_i) = y \in B$ so to speak, image *h*(*xi*) of elements *xi* in *Aj* is identical to element *y* in *B*. In this manner, every element in *A* shall be mapped to one of *B*. If the strength between A_j and A_k gets the maximum strength between $x_j \in A_j$ and $x_k \in A_k$, thus morphism is replaced with fuzzy strong homomorphism.

Definition (Fuzzy strong homomorphism): Given the fuzzy relations *R* and *S*, if *h* satisfies the followings, *h* is a fuzzy strong homomorphism.

For all $x_j \in A_j, x_k \in A_k,\ \ A_j,\ A_k \subseteq A$

$$y_1 = h(x_j),\ y_2 = h(x_k),\ y_1,\ y_2 \in B,\ \left(y_1,\ y_2\right) \in S,$$

$$\max_{x_j,\, x_k} \mu_R(x_j,\ x_k) = \mu_S(y_1,\ y_2).$$

Example 2.4.13: Consider the relations $R \subseteq A \times A$ and $S \subseteq B \times B$ in the following

R	a	b	c	d
a	0.0	0.6	0.0	0.0
b	0.0	0.0	0.8	0.0
c	1.0	0.0	0.0	0.0
d	0.0	0.6	0.0	0.0

S	α	β	γ
α	0.6	0.8	0.0
β	1.0	0.0	0.6
γ	0.6	0.0	0.0

we apply the mapping function *h* from A to B as follows $h:\ a, b \to \alpha;\ \ c \to \beta;\ \ d \to \gamma.$

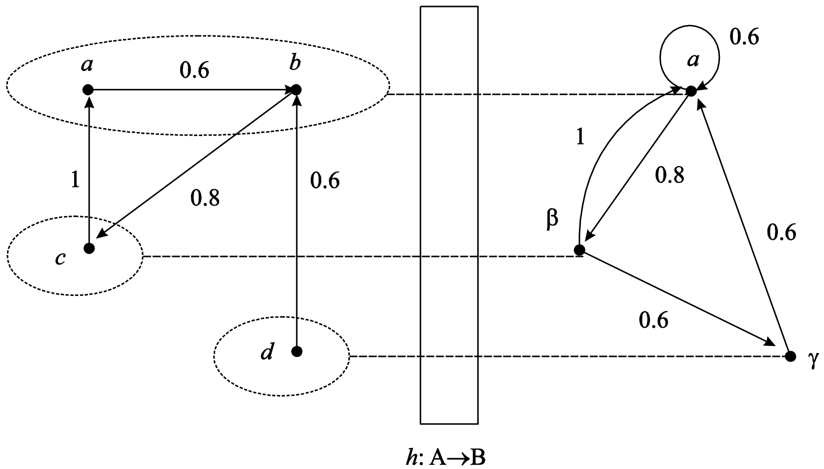


Figure 2.22: Fuzzy homomorphism

Here all $(x_1,\ x_2) \in R,$ of A has the relation $(h(x_1),\ h(x_2)) \in S,$ in $B.$ Furthermore $\mu_R(x_1,\ x_2) \le \mu_S(h(x_1),\ h(x_2))$ holds.

For example $h(c) = \beta,\ h(d) = \gamma,\ \ \mu_R(c,\ d) = 0 \le \mu_S(\beta,\ \gamma) = 0.6.$

As a consequence, this morphism is a fuzzy homomorphism (Fig 2.22). We know $\mu_S(\beta,\ \gamma) = 0.6$, but we are not able to find its corresponding pair in *R*. Therefore it is not a fuzzy strong homomorphism.

Example 2.4.14: We have two relations $R \subseteq A \times A$ and $S \subseteq B \times B.$

R	a	b	c	d	e
a	0.5	0.5	0.0	0.0	0.0
b	1.0	0.0	0.5	0.0	0.0
c	0.0	0.0	0.0	1.0	0.5
d	0.0	0.0	0.9	0.0	0.0
e	0.0	0.0	0.0	1.0	0.0

S	α	β	γ
α	0.5	0.5	0.0
β	1.0	0.5	1.0
γ	0.0	0.9	1.0

We have also a function from A to B as follows $h: a \to \alpha;\ b, c \to \beta;\ d, e \to \gamma.$

Here for all $\left(x_1,\ x_2\right) \in R,$ of A has the relation $\left(h(x_1),\ h(x_2)\right) \in S,$ in B and inversely, for all $(y_1,\ y_2) \in S,\ \left(h^{-1}(y_1),\ h^{-1}(y_2)\right) \in R$ in A. Thus h completes the conditions for fuzzy homomorphism. Now, let's consider the conditions for the fuzzy strong homomorphism.

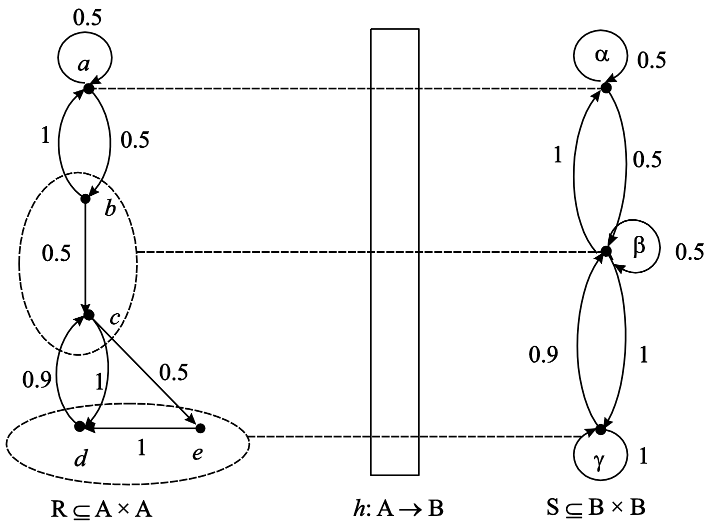


Figure 2.23: Fuzzy strong homomorphism

For example, $(\beta,\ \gamma) \in S,\ \mu_S(\beta,\ \gamma) = 1,\ h^{-1}(\beta) = \{b, c\},\ h^{-1}(\gamma) = \{d, e\}$

$$\max\left[\mu_R(c,\ d), \mu_R(c, e)\right] = \max\left[1,\ 0.5\right] = 1 = \mu_S(\beta, \gamma)$$

in the same manner, we can verify for other pairs and then we see the morphism h is a fuzzy strong homomorphism (Fig 2.23).

2.4.6. Extension Principle: An introduction to Fuzzy Control

One of the most important notions in fuzzy set theory is the extension principle. The extension principle provides a general method for combining non-fuzzy and fuzzy concepts of all kinds, e.g., for combining fuzzy sets and relations, but also for the operation of a mathematical function on fuzzy sets. Fuzzy sets can also be interpreted as fuzzy numbers. In this Case one can use the extension principle to add or multiply these fuzzy numbers.

Let $A_1, \ldots, A_n$ be fuzzy sets, defined respectively on $U_1, \ldots, U_n,$ and let f be a non-fuzzy function $f: U_1 \times \ldots \times U_n \to V.$ The aim is to extend f such that it operates on $A_1, \ldots, A_n,$ and returns a fuzzy set F on $V.$ This is done by using the sup-min composition as follows:

Definition (Extension Principle): The extension of f, operating on $A_1,\ldots,A_n$ results in the following membership function for F

$$\mu_F(v) = \sup_{\substack{u_1,\ldots,u_n \\ f(u_1,\ldots,u_n)=v}} \min\left(\mu_{A_1}(u_1),\ldots,\mu_{A_n}(u_n)\right), \tag{2.4.1}$$

when $f^{-1}(v)$ exists. Otherwise $\mu_A(v)=0.$

Another way to obtain this result is

$$F = \int_{U_1\times\ldots\times U_n} \frac{\min\left(\mu_{A_1}(u_1),\ldots,\mu_{A_n}(u_n)\right)}{f\left(u_1,\ldots,u_n\right)}. \tag{2.4.2}$$

In the binary case and on a discrete or compact domain, (2.4.1) is given by

$$\mu_{f(A_1,A_2)}(y) = \max_{\substack{x_1,x_2 \\ y=f(x_1,x_2)}} \min\left(\mu_{A_1}(x_1),\mu_{A_2}(x_2)\right). \tag{2.4.3}$$

So the function f is extended from the domain of real numbers to the domain of fuzzy numbers.

Example 2.4.15: The extension principle is often used in fuzzy arithmetic. Let m and n be two fuzzy numbers (fuzzy sets defined on $\mathbb{R}$), then the membership function of the sum of m and n (denoted $m\oplus n$) is

$$\mu_{m\oplus n}(z) = \sup_{\substack{x,y \\ z=x+y}} \min\left(\mu_m(x),\mu_n(y)\right). \tag{2.4.4}$$

So, the original function, in this case + is given in the subscript of the supremum operation. It is worthwhile to work this out. Let $\tilde{6}$ be the fuzzy number with membership degree $\Lambda\left(x;4,6,8\right)$ and 4 the fuzzy number with membership degree $\Lambda\left(y;3,4,5\right)$, then $\tilde{6}\oplus 4$ has membership degree $\Lambda\left(z;7,10,13\right), i.e.,$ approximately 10, but quite fuzzy. As a formula,

$$\tilde{6}\oplus 4 = \int_{\mathbb{R}} \max_{z=x+y} \min\left(\mu_{\tilde{6}}(x),\mu_4(y)\right)\Big/z = \int_{z=x+y} \Lambda\left(z;7,10,13\right)\big/z \tag{2.4.5}$$

In the same way, subtractions is defined by

$$\mu_{m\Theta n}(z) = \sup_{\substack{x,y \\ z=x-y}} \min\left(\mu_m(x),\mu_n(y)\right) \tag{2.4.6}$$

and

$$\mu_{m\otimes n}(z) = \sup_{\substack{x,y \\ z=x\times y}} \min\left(\mu_m(x), \mu_n(y)\right). \tag{2.4.7}$$

While the domain in fuzzy control usually is either discrete or compact, one can almost always use the max-min composition instead of the sup-min composition. An 'extended' extension operator can be found by using $\underset{\vee}{*} - \underset{*}{\wedge}$ (S-norm-T-norm) composition. That is,

$$\mu_F(v) = \underset{\substack{u_1,\ldots,u_n \\ f(u_1,\ldots,u_n)=v}}{\underset{\vee}{*}} \mu_{A_1}(u_1) \underset{*}{\wedge} \ldots \underset{*}{\wedge} \mu_{A_n}(u_n) \tag{2.4.8}$$

On continuous domains one has to be careful with choice of $\underset{\vee}{*}$. Normally, only the supremum operation and the S_W operation can be used. Another well known composition operation is the max-product or max-dot composition, that is,

$$\mu_F(v) = \max_{\substack{u_1,\ldots,u_n \\ f(u_1,\ldots,u_n)=v}} \mu_{A_1}(u_1) \cdot \mu_{A_2}(u_2) \cdot \ldots \cdot \mu_{A_n}(u_n). \tag{2.4.9}$$

2.5 CLASSICAL AND FUZZY LOGIC

2.5.1 Introduction

Logic is but a small part of the human capacity to reason. Logic can be a means to compel us to infer correct answers, but it cannot by itself be responsible for our creativity or for our ability to remember. In other words, logic can assist us in organizing words to make clear sentences, but it cannot help us determine what sentences to use in various contexts. Consider the passage above from the nineteenth-century mathematician Lewis Carroll in his classic *Through the Looking Glass*. How many of us can see the logical context in the discourse of these fictional characters? Logic for humans is a way quantitatively to develop a reasoning process that can be replicated and manipulated with mathematical precepts. The interest in logic is the study of truth in logical propositions; in classical logic this truth is binary – a proposition is either true or false.

This section introduces the reader to fuzzy logic with a review of classical logic and its operations, logical implications, and certain classical inference mechanisms such as tautologies. The concept of a proposition is introduced as are associated concepts of truth sets, tautologies, and contradictions. The operations of disjunction, conjunction, and negation are introduced as well as classical implication and equivalence; all of these are useful tools to construct compound propositions from single propositions. Operations on propositions are shown to be isomorphic with operations on sets

2.5.2 Classical Logic

In classical logic, a simple proposition P is a linguistic, or declarative, statement contained within a universe of elements, X (say), that can be identified as being a collection of elements in X that are strictly true or strictly false. Hence, a proposition P is a collection of elements, i.e., a set, where the truth values for all elements in the set are either all true or all false. The veracity (truth) of an element in the proposition P can be assigned a binary truth value, called $T(P)$ just as an element in a universe is assigned a binary quantity to measure its membership in a particular set. For binary (Boolean) classical logic, $T(P)$ is assigned a value of 1 (truth) or 0 (false). If U is the universe of all propositions, then T is a mapping of the elements, $u,$ in these propositions (sets) to the binary quantities (0, 1), or

$$T : u \in U \rightarrow (0, 1).$$

All elements u in the universe U that are true for proposition P are called the truth set of $P,$ denoted $T(P).$ Those elements u in the universe U that are false for proposition P are called the falsity set of $P.$

In logic we need to postulate the boundary conditions of truth values just as we do for sets; that is, in function-theoretic terms we need to define the truth value of a universe of discourse. For a universe Y and the null set $\phi,$ we define the following truth values:

$$T(Y) = 1 \text{ and } T(\phi) = 0.$$

Now let P and Q be two simple propositions on the same universe of discourse that can be combined using the following five logical connectives

Disjunction	$(\vee)$
Conjunction	$(\wedge)$
Negation	$(-)$
Implication	$(\rightarrow)$
Equivalence	$(\leftrightarrow)$

to form logical expressions involving the two simple propositions. These connectives can be used to form new propositions from simple propositions.

The disjunction connective, the logical *or*, is the term used to represent what is commonly referred to as the *inclusive or*. The natural language term *or* and the logical *or* differ in that the former implies exclusion (denoted in the literature as the *exclusive or*; further details are given in this chapter). For example, ‘‘soup or salad’’ on a restaurant menu implies the choice of one or the other option, but not both. The *inclusive or* is the one most often employed in logic; the inclusive or (*logical or* as used here) implies that a compound proposition is true if either of the simple propositions is true or both are true.

The equivalence connective arises from dual implication; that is, for some propositions P and Q, if $P \rightarrow Q$ and $Q \rightarrow P$, then $P \leftrightarrow Q$.

Now define sets A and B from universe X (universe X is isomorphic with universe U), where these sets might represent linguistic ideas or thoughts. A *propositional calculus* (sometimes called the *algebra of propositions*) will exist for the case where proposition P measures the truth of the statement that an element, x, from the universe X is contained in set A and the truth of the statement Q that this element, x, is contained in set B, or more conventionally,

P: truth that $x \in A$ and Q: truth that $x \in B$

where truth is measured in terms of the truth value, i.e.,

$$\text{if } x \in A,\ T(P) = 1;\ \text{otherwise, } T(P) = 0$$

$$\text{if } x \in B,\ T(Q) = 1;\ \text{otherwise, } T(Q) = 0$$

or, using the characteristic function to represent truth (1) and falsity (0), the following notation results:

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

A notion of mutual exclusivity arises in this calculus. For the situation involving two propositions P and Q, where $T(P) \cap T(Q) = \phi$, we have that the truth of P always implies the falsity of Q and vice versa; hence, P and Q are mutually exclusive propositions.

Example 2.5.1: Let P be the proposition ‘‘the structural beam is an 18WF45’’ and let Q be the proposition ‘‘the structural beam is made of steel.’’ Let X be the universe of structural members comprised of girders, beams, and columns; x is an element (beam), A is the set of all wide-flange (WF) beams, and B is the set of all steel beams. Hence,

P: x is in A and Q: x is in B.

The five logical connectives already defined can be used to create compound propositions, where a compound proposition is defined as a logical proposition formed by logically connecting two or more simple propositions. Just as we are interested in the truth of a simple proposition, classical logic also involves the assessment of the truth of compound propositions. For the case of two simple propositions, the resulting compound propositions are defined next in terms of their binary truth values.

Given a proposition $P : x \in A$, $\overline{P} : x \notin A$, we have the following for the logical connectives:

Disjunction $P \vee Q : x \in A \ or \ x \in B \left[i.e.\ T(P \vee Q) = \max\left(T(P), T(Q)\right) \right]$ (2.5.1)

Conjunction $P \wedge Q : x \in A$ and $x \in B \left[i.e. \ T(P \wedge Q) = \min\left(T(P), T(Q)\right) \right]$ (2.5.2)

Negation If $T(P) = 1,$ then $T(\overline{P}) = 0;$ if $T(P) = 0,$ then $T(\overline{P}) = 1.$ (2.5.3)

Implication $(P \to Q) : x \notin A \ or \ x \in B \left[i.\ e., T(P \to Q) = T(\overline{P} \cup Q) \right]$ (2.5.4)

Equivalence $(P \leftrightarrow Q) : T(P \leftrightarrow Q) = \begin{cases} 1, \text{ for } T(P) = T(Q) \\ 0, \text{ for } T(P) \neq T(Q) \end{cases}$ (2.5.5)

The logical connective *implication*, i.e., $P \to Q$ (P implies Q), presented here is also known as the classical implication, to distinguish it from an alternative form devised in the 1930s by Lukasiewicz, a Polish mathematician, who was first credited with exploring logics other than Aristotelian (classical or binary logic) [Rescher (1969)], and from several other forms (see end of this chapter). In this implication the proposition P is also referred to as the *hypothesis* or the *antecedent*, and the proposition Q is also referred to as the *conclusion* or the *consequent*. The compound proposition $P \to Q$ is true in all cases except where a true antecedent P appears with a false consequent, Q, i.e., a true hypothesis cannot imply a false conclusion.

Example 2.5.2 [Gill (1976)]: Consider the following four propositions:

1. If 1+1=2, then $4 > 0.$
2. If 1+1=3, then $4 > 0.$
3. If 1+1=3, then $4 < 0.$
4. If 1+1=2, then $4 < 0.$

The first three propositions are all true; the fourth is false. In the first two, the conclusion 4 > 0 is true regardless of the truth of the hypothesis; in the third case both propositions are false, produce a true conclusion.

Hence, the classical form of the implication is true for all propositions of P and Q except for those propositions that are in both the truth set of P and the false set of $Q,$ i.e.,

$$T(P \to Q) = \overline{T(P) \cap T(\overline{Q})} \tag{2.5.6}$$

This classical form of the implication operation requires some explanation. For a proposition P defined on set A and a proposition Q defined on set $B,$ the implication ‘‘ P implies Q ’’ is equivalent to taking the union of elements in the complement of set A with the elements in the set B (this result can also be derived by using De Morgan’s principles on (2.5.6). That is, the logical implication is analogous to the set-theoretic form

$$(P \to Q) \equiv \left(\overline{A} \cup B\right) \text{ is true} \equiv \text{either "not in } A\text{" or "in } B\text{"}$$

so that $$T(P \to Q) = T(\overline{P} \vee Q) = \max\left(T(\overline{P}),\, T(Q)\right) \qquad (2.5.7)$$

This expression is linguistically equivalent to the statement, " $P \to Q$ is true" when either "not A" or " B" is true (logical or). Graphically, this implication and the analogous set operation are represented by the Venn diagram in Fig. 2.5.1. As noted in the diagram, the region represented by the difference $A|B$ is the set region where the implication $P \to Q$ is false (the implication "fails"). The shaded region in Fig. 2.5.1 represents the collection of elements in the universe where the implication is true; that is, the set

$$\overline{A|B} = \overline{A} \cup \overline{\overline{B}} = \overline{A \cup \overline{B}}.$$

If x is in A *and* x is not in B, then

$$A \to B \text{ fails} \equiv A|B \text{ (difference)}$$

Now, with two propositions (P and Q) each being able to take on one of two truth values (true or false, 1 or 0), there will be a total of $2^2 = 4$ propositional situations. These situations are illustrated, along with the appropriate truth values, for the propositions P and

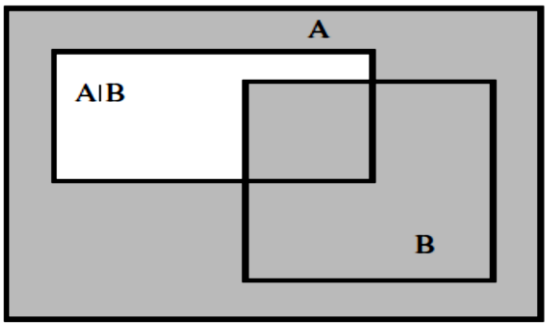


Figure 2.24 Graphical analog of the classical implication operation; gray area is where implication holds.

Table 2.5: Truth table for various compound propositions

P	**Q**	$\overline{\mathbf{P}}$	**P ∨ Q**	**P ∧ Q**	**P → Q**	**P ↔ Q**
T (1)	T (1)	F (0)	T (1)	T (1)	T (1)	T (1)
T (1)	F (0)	F (0)	T (1)	F (0)	F (0)	F (0)
F (0)	T (1)	T (1)	T (1)	F (0)	T (1)	F (0)
F (0)	F (0)	T (1)	F (0)	F (0)	T (1)	T (1)

and the various logical connectives between them in Table 5.1. The values in the last five columns of the table are calculated using the expressions in (2.5.1) and (2.5.7). In Table 2.5 T (or 1) denotes true and F (or 0) denotes false.

Suppose the implication operation involves two different universes of discourse; P is a proposition described by set A, which is defined on universe X, and Q is a proposition described by set B, which is defined on universe Y. Then the implication $P \to Q$ can be represented in set-theoretic terms by the relation R, where R is defined by

$$R = (A \times B) \cup (\overline{A} \times Y) \equiv IF\ \ A,\ \ THEN\ B$$

$$IF\ \ x \in A \text{ where } x \in X \text{ and } A \subset X\ \ THEN\ \ y \in B \text{ where } y \in Y \text{ and } B \subset Y \qquad (2.5.8)$$

This implication, (2.5.8), is also equivalent to the linguistic rule form, IF A, THEN B. The graphic shown in Fig. 2.5.2 represents the space of the Cartesian product X × Y, showing typical sets A and B; and superposed on this space is the set-theoretic equivalent of the implication. That is,

$$P \to Q : IF\ \ x \in A,\ THEN\ \ y \in B, \text{ or } P \to Q \equiv \overline{A} \cup B.$$

The shaded regions of the compound Venn diagram in Fig. 2.5.2 represent the truth domain of the implication, IF A, THEN $B\,(P \to Q)$.

Another compound proposition in linguistic rule form is the expression $IF\ \ A,\ THEN\ \ B,\ ELSE\ \ C$

Linguistically, this compound proposition could be expressed as

$$IF\ \ A,\ THEN\ \ B, \text{ and } IF\ \overline{A},\ THEN\ C$$

In classical logic this rule has the form

$$(P \to Q) \wedge (\overline{P} \to S) \qquad (2.5.9)$$

$$P : x \in A,\ A \subset X;\ \ Q : y \in B,\ B \subset Y;\ \ S : y \in C,\ C \subset Y$$

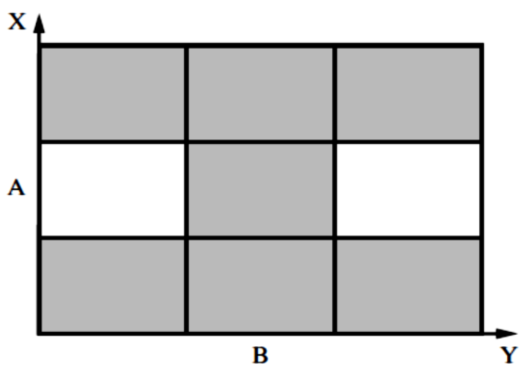


Figure 2.25 The Cartesian space showing the implication IF A, THEN B

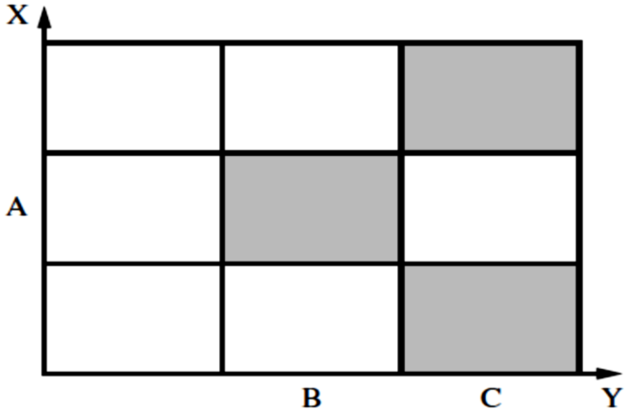


Figure 2.26 Truth domain for IF A, THEN B, ELSE C.

The set-theoretic equivalent of this compound proposition is given by

$$IF\ \ A, THEN\ \ B,\ ELSE\ \ C \equiv (A \times B) \cup (\overline{A} \times C) = R = \text{relation on } X \times Y. \qquad (2.5.10)$$

The graphic in Fig. 2.26 illustrates the shaded region representing the truth domain for this compound proposition for the particular case where $B \cap C = \phi$.

Tautologies

In classical logic it is useful to consider compound propositions that are always true, irrespective of the truth values of the individual simple propositions. Classical logical compound propositions with this property are called tautologies. Tautologies are useful for deductive reasoning, for proving theorems, and for making deductive inferences. So, if a compound proposition can be expressed in the form of a tautology, the truth value of that compound proposition is known to be true. Inference schemes in expert systems often employ tautologies because tautologies are formulas that are true on logical grounds alone. For example, if A is the set of all prime numbers $(A_1 = 1,\ A_2 = 2,\ A_3 = 3,\ A_4 = 5, \ldots)$ on the real line universe, X, then the proposition ‘‘Ai is not divisible by 6’’ is a tautology.

One tautology, known as modus ponens deduction, is a very common inference scheme used in forward-chaining rule-based expert systems. It is an operation whose task is to find the truth value of a consequent in a production rule, given the truth value of the antecedent in the rule. Modus ponens deduction concludes that, given two propositions, P and $P \rightarrow Q$, both of which are true, then the truth of the simple proposition Q is automatically inferred. Another useful tautology is the modus tollens inference, which is used in backward-chaining expert systems. In modus tollens an implication between two propositions is combined with a second proposition and both are used to imply a third proposition. Some common tautologies follow:

$$\overline{B} \cup B = X; \quad A \cup X; \quad \overline{A} \cup X \leftrightarrow X$$

$$\big(A \wedge (A \rightarrow B)\big) \rightarrow B \ \ \text{(modus ponens)} \qquad (2.5.11)$$

$$\left(\overline{B} \wedge (A \to B)\right) \to \overline{A} \ \text{(modus tollens)} \tag{2.5.12}$$

A simple proof of the truth value of the modus ponens deduction is provided here, along with the various properties for each step of the proof, for purposes of illustrating the utility of a tautology in classical reasoning.

Proof

$$\left(A \wedge (A \to B)\right) \to B$$

$$\left(A \wedge (\overline{A} \cup B)\right) \to B \ \text{Implication}$$

$$\left((A \wedge \overline{A}) \cup (A \wedge B)\right) \to B \ \text{Distributivity}$$

$$\left(\phi \cup (A \wedge B)\right) \to B \ \text{Excluded middle axioms}$$

$$\left(A \wedge B\right) \to B \ \text{Identity}$$

$$\left(\overline{A \wedge B}\right) \cup B \ \text{Implication}$$

$$\left(\overline{A} \vee \overline{B}\right) \cup B \ \text{De Morgan's principles}$$

$$\overline{A} \vee \left(\overline{B} \cup B\right) \ \text{Associativity}$$

$$\overline{A} \cup X \ \text{Excluded middle axioms}$$

$$X \Rightarrow T(X) = 1 \ \text{Identity; QED}$$

A simpler manifestation of the truth value of this tautology is shown in Table 2.6 in truth table form, where a column of all ones for the result shows a tautology.

Table 2.6 Truth Table (modus ponens)

A	**B**	**A → B**	**(A ∧ (A → B))**	**(A ∧ (A → B)) → B**	
0	0	1	0	1	
0	1	1	0	1	Tautology
1	0	0	0	1	
1	1	1	1	1	

Table 2.7 Truth Table (modus tollens)

A	B	$\overline{A}$	$\overline{B}$	$A \rightarrow B$	$(\overline{B} \wedge (A \rightarrow B))$	$(\overline{B} \wedge (A \rightarrow B)) \rightarrow \overline{A}$	
0	0	1	1	1	1	1	
0	1	1	0	1	0	1	Tautology
1	0	0	1	0	0	1	
1	1	0	0	1	0	1	

Similarly, a simple proof of the truth value of the modus tollens inference is listed here.

Proof

$$\left(\overline{B} \wedge (A \rightarrow B)\right) \rightarrow \overline{A}$$

$$\left(\overline{B} \wedge (\overline{A} \cup B)\right) \rightarrow \overline{A}$$

$$\left((\overline{B} \wedge \overline{A}) \cup (\overline{B} \wedge B)\right) \rightarrow \overline{A}$$

$$\left((\overline{B} \wedge \overline{A}) \cup \phi\right) \rightarrow \overline{A}$$

$$\left(\overline{B} \wedge \overline{A}\right) \rightarrow \overline{A}$$

$$\left(\overline{\overline{B} \wedge \overline{A}}\right) \cup \overline{A}$$

$$\left(\overline{\overline{B}} \vee \overline{\overline{A}}\right) \cup \overline{A}$$

$$B \cup \left(A \cup \overline{A}\right)$$

$$B \cup X = X \Rightarrow T(X) = 1 \text{ QED}$$

The truth table form of this result is shown in Table 2.7.

Contradictions

Compound propositions that are always false, regardless of the truth value of the individual simple propositions constituting the compound proposition, are called contradictions. For example, if A is the set of all prime numbers $(A_1 = 1,\ A_2 = 2,\ A_3 = 3,\ A_4 = 5, \ldots)$ on the real line universe, X, then the proposition ‘‘ A_i is a multiple of 4’’ is a contradiction. Some simple contradictions are listed here:

$$\overline{B} \cap B;\ A \cap \phi;\ \overline{A} \cap \phi.$$

Equivalence

As mentioned, propositions P and Q are equivalent, i.e., $P \leftrightarrow Q$, is true only when both P and Q are true or when both P and Q are false. For example, the propositions P: ‘‘triangle is equilateral’’ and Q: ‘‘triangle is equiangular’’ are equivalent because they are either both true or both false for some triangle. This condition of equivalence is shown in Fig. 2.27, where the shaded region is the region of equivalence.

It can be easily proved that the statement $P \leftrightarrow Q$ is a tautology if P is identical to Q, i.e., if and only if $T(P) = T(Q)$.

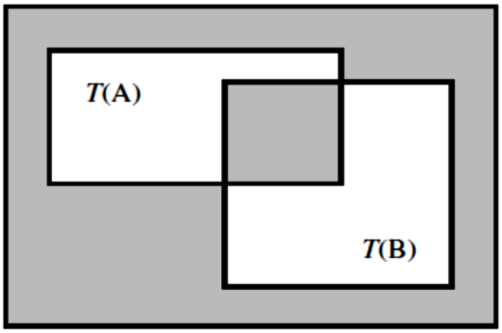


Figure 2.27: Venn diagram for equivalence

Example 2.5.3: Suppose we consider the universe of positive integers, $X = \{1 \le n \le 8\}$. Let P = ‘‘n is an even number’’ and let $Q = "(3 \le n \le 7) \wedge (n \ne 6)."$ Then $T(P) = \{2,4,6,8\}$ and $T(Q) = \{3,4,5,7\}$. The equivalence $P \leftrightarrow Q$ has the truth set

$$T(P \leftrightarrow Q) = \left(T(P) \cap T(Q)\right) \cup \left(\overline{T(P)} \cap \overline{T(Q)}\right) = \{4\} \cup \{1\} = \{1,\ 4\}$$

One can see that ‘‘1 is an even number’’ and $"(3 \le 1 \le 7) \wedge (1 \ne 6)"$ are both false, and ‘‘4 is an even number’’ and $"(3 \le 4 \le 7) \wedge (4 \ne 6)"$ are both true.

Example 2.5.4: Prove that $P \leftrightarrow Q$ if P = ‘‘n is an integer power of 2 less than 7 and greater than zero’’ and $Q = "n^2 - 6n + 8 = 0."$ Since $T(P) = \{2,4\}$ and $T(Q) = \{2,4\}$ it follows that $P \leftrightarrow Q$ is an equivalence.

Suppose a proposition R has the form $P \to Q$. Then the proposition $Q \to P$ is called the contrapositive of R; the proposition $Q \to P$ is called the converse of R; and the proposition $P \to Q$ is called the inverse of R.

The dual of a compound proposition that does not involve implication is the same proposition with false (0) replacing true (1) (i.e., a set being replaced by its complement), true replacing false, conjunction ($\wedge$) replacing disjunction ($\vee$) and disjunction replacing conjunction. If a proposition is true, then its dual is also true.

Exclusive Or and Exclusive Nor

Two more interesting compound propositions are worthy of discussion. These are the exclusive or and the exclusive nor. The exclusive or is of interest because it arises in many situations involving natural language and human reasoning. For example, when you are going to travel by plane or boat to some destination, the implication is that you can travel by air or sea, but not both, i.e., one or the other. For two propositions, P and Q, the exclusive or, denoted here as XOR, is given in Table 2.27 and Fig. 2.28.

Table 2.8: Truth table for exclusive or

P	Q	P *XOR* Q
1	1	0
1	0	1
0	1	1
0	0	0

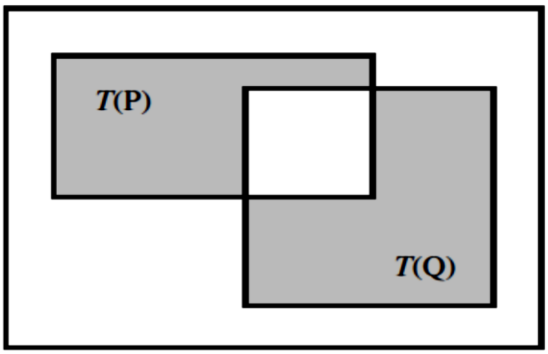


Figure 2.28: Exclusive or shown in black areas

The exclusive nor is the complement of the exclusive or [Mano (1988)]. A look at its truth table, Table 2.5.5, shows that it is an equivalence operation, i.e.,

Table 2.9: Truth table for exclusive nor

P	Q	$\overline{\text{P } XOR \text{ Q}}$
1	1	1
1	0	0
0	1	0
0	0	1

$$\overline{P\ XOR\ Q} \leftrightarrow (P \leftrightarrow Q)$$

and, hence, it is graphically equivalent to the Venn diagram in Fig. 2.5.4.

Logical Proofs

Logic involves the use of inference in everyday life, as well as in mathematics. In the latter, we often want to prove theorems to form foundations for solution procedures. In natural language, if we are given some hypotheses it is often useful to make certain conclusions from them – the so-called process of inference (inferring new facts from established facts). In the terminology we have been using, we want to know if the proposition $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \to Q$ is true. That is, is the statement a tautology?

The process works as follows. First, the linguistic statement (compound proposition) is made. Second, the statement is decomposed into its respective single propositions. Third, the statement is expressed algebraically with all pertinent logical connectives in place. Fourth, a truth table is used to establish the veracity of the statement.

Example 2.5.5:

Hypotheses: Engineers are mathematicians. Logical thinkers do not believe in magic. Mathematicians are logical thinkers.

Conclusion: Engineers do not believe in magic.

Let us decompose this information into individual propositions.

P: a person is an engineer

Q: a person is a mathematician

R : a person is a logical thinker

S : a person believes in magic

The statements can now be expressed as algebraic propositions as

$$\big((P \to Q) \wedge (R \to \overline{S}) \wedge (Q \to R)\big) \to (P \to \overline{S})$$

It can be shown that this compound proposition is a tautology.

Sometimes it might be difficult to prove a proposition by a direct proof (i.e., verify that it is true), so an alternative is to use an indirect proof. For example, the popular proof by contradiction (reductio ad absurdum) exploits the fact that $P \to Q$ is true if and only if $P \wedge \overline{Q}$ is false. Hence, if we want to prove that the compound statement $(P_1 \wedge P_2 \wedge \ldots \wedge P_n) \to Q$ is a tautology, we can alternatively show that the alternative statement $P_1 \wedge P_2 \wedge \ldots \wedge P_n \wedge \overline{Q}$ is a contradiction.

Example 2.5.6:

Hypotheses: If an arch-dam fails, the failure is due to a poor sub grade. An arch-dam fails.

Conclusion: The arch-dam failed because of a poor sub grade.

This information can be shown to be algebraically equivalent to the expression

$$\left((P \to Q) \wedge P\right) \to Q$$

To prove this by contradiction, we need to show that the algebraic expression

$$\left((P \to Q) \wedge P \wedge \overline{Q}\right)$$

is a contradiction. We can do this by constructing the truth table in Table 5.6. Recall that a contradiction is indicated when the last column of a truth table is filled with zeros.

Table 2.10: Truth table for dam failure problem

P	**Q**	$\overline{\mathbf{P}}$	$\overline{\mathbf{Q}}$	$\overline{\mathbf{P}} \vee \mathbf{Q}$	$(\overline{\mathbf{P}} \vee \mathbf{Q}) \wedge \mathbf{P} \wedge \overline{\mathbf{Q}}$
0	0	1	1	1	0
0	1	1	0	1	0
1	0	0	1	0	0
1	1	0	0	1	0

Deductive Inferences

The modus ponens deduction is used as a tool for making inferences in rule-based systems. A typical if–then rule is used to determine whether an antecedent (cause or action) infers a consequent (effect or reaction). Suppose we have a rule of the form IF A, THEN B, where A is a set defined on universe X and B is a set defined on universe Y. As discussed before, this rule can be translated into a relation between sets A and B; that is, recalling (2.5.8), $R = (A \times B) \cup (\overline{A} \times Y)$. Now suppose a new antecedent, say A, is known. Can we use modus ponens deduction, (2.5.11), to infer a new consequent, say B, resulting from the new antecedent? That is, can we deduce, in rule form, IF A, THEN B? The answer, of course, is yes, through the use of the composition operation. Since "A implies B" is defined on the

Cartesian space $X \times Y$, B can be found through the following set-theoretic formulation, again from (2.5.8):

$$B = A^{'} \circ R = A^{'} \circ \left((A \times B) \cup (\overline{A} \times Y)\right)$$

where the symbol $\circ$ denotes the composition operation. Modus ponens deduction can also be used for the compound rule IF A, THEN B, ELSE C, where this compound rule is equivalent to the relation defined in (2.5.10) as $R = (A \times B) \cup (\overline{A} \times C)$. For this compound rule, if we define another antecedent A, the following possibilities exist, depending on (1) whether A is fully contained in the original antecedent A, (2) whether A is contained only in the complement of A, or (3) whether A and A overlap to some extent as described next:

$$IF\ \ A^{'} \subset A, THEN\ \ y = B;\ IF\ \ A^{'} \subset \overline{A}, THEN\ \ y = C$$

$$IF\ \ A^{'} \cap A \neq \phi,\ A^{'} \cap \overline{A} \neq \phi,\ THEN\ \ y = B \cup C$$

The rule IF A, THEN B (proposition P is defined on set A in universe X, and proposition Q is defined on set B in universe Y), i.e., $(P \rightarrow Q) = R = (A \times B) \cup (\overline{A} \times Y)$, is then defined in function-theoretic terms as

$$\chi_R(x,\ y) = \max\left[\left(\chi_A(x) \wedge \chi_B(y)\right),\ \left((1 - \chi_A(x)) \wedge 1\right)\right] \quad (2.5.13)$$

Example 2.5.7: Suppose we have two universes of discourse for a heat exchanger problem described by the following collection of elements, X = {1, 2, 3, 4} and Y = {1, 2, 3, 4, 5, 6}.

Suppose X is a universe of normalized temperatures and Y is a universe of normalized pressures. Define crisp set A on universe X and crisp set B on universe Y as follows: A = {2, 3} and B = {3, 4}. The deductive inference IF A, THEN B (i.e., IF temperature is A, THEN pressure is B) will yield a matrix describing the membership values of the relation R, i.e., $\chi_R(x,\ y)$ through the use of (2.5.13). That is, the matrix R represents the rule IF A, THEN B as a matrix of characteristic (crisp membership) values.

Crisp sets A and B can be written using Zadeh's notation,

$$A = \left\{\frac{0}{1} + \frac{1}{2} + \frac{1}{3} + \frac{0}{4}\right\},\quad B = \left\{\frac{0}{1} + \frac{0}{2} + \frac{1}{3} + \frac{1}{4} + \frac{0}{5} + \frac{0}{6}\right\}.$$

If we treat set A as a column vector and set B as a row vector, the following matrix results from the Cartesian product of $A \times B$ is

$$A \times B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The Cartesian product A × Y can be determined by arranging A as a column vector and the universe Y as a row vector (sets A and Y can be written using Zadeh's notation),

$$\overline{A} = \left\{ \frac{1}{1} + \frac{0}{2} + \frac{0}{3} + \frac{1}{4} \right\}, \quad Y = \left\{ \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} \right\}.$$

$$\overline{A} \times Y = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Then the full relation R describing the implication IF A, THEN B is the maximum of the two matrices $A \times B$ and $\overline{A} \times Y$ or, using (2.5.13),

$$R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \!\!\! \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}.$$

The compound rule IF A, THEN B, ELSE C can also be defined in terms of a matrix relation as $R = (A \times B) \cup (\overline{A} \times C) \Rightarrow (P \to Q) \wedge (\overline{P} \to S)$, as given by (2.5.9) and (2.5.10), where the membership function is determined as

$$\chi_R(x,\ y) = \max\left[\left(\chi_A(x) \wedge \chi_B(y) \right),\ \left((1 - \chi_A(x)) \wedge \chi_C(y) \right) \right] \quad (2.5.14)$$

Example 2.5.8: Continuing with the previous heat exchanger example, suppose we define a crisp set C on the universe of normalized temperatures Y as C = {5, 6}, or, using Zadeh's notation,

$$C = \left\{ \frac{0}{1} + \frac{0}{2} + \frac{0}{3} + \frac{0}{4} + \frac{1}{5} + \frac{1}{6} \right\}.$$

The deductive inference IF A, THEN B, ELSE C (i.e., IF pressure is A, THEN temperature is B, ELSE temperature is C) will yield a relational matrix R, with characteristic values $\chi_R(x, y)$ obtained using Eq. (2.5.14). The first half of the expression in Eq. (5.10) (i.e., A × B) has already been determined in the previous example. The Cartesian product $\overline{A} \times C$ can be determined by arranging the set $\overline{A}$ as a column vector and the set C as a row vector

$$\overline{A} \times C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Then the full relation R describing the implication IF A, THEN B, ELSE C is the maximum of the two matrices A × B and $\overline{A} \times C$,

$$R = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}.$$

2.5.3 Fuzzy Logic

A fuzzy logic proposition, P, is a statement involving some concept without clearly defined boundaries. Linguistic statements that tend to express subjective ideas and that can be interpreted slightly differently by various individuals typically involve fuzzy propositions. Most natural language is fuzzy, in that it involves vague and imprecise terms. Statements describing a person's height or weight or assessments of people's preferences about colors or menus can be used as examples of fuzzy propositions. The truth value assigned to P can be any value on the interval [0, 1]. The assignment of the truth value to a proposition is actually a mapping from the interval [0, 1] to the universe U of truth values, T, as indicated in (2.5.15),

$$T : u \in U \rightarrow (0, 1) \quad (2.5.15)$$

As in classical binary logic, we assign a logical proposition to a set in the universe of discourse. Fuzzy propositions are assigned to fuzzy sets. Suppose proposition P is assigned to fuzzy set A; then the truth value of a proposition, denoted $T(P)$ is given by

$$T(P) = \mu_A(x), \quad \text{where } 0 \leq \mu_A(x) \leq 1 \quad (2.5.16)$$

(2.5.16) indicates that the degree of truth for the proposition $P : x \in A$ is equal to the membership grade of x in the fuzzy set A.

The logical connectives of negation, disjunction, conjunction, and implication are also defined for a fuzzy logic. These connectives are given in (2.5.17)–(2.5.20) for two simple propositions: proposition P defined on fuzzy set A and proposition Q defined on fuzzy set B.

Negation $T(\overline{P}) = 1 - T(P)$ (2.5.17)

Disjunction $P \vee Q : x$ is A or B $\left[i.e., \ T(P \vee Q) = \max\left(T(P), T(Q)\right) \right]$ (2.5.18)

Conjunction $P \wedge Q : x$ is A and B $\left[i.e., \ T(P \wedge Q) = \min\left(T(P), T(Q)\right) \right]$ (2.5.19)

Implication [Zadeh (1973)]

$P \to Q$: x is A, then x is B $\left[i.e., \ T(P \to Q) = T\left(\overline{P} \vee Q\right) = \max(T(\overline{P}), T(Q)) \right]$ (2.5.20)

As before in binary logic, the implication connective can be modelled in rule-based form; $P \to Q$ is, IF x is A, THEN y is B and it is equivalent to the following fuzzy relation, $R = (A \times B) \cup (\overline{A} \times Y)$, just as it is in classical logic. The membership function of R is expressed by the following formula:

$$\chi_R(x, \ y) = \max\left[\left(\chi_A(x) \wedge \chi_B(y) \right), \ (1 - \chi_A(x)) \right] \quad (2.5.21)$$

Example 2.5.9: Suppose we are evaluating a new invention to determine its commercial potential. We will use two metrics to make our decisions regarding the innovation of the idea. Our metrics are the ‘‘uniqueness’’ of the invention, denoted by a universe of novelty scales, X = {1, 2, 3, 4}, and the ‘‘market size’’ of the invention’s commercial market, denoted on a universe of scaled market sizes, Y = {1, 2, 3, 4, 5, 6}. In both universes the lowest numbers are the ‘‘highest uniqueness’’ and the ‘‘largest market,’’ respectively. A new invention in your group, say a compressible liquid of very useful temperature and viscosity conditions, has just received scores of ‘‘medium uniqueness,’’ denoted by fuzzy set A, and ‘‘medium market size,’’ denoted fuzzy set B. We wish to determine the implication of such a result, i.e., IF A, THEN B. We assign the invention the following fuzzy sets to represent its ratings:

$$A = \text{ medium uniqueness } = \left\{ \frac{0.6}{2} + \frac{1}{3} + \frac{0.2}{4} \right\}$$

$$B = \text{ medium market size } = \left\{ \frac{0.4}{2} + \frac{1}{3} + \frac{0.8}{4} + \frac{0.3}{5} \right\}$$

$$C = \text{ diffuse market size } = \left\{ \frac{0.3}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.6}{4} + \frac{0.5}{5} + \frac{0.3}{6} \right\}$$

The following matrices are then determined in developing the membership function of the implication, $\mu_R(x, \ y)$ illustrated in (2.5.21),

$$A \times B = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0.4 & 0.6 & 0.6 & 0.3 & 0 \\ 3 & 0 & 0.4 & 1 & 0.8 & 0.3 & 0 \\ 4 & 0 & 0.2 & 0.2 & 0.2 & 0.2 & 0 \end{array},$$

$$A \times B = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \end{array},$$

and finally, $R = \max\left(A \times B,\ \overline{A} \times Y\right)$

$$R = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0.4 & 0.4 & 0.6 & 0.6 & 0.4 & 0.4 \\ 3 & 0 & 0.4 & 1 & 0.8 & 0.3 & 0 \\ 4 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \end{array}.$$

When the logical conditional implication is of the compound form

$$IF\ \ x \text{ is } A,\ THEN\ \ y \text{ is } B,\ ELSE\ \ y \text{ is } C$$

then the equivalent fuzzy relation, R, is expressed as $R = (A \times B) \cup (\overline{A} \times C),$ in a form just as (2.5.10), whose membership function is expressed by the following formula:

$$\chi_R(x,\ y) = \max\left[\left(\chi_A(x) \wedge \chi_B(y)\right),\ \left((1 - \chi_A(x)) \wedge \chi_C(y)\right)\right] \tag{2.5.22}$$

Hence, using the result of (2.5.22), the new relation is

$$R = (A \times B) \cup (\overline{A} \times C) : \overline{A} \times C = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0.3 & 0.5 & 0.6 & 0.6 & 0.5 & 0.3 \\ 2 & 0.3 & 0.4 & 0.4 & 0.4 & 0.4 & 0.3 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0.3 & 0.5 & 0.6 & 0.6 & 0.5 & 0.3 \end{array}$$

and finally,

$$R = \begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0.3 & 0.5 & 0.6 & 0.6 & 0.5 & 0.3 \\ 0.3 & 0.4 & 0.6 & 0.6 & 0.4 & 0.3 \\ 0 & 0.4 & 1 & 0.8 & 0.3 & 0 \\ 0.3 & 0.5 & 0.6 & 0.6 & 0.5 & 0.3 \end{bmatrix} \end{array}.$$

2.5.4 Approximate Reasoning

The ultimate goal of fuzzy logic is to form the theoretical foundation for reasoning about imprecise propositions; such reasoning has been referred to as approximate reasoning [Zadeh (1976, 1979)]. Approximate reasoning is analogous to classical logic for reasoning with precise propositions, and hence is an extension of classical propositional calculus that deals with partial truths.

Suppose we have a rule-based format to represent fuzzy information. These rules are expressed in conventional antecedent-consequent form, such as

Rule 1: IF x is A, THEN y is B, where A and B represent fuzzy propositions (sets).

Now suppose we introduce a new antecedent, say A' and we consider the following rule:

Rule 2: IF x is A', THEN y is B'.

From information derived from Rule 1, is it possible to derive the consequent in Rule 2, B'? The answer is yes, and the procedure is fuzzy composition. The consequent B' can be found from the composition operation, $B' = A' \circ R$.

The two most common forms of the composition operator are the max–min and the max–product compositions.

Example 2.5.10: Suppose that the fuzzy relation just developed, i.e., R describes the invention's commercial potential. We wish to know what market size would be associated with a uniqueness score of ‘‘almost high uniqueness.’’ That is, with a new antecedent A the following consequent B can be determined using composition. Let

$$A' = \text{almost high uniqueness} = \left\{ \frac{0.5}{1} + \frac{1}{2} + \frac{0.3}{3} + \frac{0}{4} \right\}$$

Then, using the following max–min composition,

$$B' = A' \circ R = \left\{ \frac{0.5}{1} + \frac{0.5}{2} + \frac{0.6}{3} + \frac{0.6}{4} + \frac{0.5}{5} + \frac{0.5}{6} \right\}$$

we get the fuzzy set describing the associated market size. In other words, the consequent is fairly diffuse, where there is no strong (or weak) membership value for any of the market size scores (i.e., no membership values near 0 or 1).

This power of fuzzy logic and approximate reasoning to assess qualitative knowledge can be illustrated in more familiar terms to engineers in the context of the following example in the field of biophysics.

Example 2.5.11: For research on the human visual system, it is sometimes necessary to characterize the strength of response to a visual stimulus based on a magnetic field measurement or on an electrical potential measurement. When using magnetic field measurements, a typical experiment will require nearly 100 off/on presentations of the stimulus at one location to obtain useful data. If the researcher is attempting tomap the visual cortex of the brain, several stimulus locations must be used in the experiments. When working with a new subject, a researcher will make preliminary measurements to determine if the type of stimulus being used evokes a good response in the subject. The magnetic measurements are in units of femtotesla (10^{-15} tesla). Therefore, the inputs and outputs are both measured in terms of magnetic units.

We will define inputs on the universe X = [0, 50, 100, 150, 200] femtotesla, and outputs on the universe Y = [0, 50, 100, 150, 200] femtotesla. We will define two fuzzy sets, two different stimuli, on universe X:

$$W = \text{`` weak stimulus''} = \left\{\frac{1}{0} + \frac{0.9}{50} + \frac{0.3}{100} + \frac{0}{150} + \frac{0}{200}\right\} \subset X$$

$$M = \text{`` medium stimulus''} = \left\{\frac{0}{0} + \frac{0.4}{50} + \frac{1}{100} + \frac{0.4}{150} + \frac{0}{200}\right\} \subset X$$

and one fuzzy set on the output universe Y,

$$S = \text{`` severe stimulus''} = \left\{\frac{0}{0} + \frac{0}{50} + \frac{0.5}{100} + \frac{0.9}{150} + \frac{1}{200}\right\} \subset Y$$

The complement of S will then be

$$\overline{S} = \left\{\frac{1}{0} + \frac{1}{50} + \frac{0.5}{100} + \frac{0.1}{150} + \frac{0}{200}\right\}.$$

We will construct the proposition: IF ‘‘weak stimulus’’ THEN not ‘‘severe response,’’ using classical implication.

$$IF\ W\ THEN\ \overline{S} = W \rightarrow \overline{S} = (W \times \overline{S}) \cup (\overline{W} \times Y)$$

$$W \times \overline{S} = \begin{bmatrix} 1 \\ 0.9 \\ 0.3 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0.5 & 0.1 & 0 \end{bmatrix} = \begin{array}{c} \\ 0 \\ 50 \\ 100 \\ 150 \\ 200 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 50 & 100 & 150 & 200 \end{array} \\ \begin{bmatrix} 1 & 1 & 0.5 & 0.1 & 0 \\ 0.9 & 0.9 & 0.5 & 0.1 & 0 \\ 0.3 & 0.3 & 0.3 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\overline{W} \times Y = \begin{bmatrix} 0 \\ 0.1 \\ 0.7 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{array}{c} \\ 0 \\ 50 \\ 100 \\ 150 \\ 200 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 50 & 100 & 150 & 200 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

$$R = \left(W \times \overline{S}\right) \cup \left(\overline{W} \times Y\right) = \begin{array}{c} \\ 0 \\ 50 \\ 100 \\ 150 \\ 200 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 50 & 100 & 150 & 200 \end{array} \\ \begin{bmatrix} 1 & 1 & 0.5 & 0.1 & 0 \\ 0.9 & 0.9 & 0.5 & 0.1 & 0.1 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

This relation R, then, expresses the knowledge embedded in the rule: IF ‘‘weak stimuli’’ THEN not ‘‘severe response.’’ Now, using a new antecedent (IF part) for the input, M= ‘‘medium stimuli,’’ and a max–min composition we can find another response on the Y universe to relate approximately to the new stimulus M, i.e., to find $M \circ R$:

$$M \circ R = \begin{bmatrix} 0 & 0.4 & 1 & 0.4 & 0 \end{bmatrix} \begin{array}{c} \begin{array}{ccccc} 0 & 50 & 100 & 150 & 200 \end{array} \\ \begin{bmatrix} 1 & 1 & 0.5 & 0.1 & 0 \\ 0.9 & 0.9 & 0.5 & 0.1 & 0.1 \\ 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} = \begin{bmatrix} 0.7 & 0.7 & 0.7 & 0.7 & 0.7 \end{bmatrix}.$$

This result might be labelled linguistically as ‘‘no measurable response.’’

An interesting issue in approximate reasoning is the idea of an inverse relationship between fuzzy antecedents and fuzzy consequences arising from the composition operation.

Consider the following problem. Suppose we use the original antecedent A in the fuzzy composition. Do we get the original fuzzy consequent B as a result of the operation? Does the composition operation have a unique inverse, i.e., $B = A \circ R$? The answer is an unqualified no, and one should not expect an inverse to exist for fuzzy composition.

Example 2.5.12: Again, continuing with the invention example, Examples 2.5.9 and 2.5.10, suppose that $A' = A =$ ‘‘medium uniqueness.’’ Then

$$B' = A' \circ R = A \circ R = \left\{ \frac{0.4}{1} + \frac{0.4}{2} + \frac{1}{3} + \frac{0.8}{4} + \frac{0.4}{5} + \frac{0.4}{6} \right\} \neq B$$

That is, the new consequent does not yield the original consequent (B = medium market size) because the inverse is not guaranteed with fuzzy composition.

In classical binary logic this inverse does exist; that is, crisp modus ponens would give

$$B' = A' \circ R = A \circ R = B$$

where the sets A and B are crisp, and the relation R is also crisp. In the case of approximate reasoning, the fuzzy inference is not precise but rather is approximate. However, the inference does represent an approximate linguistic characteristic of the relation between two universes of discourse, X and Y.

Example 2.5.13: Suppose you are a soils engineer and you wish to track the movement of soil particles under applied loading in an experimental apparatus that allows viewing of the soil motion. You are building pattern recognition software to enable a computer to monitor and detect the motions. However, there are some difficulties in ‘‘teaching’’ your software to view the motion. The tracked particle can be occluded by another particle. The occlusion can occur when a tracked particle is behind another particle, behind a mark on the camera’s lens, or partially out of sight of the camera. We want to establish a relationship between particle occlusion, which is a poorly known phenomenon, and lens occlusion, which is quite well-known in photography. Let these membership functions,

$$A = \left\{ \frac{0.1}{x_1} + \frac{0.9}{x_2} + \frac{0}{x_3} \right\} \text{ and } B = \left\{ \frac{0}{y_1} + \frac{1}{y_2} + \frac{0}{y_3} \right\}$$

describe fuzzy sets for a tracked particle moderately occluded behind another particle and a lens mark associated with moderate image quality, respectively. Fuzzy set A is defined on a universe $X = \{x_1, x_2, x_3\}$ of tracked particle indicators, and fuzzy set B (note in this case that B is a crisp singleton) is defined on a universe $Y = \{y_1, y_2, y_3\}$ of lens obstruction indices. A typical rule might be: IF occlusion due to particle occlusion is moderate, THEN image quality will be similar to a moderate lens obstruction, or symbolically,

$$IF\ \ x \text{ is } A,\ THEN\ \ y \text{ is } B \left[i.e.,\ (A \times B) \cup (\overline{A} \times Y) = R \right]$$

We can find the relation, R, as follows:

$$A \times B = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_1 & 0 & 0.1 & 0 \\ x_2 & 0 & 0.9 & 0 \\ x_3 & 0 & 0 & 0 \end{array} \quad \text{and} \quad \bar{A} \times Y = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_1 & 0.9 & 0.9 & 0.9 \\ x_2 & 0.1 & 0.1 & 0.1 \\ x_3 & 1 & 1 & 1 \end{array}$$

$$R = (A \times B) \cup (\bar{A} \times Y) = \begin{array}{c|ccc} & y_1 & y_2 & y_3 \\ \hline x_1 & 0.9 & 0.9 & 0.9 \\ x_2 & 0.1 & 0.1 & 0.1 \\ x_3 & 1 & 1 & 1 \end{array}$$

This relation expresses in matrix form all the knowledge embedded in the implication. Let A be a fuzzy set, in which a tracked particle is behind a particle with slightly more occlusion than the particle expressed in the original antecedent A given by

$$A' = \left\{ \frac{0.3}{x_1} + \frac{1}{x_2} + \frac{0}{x_3} \right\}$$

We can find the associated membership of the image quality using max–min composition. For example, approximate reasoning will provide

$IF\ \ x\ is\ A',\ THEN\ \ B' = A' \circ R$ and we get

$$B' = \begin{bmatrix} 0.3 & 1 & 0 \end{bmatrix} \circ \begin{bmatrix} 0.9 & 0.9 & 0.9 \\ 0.1 & 0.1 & 0.1 \\ 1 & 1 & 1 \end{bmatrix} = \left\{ \frac{0.3}{y_1} + \frac{0.9}{y_2} + \frac{0.3}{y_3} \right\}$$

This image quality B' is fuzzier than B as indicated by the former's membership function.

OTHER FORMS OF THE IMPLICATION OPERATION

There are other techniques for obtaining the fuzzy relation R based on the IF A, THEN B, or $R = A \rightarrow B$. These are known as fuzzy implication operations, and they are valid for all values of $x \in X$ and $y \in Y$. The following forms of the implication operator show different techniques for obtaining the membership function values of fuzzy relation R defined on the Cartesian product space $X \times Y$.

$$\mu_R(x,\ y) = \max\left[\mu_B(x),\ 1 - \mu_A(x)\right] \quad (2.5.23)$$

$$\mu_R(x,\ y) = \min\left[\mu_B(x),\ \mu_A(x)\right] \quad (2.5.24)$$

$$\mu_R(x,\ y) = \min\left\{1,\ \left[1 + \mu_B(x) - \mu_A(x)\right]\right\} \quad (2.5.25)$$

$$\mu_R(x,\ y) = \mu_B(x) \cdot \mu_A(x) \quad (2.5.26)$$

$$\mu_R(x,\ y) = \begin{cases} 1, & \text{for } \mu_A(x) \le \mu_B(x) \\ \mu_B(x), & \text{otherwise} \end{cases} \quad (2.5.27)$$

In situations where the universes are represented by discrete elements the fuzzy relation R is a matrix.

EXERCISE-2

1. Using the inference method, find the membership values of the triangular shapes for each of the following triangles: $(a)\ 60^\circ, 40^\circ, 80^\circ$ $(b)\ 45^\circ, 65^\circ, 70^\circ$ and $(c)\ 75^\circ, 55^\circ, 50^\circ$.
2. Using your own intuition, develop fuzzy number "approximately 4 or approximately 8" using the following function shapes:

   (1) Symmetric triangle

   (2) Trapezoids

   (3) Gaussian functions.
3. Develop membership function for trapezoidal similar to algorithm developed for triangle and the function should have two independent variables hence it can be passed. For the shown in table, show the first iteration in trying to compute the membership values for input variables x_1, x_2 and x_3 in the output regions R^1 and R^2

x_1	x_2	x_3	R^1	R^2
1.0	0.5	2.3	1.0	0.0

   (a) Use $3\times3\times1$ neural network,

   (b) Use $3\times3\times2$ neural network.
4. The three variables of interest in the MOSFET are the amount of current that can be switched, the voltage that can be switched and the cost. The following membership function for the transistor was developed

   Current $= I = \left\{\frac{0.4}{0.8}+\frac{0.7}{0.9}+\frac{1}{1}+\frac{0.8}{1.1}+\frac{0.6}{1.2}\right\}$,

   Voltage $= V = \left\{\frac{0.2}{30}+\frac{0.8}{45}+\frac{1}{60}+\frac{0.9}{75}+\frac{0.7}{90}\right\}$,

   Cost$==\left\{\frac{0.4}{0.5}+\frac{1}{0.6}+\frac{0.5}{0.7}\right\}$.

   The power is given by $P = V\,I$.

   (a) Find the fuzzy Cartesian product $P = V\times I$.

   (b) Find the fuzzy Cartesian product $T = I\times C$.

   (c) Using max–min composition find $E = P\circ T$.

   (d) Using max–product composition find $E = P\circ T$.

5. Find the relation between two fuzzy sets R_1 and R_2 using

(a) Max–min composition

(b) Max–product composition

(c) Max–average composition

$$R_1 = \begin{array}{c} \\ x_1 \\ x_2 \end{array}\begin{array}{c} \begin{array}{cccc} y_1 & y_2 & y_3 & y_4 \end{array} \\ \begin{bmatrix} 0.3 & 0.1 & 0.6 & 0.3 \\ 0.1 & 1 & 0.2 & 0.1 \end{bmatrix} \end{array},$$

$$R_2 = \begin{array}{c} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{array}\begin{array}{c} \begin{array}{ccc} z_1 & z_2 & z_3 \end{array} \\ \begin{bmatrix} 0.9 & 0.1 & 1 \\ 0.1 & 0.5 & 0.4 \\ 0.6 & 0.8 & 0.5 \\ 0.1 & 0 & 0 \end{bmatrix} \end{array}.$$

6. Discuss the reflexivity property of the following fuzzy relation

$$R = \begin{bmatrix} 1 & 0.7 & 0.3 \\ 0.4 & 0.5 & 0.8 \\ 0.7 & 0.5 & 1 \end{bmatrix}.$$

7. For each of the following relation on single set state whether the relation is reflexive, symmetric, and transitive
   (a) "is a sibling of,"
   (b) "is a parent of,"
   (c) "is a smarter than,"
   (d) "is the same height as,"
   (e) "is at least as tall as."
8. A factory process control operation involves two linguistic (atomic) parameters consisting of pressure and temperature in a fluid delivery system. Nominal pressure limits range from 400 psi minimum to 1000 psi maximum. Nominal temperature limits are 130 to 140◦F. We characterize each parameter in fuzzy linguistic terms as follows:

$$\text{Low temperature } = \left\{\frac{1}{131}+\frac{0.8}{132}+\frac{0.6}{133}+\frac{0.4}{134}+\frac{0.2}{135}+\frac{0}{136}\right\}$$

$$\text{High temperature } = \left\{\frac{0}{134}+\frac{0.2}{135}+\frac{0.4}{136}+\frac{0.6}{137}+\frac{0.8}{138}+\frac{1}{139}\right\}$$

$$\text{High pressure } = \left\{\frac{0}{400}+\frac{0.2}{600}+\frac{0.4}{700}+\frac{0.6}{800}+\frac{0.8}{900}+\frac{1}{1000}\right\}$$

$$\text{Low pressure } = \left\{ \frac{1}{400} + \frac{0.8}{600} + \frac{0.6}{700} + \frac{0.4}{800} + \frac{0.2}{900} + \frac{0}{1000} \right\}$$

(a) Find the following membership functions:

(i) Temperature not very low

(ii) Temperature not very high

(iii) Temperature not very low and not very high

(b) Find the following membership functions:

(i) Pressure slightly high

(ii) Pressure fairly high $\left([\text{high}]^{2/3}\right)$

(iii) Pressure not very low or fairly low

9. Show that the following propositions from Lewis Carroll are tautologies [Gill, 1976]:

(a) No ducks waltz; no officers ever decline to waltz; all my poultry are ducks. Therefore, none of my poultry are officers.

(b) Babies are illogical; despised persons cannot manage crocodiles; illogical persons are despised; therefore, babies cannot manage crocodiles.

(c) Promise-breakers are untrustworthy; wine-drinkers are very communicative; a man who keeps his promise is honest; all pawnbrokers are wine-drinkers; we can always trust a very communicative person; therefore, all pawnbrokers are honest. (This problem requires $2^6 = 64$ lines of a truth table; perhaps it should be tackled with a computer.)

10. Given the fuzzy sets A and B on X and Y, respectively,

$$A = \int \left\{ \frac{1 - 0.1x}{x} \right\}, \quad \text{for } x \in [0, 10] \text{ and } B = \int \left\{ \frac{0.2y}{y} \right\}, \quad \text{for } y \in [0, 5]$$

$\mu_A(x) = 0$ outside the interval $[0, 10]$ and $\mu_B(x) = 0$ outside the interval $[0, 5]$.

(a) Construct a fuzzy relation R for the implication $A \rightarrow B$ using the classical implication operation, i.e., construct $R = \left(A \times B\right) \cup \left(\overline{A} \times Y\right)$.

(b) Use max–min composition to find B', given $A' = \left\{ \frac{1}{3} \right\}$.

BIBLIOGRAPHY

1. Belohlavek, R., Klir, G., Lewis, H., and Way, E. (2002), On the capability of fuzzy set theory to represent concepts, Int. J. Gen. Syst., vol. 31, pp. 569–585.
2. Gill, A., Applied algebra for the computer sciences, Prentice Hall, Englewood Cliffs, NJ, 1976.
3. Klir, G. J. and Yuan, B., Fuzzy Sets and Fuzzy Logic-Theory and Applications, Prentice Hall, New Jersey, 1995.
4. Kosko, B., Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.
5. Lee, K. H., First Course on Fuzzy Theory and Applications, Springer, Berlin Heidelberg, NewYork, 2005.
6. Osherson, D. and Smith, E. (1981), On the adequacy of prototype theory as a theory of concepts, Cognition, vol. 9, pp. 35–58.
7. Rescher, N., Many-valued logic, McGraw-Hill, New York, 1969.
8. Ross, T., Fuzzy logic with engineering applications, McGraw-Hill, New York, 1995.
9. Sanchez, E. (1976), Resolution of composite fuzzy relation equations, Inf. Control, vol. 30, pp. 38–48.
10. Sivanandam, S. N., Sumathi, S. and Deepa, S. N., Introduction to Fuzzy Logic using MATLAB, Springer, Berlin Heidelberg, New York, 2007.
11. Zadeh, L. A. (1976), The concept of a linguistic variable and its application to approximate reasoning – Part 3, Inf. Sci., vol. 9, pp. 43–80.
12. Zadeh, L. A., A theory of approximate reasoning, in J. Hayes, D. Michie, and L. Mikulich (eds.), Machine Intelligence, Halstead Press, New York, pp. 149–194, 1979.
13. Zadeh, L. A. (1965), Fuzzy sets, Inform. Contr., vol. 8, pp. 338-353.
14. Zimmermann, H. J., Fuzzy set theory-its applications, Kluwer Academic publishers, USA, 1996.

Chapter-3

Fuzzy Systems and Fuzzy Models

3.1. DEFUZZIFICATION

Apart from the lambda cut sets and relations which convert fuzzy sets or relations into crisp sets or relations, there are other various defuzzification methods employed to convert the fuzzy quantities into crisp quantities. The output of an entire fuzzy process can be union of two or more fuzzy membership functions. To explain this in detail, consider a fuzzy output, which is formed by two parts, one part being triangular shape (Figure 3.1 (a)) and other part being trapezoidal (Figure 3.1 (b)). The union of these two forms (Figure 3.1 (c)) the outer envelope of the two shapes.

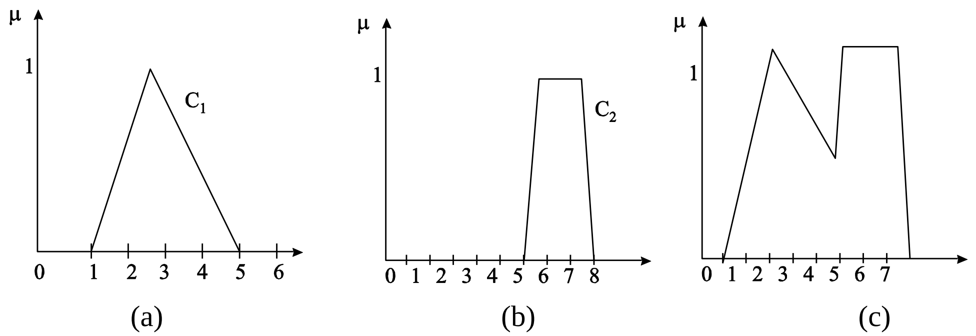


Figure 3.1: Typical fuzzy output

There are seven methods used for defuzzifying the fuzzy output functions:

(i) Max-membership principle

(ii) Centroid method

(iii) Weighted average method

(iv) Mean–max membership

(v) Centre of sums

(vi) Centre of largest area

(vii) First of maxima or last of maxima

Next, we explain each method with graph.

Max-membership principle

This method is given by the expression

$$\mu_C(z^*) \geq \mu_C(z), \text{ for all } z \in X. \tag{3.1.1}$$

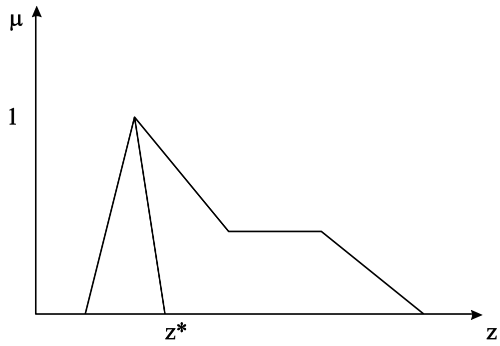


Figure 3.2: Max membership method

This method is also referred as height method. This is shown in Figure 3.2.

(i) Centroid method

This is the most widely used method. This can be called as centre of gravity or centre of area method. It can be defined by the algebraic expression

$$z^* = \int \frac{\mu_C(z) z dx}{\mu_C(z) dx}, \tag{3.1.2}$$

where $\int$ is used for algebraic integration. Fig. 3.3 represents this method graphically.

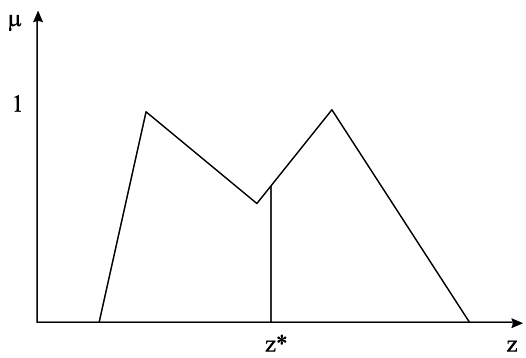


Figure 3.3: Centroid method

(ii) Weighted Average method

This method cannot be used for asymmetrical output membership functions, can be used only for symmetrical output membership functions. Weighting each membership function in the obtained output by its largest membership value forms this method. The evaluation expression for this method is

$$z^* = \frac{\sum \mu_C(\bar{z})\bar{z}}{\sum \mu_C(\bar{z})}, \tag{3.1.3}$$

where $\sum$ is used for algebraic sum.

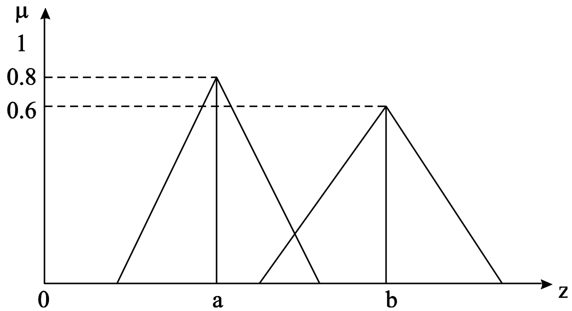


Figure 3.4: weighted average method

(iii) Mean-max membership

This method is related to max-membership principle, but the present of the maximum membership need not be unique, i.e., the maximum membership need not be a single point, it can be a range. This method is also called as middle of maxima method the expression is given as

$$z^* = \frac{a+b}{2}, \tag{3.1.4}$$

where a × b are the end point of the maximum membership range as shown in Fig. 3.5.

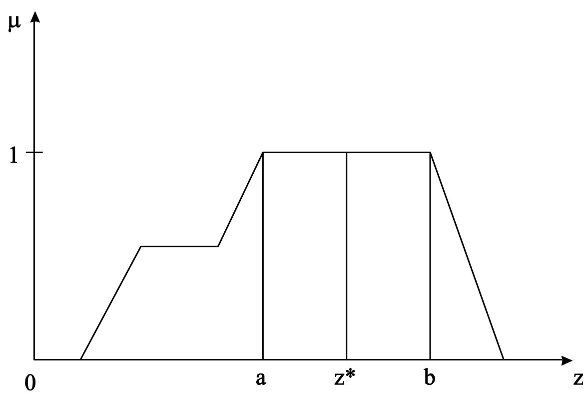


Figure 3.5: Mean-max membership

(iv) Centre of sums

It involves the algebraic sum of individual output fuzzy sets, say c_1 and c_2 instead of union. In this method, it is noted that the intersecting areas are added twice. This method is similar to the weighted average method, but in centre of sums, the weights are the areas of the respective membership functions whereas in the weighted average method, the weights are individual membership values.

The defuzzified value z^* is given as

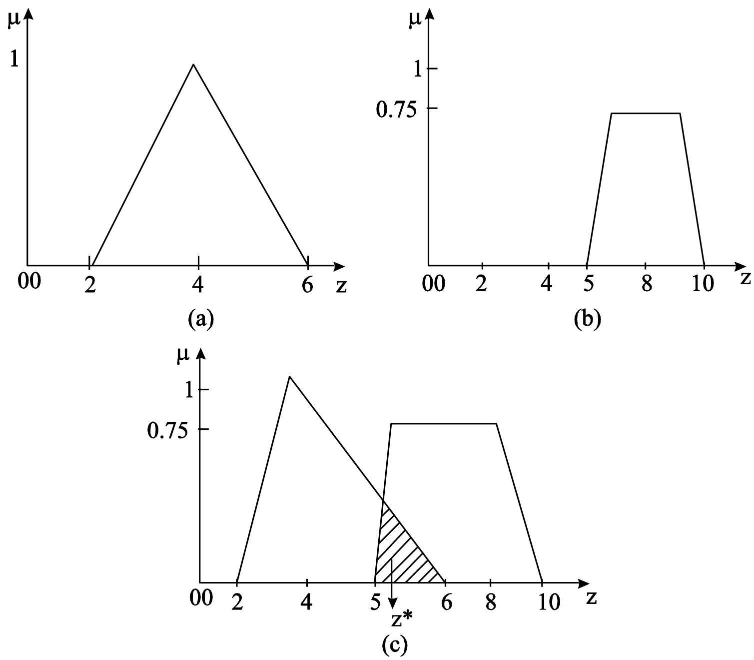


Figure 3.6: (a) First membership (b) Second membership (c) defuzzification step

$$z^* = \frac{\int x \sum_{k=1}^{n} \mu_{C_k}(z)dx}{\int x \sum_{k=1}^{n} \mu_{C_k}(z)dz}. \qquad (3.1.5)$$

(v) Centre of largest area

If the fuzzy set has two convex sub regions, then the entire of gravity of the convex sub region with the largest area can be used to calculate the defuzzification value. The equation is given as

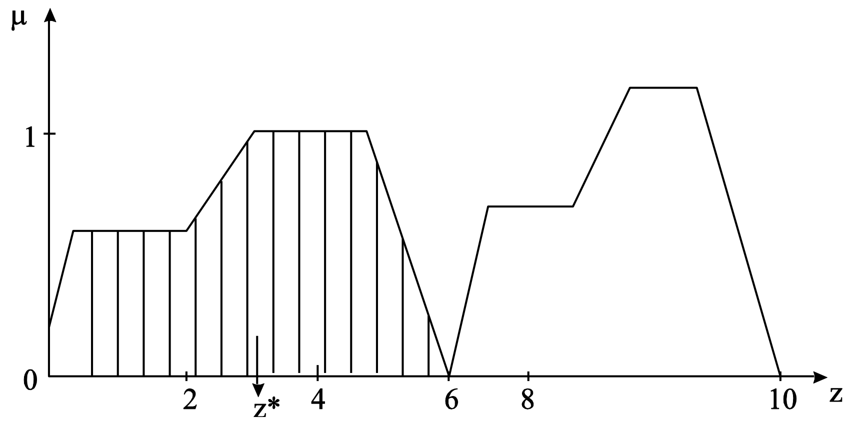


Figure 3.7: Centre of largest area

$$z^* = \frac{\int \mu_{C_m}(z)\, z\, dx}{\int \mu_{C_m}(z) dx}, \tag{3.1.6}$$

Where c_m is the convex region with largest area. The value z^* is same as the Value z^* obtained by centroid method. This can be done even for non-convex regions.

(vi) First of maxima or last of minima

Here, the computed output of all individual output fuzzy sets c_k is used to determine the smallest value, with maximized membership degree in c_m .

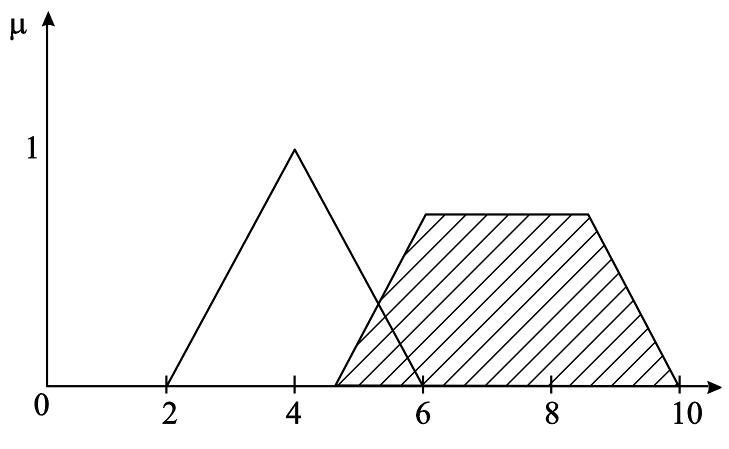


Figure 3.8: First of maxima or last of minima

Example 3.1.1: For the given membership function as shown in Fig. 3.8 determines the defuzzified output value by seven methods.

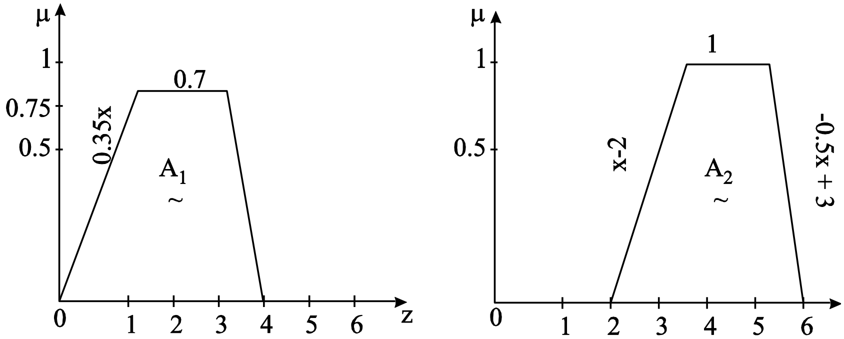


Figure 3.9: Membership function

Solution. (a) Centroid method

$$A_{11}: (0,0),\ (2,\ -0.7),$$

The straight line may be:

$$y-0=\frac{0.7}{2}(x-0),\ y=3.5x.$$

$$A_{12}: y=0.7.$$

$$A_{13}: \text{not needed.}$$

$$A_{21}: (2,0),\ (3,1),$$

$$y-0=\frac{0-1}{3-2}(x-2),\ y=x-2.$$

$$A_{12}: y=1.$$

$$A_{23}: (4,1),\ (6,0),$$

$$y-1=\frac{0-1}{6-4}(x-4),\ y=-0.5x+3.$$

Solving A_{12} and A_{21}, we have $x=2.7,\ \ y=0.7.$

$$\text{Numerator } = \int_0^2 0.35z^2 dz + \int_2^{2.7} 0.7K dz + \int_{2.7}^{3} \left(z^2-2z\right)dz + \int_3^4 z dz + \int_4^6 \left(-0.5z^2+3z\right)dz$$

$$=10.98.$$

$$\text{Denominator } = \int_0^2 0.35z^2 dz + \int_2^{2.7} 0.7K dz + \int_{2.7}^3 \left(z^2 - 2\right) dz + \int_3^4 dz + \int_4^6 \left(-0.5z^2 + 3z\right) dz$$

$$= 3.445.$$

$$z^* = \frac{\text{Numerator}}{\text{Denominator}} = \frac{10.98}{3.445} = 3.187.$$

(b) Weighted average method

$$z^* = \frac{2 \times 0.7 + 4 \times 1}{1 + 0.7} = 3.176.$$

(c) Mean–max method

$$z^* = \frac{2.5 + 3.5}{2} = 3.$$

(d) Center of sums method

$$z^* = \frac{\int_0^6 \left[\left(\frac{1}{2} \times 0.7 \times (3+2) \times 2\right) + \left(\frac{1}{2} \times 1 \times (2+4) \times 4\right)\right] dz}{\int_0^6 \left[\left(\frac{1}{2} \times 0.7 \times (3+2)\right) + \left(\frac{1}{2} \times 1 \times (2+4)\right)\right] dz}$$

$$z^* = \frac{\int_0^6 \left(3.5 + 12\right) dz}{\int_0^6 \left(1.75 + 3\right) dz} = 2.84.$$

(e) First of maxima

$$z^* = 3.$$

(f) First of maxima

$$z^* = 4.$$

(g) Centre of largest area

$$\text{Area of } I = \frac{1}{2} \times 0.7 \times (2.7 + 0.7) = 1.19,$$

$$\text{Area of } II = \frac{1}{2} \times 1 \times (2+3) \times \frac{1}{2} \times 0.7 = 2.255.$$

Area of II is larger, So,

$$z^* = \frac{\int_{2.7}^{3} \frac{1}{2} \times 0.3 \times 0.3 \times 2.85\, dz + \int_{3}^{4} 1 \times 1 \times 3.5\, dz + \int_{4}^{6} \frac{1}{2} \times 2 \times 5\, dz}{\int_{2.7}^{3} \frac{1}{2} \times 0.3 \times 0.3\, dz + \int_{3}^{4} 1 \times 1\, dz + \int_{4}^{6} \frac{1}{2} \times 2 \times 1\, dz}$$

$$z^* = \frac{\int_{2.7}^{3} 0.12825\, dz + \int_{3}^{4} 3.5\, dz + \int_{4}^{6} 5\, dz}{\int_{2.7}^{3} 0.045\, dz + \int_{3}^{4} dz + \int_{4}^{6} dz} = 4.49.$$

3.2 FUZZY SYSTEMS

Natural language is perhaps the most powerful form of conveying information that humans possess for any given problem or situation that requires solving or reasoning. This power has largely remained untapped in today's mathematical paradigms; however it can be handled with the utility of fuzzy logic. Consider the information contained in the passage above from Charles Dickens' A Tale of Two Cities. Imagine reducing this passage to a more precise form such that it could be assimilated by a binary computer. First, we will have to remove the fuzziness inherent in the passage limiting the statements to precise, either–or, Aristotelian logic. Consider the following crisp version of the first few words of the Dickens passage:

The time interval x was the period exhibiting a 100 percent maximum of possible values as measured along some arbitrary social scale and the interval x was also the period of time exhibiting a 100 percent minimum of these values as measured along the same scale [Clark, (1992)].

The crisp version of this passage has established an untenable paradox, identical to that posed by the excluded middle axioms in probability theory. Another example is available from the same classic, the last sentence in Dickens' A Tale of Two Cities: ''It is a far, far better thing that I do, than I have ever done; it is a far, far better rest that I go to, than I have ever known.'' It would also be difficult to address this original fuzzy phrase by an intelligent machine using binary logic. Both of these examples demonstrate the power of communication inherent in natural language, and they demonstrate how far we are from enabling intelligent machines to reason the way humans do – a long way!

Cognitive scientists tell us that humans base their thinking primarily on conceptual patterns and mental images rather than on any numerical quantities. In fact the expert system paradigm known as ''frames'' is based on the notion of a cognitive picture in one's mind. Furthermore, humans communicate with their own natural language by referring to previous mental images with rather vague but simple terms. Despite the vagueness and ambiguity in natural language, humans communicating in a common language have very little trouble in basic understanding. Our language has been termed the shell of our thoughts [Zadeh, 1975]. Hence, any attempts to model the human thought process as expressed in our communications with one another must be preceded by models that attempt to emulate our natural language.

Our natural language consists of fundamental terms characterized as atoms in the literature. A collection of these atoms will form the molecules, or phrases, of our natural language. The fundamental terms can be called atomic terms. Examples of some atomic terms are slow, medium, young, beautiful, etc. A collection of atomic terms is called a composite, or simply a set of terms. Examples of composite terms are very slow horse, medium-weight female, young tree, fairly beautiful painting, etc. Suppose we define the atomic terms and sets of atomic terms to exist as elements and sets on a universe of natural language terms, say universe X. Furthermore, let us define another universe, called Y, as a universe of cognitive interpretations, or meanings. Although it may seem straightforward to envision a universe of terms, it may be difficult to ponder a universe of interpretations. Consider this universe, however, to be a collection of individual elements and sets that represent the cognitive patterns and mental images referred to earlier in this chapter. Clearly, these interpretations would be rather vague, and they might best be represented as fuzzy sets. Hence, an atomic term, or as [Zadeh, (1975)] defines it, a linguistic variable, can be interpreted using fuzzy sets.

The need for expressing linguistic variables using the precepts of mathematics is quite well established. Leibniz, who was an early developer of calculus, once claimed, ‘‘If we could find characters or signs appropriate for expressing all our thoughts as definitely and as exactly as arithmetic expresses numbers or geometric analysis expresses lines, we could in all subjects, in so far as they are amenable to reasoning, accomplish what is done in arithmetic and geometry.’’ Fuzzy sets are a relatively new quantitative method to accomplish just what Leibniz had suggested.

With these definitions and foundations, we are now in a position to establish a formal model of linguistics using fuzzy sets. Suppose we define a specific atomic term in the universe of natural language, X, as element α and we define a fuzzy set A in the universe of interpretations, or meanings, Y, as a specific meaning for the term α. Then natural language can be expressed as a mapping M from a set of atomic terms in X to a corresponding set of interpretations defined on universe Y. Each atomic term α in X corresponds to a fuzzy set A in Y, which is the ‘‘interpretation’’ of α. This mapping, which can be denoted $M(\alpha, A)$ is shown schematically in Fig. 3.10.

The fuzzy set A represents the fuzziness in the mapping between an atomic term and its interpretation and can be denoted by the membership function $\mu_M(\alpha, y)$ or more simply by

$$\mu_M(\alpha, y) = \mu_A(y) \tag{3.2.1}$$

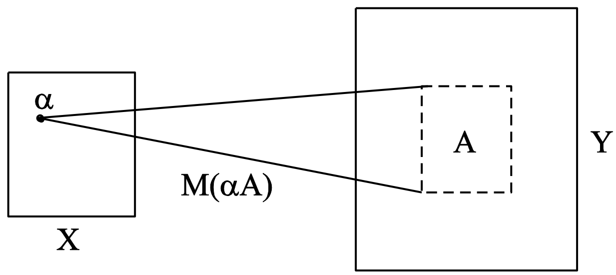


Figure 3.10: Mapping of a linguistic atom, α, to a cognitive interpretation A

As an example, suppose we have the atomic term ''young'' (α) and we want to interpret this linguistic atom in terms of age, y, by a membership function that expresses the term ''young.'' The membership function given here in the notation of [Zadeh,(1975b)] and labeled A, might be one interpretation of the term young expressed as a function of age,

$$A = "young" = \int_0^{25} \frac{1}{y} + \int_{25}^{100} \frac{1}{y}\left[1+\left(\frac{y-25}{5}\right)^2\right]^{-1}$$

or alternatively,

$$\mu_M(\text{young}, y) = \begin{cases} \left[1+\left(\dfrac{y-25}{5}\right)^2\right]^{-1}, & y > 25\,\text{years} \\ 1, & y \le 25\,\text{years} \end{cases}$$

Similarly, the atomic term ''old'' might be expressed as another fuzzy set, O, on the universe of interpretation, Y, as

$$\mu_M(\text{old}, y) = 1 - \left[1+\left(\frac{y-25}{5}\right)^2\right]^{-1}, \text{ for } 50 \le y \le 100.$$

On the basis of the foregoing, we can call α a natural language variable whose ''value'' is defined by the fuzzy set $\mu_\alpha(y)$ Hereinafter, the ''value'' of a linguistic variable will be synonymous with its interpretation.

As suggested before, a composite is a collection, or set of atomic terms combined by various linguistic connectives such as and, or, and not. Define two atomic terms α and β on the universe X. The interpretation of the composite, defined on universe Y can be defined by the following set-theoretic operations [Zadeh, (1975b)],

$$\alpha \text{ or } \beta : \mu_{\alpha \text{ or } \beta}(y) = \max\left(\mu_\alpha(y), \mu_\beta(y)\right), \quad (3.2.2)$$

$$\alpha \ and \ \beta: \ \mu_{\alpha \, and \, \beta}(y) = \min\left(\mu_\alpha(y), \ \mu_\beta(y)\right), \quad (3.2.3)$$

$$Not \, \alpha = \overline{\alpha} : \ \mu_{\overline{\alpha}}(y) = 1 - \mu_\alpha(y). \quad (3.2.4)$$

These operations are analogous to those proposed earlier in this chapter (standard fuzzy operations), where the natural language connectives and, or, and not were logical connectives. In linguistics, fundamental atomic terms are often modified with adjectives (nouns) or adverbs (verbs) like very, low, slight, more or less, fairly, slightly, almost, barely, mostly, roughly, approximately, and so many more that it would be difficult to list them all. We will call these modifiers ‘‘linguistic hedges’’: that is, the singular meaning of an atomic term is modified, or hedged, from its original interpretation. Using fuzzy sets as the calculus of interpretation, these linguistic hedges have the effect of modifying the membership function for a basic atomic term [Zadeh, (1972)]. As an example, let us look at the basic linguistic atom, α, and subject it to some hedges. Define $\alpha = \int_Y \mu_\alpha(y)/y$; then

$$"Very"\alpha = \alpha^2 = \int_Y \frac{[\mu_\alpha(y)]^2}{y}, \tag{3.2.5}$$

$$"Very, \ very"\alpha = \alpha^4, \tag{3.2.6}$$

$$"Plus"\alpha = \alpha^{1.25}, \tag{3.2.7}$$

$$"Slightly"\alpha = \sqrt{\alpha} = \int_Y \frac{[\mu_\alpha(y)]^{0.5}}{y}, \tag{3.2.8}$$

$$"Minus"\alpha = \alpha^{0.75}. \tag{3.2.9}$$

The expressions shown in (3.2.5)-(3.2.7) are linguistic hedges known as concentrations [Zadeh, 1972]. Concentrations tend to concentrate the elements of a fuzzy set by reducing the degree of membership of all elements that are only ‘‘partly’’ in the set. The less an element is in a set (i.e., the lower its original membership value), the more it is reduced in membership through concentration. For example, by using (3.2.5) for the hedge very, a membership value of 0.9 is reduced by 10% to a value of 0.81, but a membership value of 0.1 is reduced by an order of magnitude to 0.01. This decrease is simply a manifestation of the properties of the membership value itself; for $0 \le \mu \le 1$, then $\mu \ge \mu^2$. Alternatively, the expressions given in (3.2.8) and (3.2.9) are linguistic hedges known as dilations (or dilutions in some publications). Dilations stretch or dilate a fuzzy set by increasing the membership of elements that are ‘‘partly’’ in the set [Zadeh, 1972]. For example, using (3.2.8) for the hedge slightly, a membership value of 0.81 is increased by 11% to a value of 0.9, whereas a membership value of 0.01 is increased by an order of magnitude to 0.1.

Another operation on linguistic fuzzy sets is known as intensification. This operation acts in a combination of concentration and dilation. It increases the degree of membership of those elements in the set with original membership values greater than 0.5, and it decreases the degree of membership of those elements in the set with original membership values less than 0.5. This also has the effect of making the boundaries of the membership function steeper. Intensification can be expressed by numerous algorithms, one of which, proposed by Zadeh (1972) is

$$\text{"intensify"}\,\alpha = \begin{cases} 2\mu_\alpha^2(y), & \text{for } 0 \le \mu_\alpha(y) \le 1 \\ 1-2\left[1-\mu_\alpha(y)\right]^2, & \text{for } 0.5 \le \mu_\alpha(y) \le 1 \end{cases} \tag{3.2.10}$$

Intensification increases the contrast between the elements of the set that have more than half-membership and those that have less than half-membership. Fig. 3.11 (a) and (b), and 3.12 illustrate the operations of concentration, dilation, and intensification, respectively, for fuzzy linguistic hedges on a typical fuzzy set A.

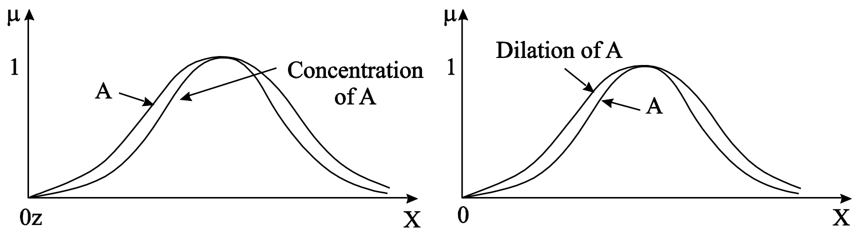


Figure 3.11: (a) Fuzzy concentration (b) Fuzzy dilation

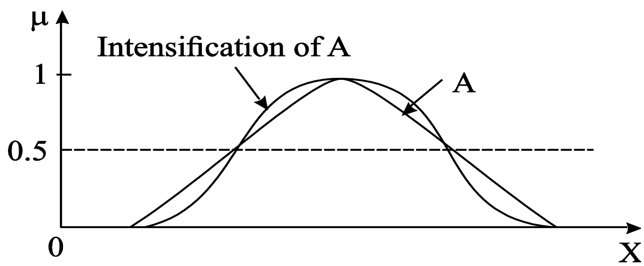


Figure 3.12: Fuzzy intensification

Composite terms can be formed from one or more combinations of atomic terms, logical connectives, and linguistic hedges. Since an atomic term is essentially a fuzzy mapping from the universe of terms to a universe of fuzzy sets represented by membership functions, the implementation of linguistic hedges and logical connectives is manifested as function-theoretic operations on the values of the membership functions. In order to conduct the function-theoretic operations, a precedence order must be established. For example, suppose we have two atomic terms ‘‘small’’ and ‘‘red,’’ and their associated membership functions, and we pose the following linguistic expression: a ‘‘not small’’ and ‘‘very red’’ fruit. Which of the operations, i.e., not, and, very, would we perform first, which would we perform second, and so on? In the literature, the following preference table (Table 3.1) has been suggested for standard Boolean operations.

Parentheses may be used to change the precedence order and ambiguities may be resolved by the use of association-to-the-right. For example, ‘‘plus very minus very small’’ should be interpreted as

$$\text{plus}\big(\text{very}\,(\text{minus}\,(\text{very}\,(\text{small})))\big)$$

Table 3.1: Precedence for linguistic hedges and logical operations

Precedence	Operation
First	Hedge. Not
Second	And
Third	Or

Every atomic term and every composite term has a syntax represented by its linguistic label and a semantics, or meaning (interpretation),which is given by a membership function. The use of a membership function gives the flexibility of an elastic meaning to a linguistic term. On the basis of this elasticity and flexibility, it is possible to incorporate subjectivity and bias into the meaning of a linguistic term. These are some of the most important benefits of using fuzzy mathematics in the modelling of linguistic variables. This capability allows us to encode and automate human knowledge, which is often expressed in natural language propositions.

In our example, a ‘‘not small’’ and ‘‘very red’’ fruit, we would perform the hedges ‘‘not small’’ and ‘‘very red’’ first, then we would perform the logical operation and on the two phrases as suggested in Table 3.2.1. To further illustrate Table 3.2.1, consider the following numerical example.

Example 3.2.1: Suppose we have a universe of integer, $Y=\{1,\ 2,\ 3,\ 4,\ 5\}$. We define the following linguistic terms as a mapping onto Y:

$$"Small"=\left\{\frac{1}{1}+\frac{0.8}{2}+\frac{0.6}{3}+\frac{0.4}{4}+\frac{0.2}{5}\right\} \text{ and } "\text{Large}"=\left\{\frac{0.2}{1}+\frac{0.4}{2}+\frac{0.6}{3}+\frac{0.8}{4}+\frac{1}{5}\right\}$$

Now we modify these two linguistic terms with hedges,

$$"Very\ small"="small"^2=\left\{\frac{1}{1}+\frac{0.64}{2}+\frac{0.36}{3}+\frac{0.16}{4}+\frac{0.04}{5}\right\}$$

$$"Not\,very\ small"=1-"Very\ small"="small"^2=\left\{\frac{0}{1}+\frac{0.36}{2}+\frac{0.64}{3}+\frac{0.84}{4}+\frac{0.96}{5}\right\}.$$

Then we construct a phrase or a composite term:

$$\alpha="\text{not very small and not very large}"$$

which involves the following set-theoretic operations:

$$\alpha = \left\{ \frac{0}{1} + \frac{0.36}{2} + \frac{0.64}{3} + \frac{0.84}{4} + \frac{0.96}{5} \right\} \cap \left\{ \frac{1}{1} + \frac{1}{2} + \frac{0.9}{3} + \frac{0.6}{4} \right\} = \left\{ \frac{0.36}{2} + \frac{0.64}{3} + \frac{0.6}{4} \right\}.$$

In summary, the foregoing material introduces the idea of a linguistic variable (atomic term), which is a variable whose values (interpretation) are natural language expressions referring to the contextual semantics of the variable. Zadeh [Zadeh (1975b)] described this notion quite well:

A linguistic variable differs from a numerical variable in that its values are not numbers but words or sentences in a natural or artificial language. Since words, in general, are less precise than numbers, the concept of a linguistic variable serves the purpose of providing a means of approximate characterization of phenomena which are too complex or too ill-defined to be amenable to description in conventional quantitative terms. More specifically, the fuzzy sets which represent the restrictions associated with the values of a linguistic variable may be viewed as summaries of various subclasses of elements in a universe of discourse. This, of course, is analogous to the role played by words and sentences in a natural language. For example, the adjective handsome is a summary of a complex of characteristics of the appearance of an individual. It may also be viewed as a label for a fuzzy set which represents a restriction imposed by a fuzzy variable named handsome. From this point of view, then, the terms very handsome, not handsome, extremely handsome, quite handsome, etc., are names of fuzzy sets which result from operating on the fuzzy set handsome with the modifiers named very, not, extremely, quite, etc. In effect, these fuzzy sets, together with the fuzzy set labelled handsome, play the role of values of the linguistic variable Appearance.

Fuzzy (Rule-Based) Systems

In the field of artificial intelligence (machine intelligence) there are various ways to represent knowledge. Perhaps the most common way to represent human knowledge is to form it into natural language expressions of the type

IF premise (antecedent), THEN conclusion (consequent) (3.2.11)

The form in Expression (3.2.11) is commonly referred to as the IF–THEN rule-based form; this form generally is referred to as the deductive form. It typically expresses an inference such that if we know a fact (premise, hypothesis, antecedent), then we can infer, or derive, another fact called a conclusion (consequent). This form of knowledge representation, characterized as shallow knowledge, is quite appropriate in the context of linguistics because it expresses human empirical and heuristic knowledge in our own language of communication. It does not, however, capture the deeper forms of knowledge usually associated with intuition, structure, function, and behavior of the objects around us simply because these latter forms of knowledge are not readily reduced to linguistic phrases or representations; this deeper form, as described in Chapter 1, is referred to as inductive.

Table 3.2: The canonical form for a fuzzy rule-based system

Rule 1 : IF condition C^1, THEN restriction R^1
Rule 2 : IF condition C^2, THEN restriction R^2
Rule r : IF condition C^r, THEN restriction R^r

The fuzzy rule-based system is most useful in modelling some complex systems that can be observed by humans because it makes use of linguistic variables as its antecedents and consequents; as described here these linguistic variables can be naturally represented by fuzzy sets and logical connectives of these sets.

By using the basic properties and operations defined for fuzzy sets, any compound rule structure may be decomposed and reduced to a number of simple canonical rules as given in Table 3.2. These rules are based on natural language representations and models, which are themselves based on fuzzy sets and fuzzy logic. The fuzzy level of understanding and describing a complex system is expressed in the form of a set of restrictions on the output based on certain conditions of the input (see Table 3.2). Restrictions are generally modelled by fuzzy sets and relations. These restriction statements are usually connected by linguistic connectives such as ''and,'' ''or,'' or ''else.'' The restrictions $R^1, R^2, \ldots, R^r$ apply to the output actions, or consequents of the rules. The following illustrates a couple of the most common techniques [Ross (1995)] for decomposition of linguistic rules involving multiple antecedents into the simple canonical form illustrated in Table 3.2.

Multiple conjunctive antecedents

$$IF\ \ x \text{ is } A^1 \text{ and } A^2 \ldots \text{and } A^L\ \ THEN\ \ y \text{ is } B^s$$

Assuming a new fuzzy subset A^s as

$$A^s = A^1 \cap A^2 \cap \ldots \cap A^L,$$

expressed by means of membership function

$$\mu_{A^s}(x) = \min\left[\mu_{A^1}(x), \mu_{A^2}(x), \ldots, \mu_{A^L}(x)\right]$$

based on the definition of the standard fuzzy intersection operation, the compound rule may be rewritten as

$$IF\ \ A^s\ \ THEN\ \ B^s$$

Multiple conjunctive antecedents

$$IF\ \ x \text{ is } A^1\ or\ A^2 ... or\ \ A^L\ \ THEN\ \ y \text{ is } B^s$$

Could be rewritten as

$$IF\ \ A^s\ \ THEN\ \ B^s,$$

where fuzzy set A^s is defined as

$$A^s = A^1 \cup A^2 \cup ... \cup A^L,$$

$$\mu_{A^s}(x) = \max\left[\mu_{A^1}(x), \mu_{A^2}(x), ..., \mu_{A^L}(x)\right].$$

This is based on the definition of the standard fuzzy union operation.

3.2.1 Fuzzy Expert Systems

An expert system is a program which contains human expert's knowledge and gives answers to the user's query by using an inference method. The knowledge is often stored in the form of rule base, and the most popular form is that of "IF-THEN".

A fuzzy expert system is an expert system which can deal uncertain and fuzzy information. In our real world, a human expert has his knowledge in the form of linguistic terms. Therefore it is natural to represent the knowledge by fuzzy rules and thus to use fuzzy inference methods.

Table 3.3: Lookup table

e \ ce	−1.0	−0.8	−0.6	−0.4	−0.2	0	0.2	0.4	0.6	0.8	1.0
−1.0	−1.0	−1.0	−1.0	−1.0	−1.0	−0.8	−0.8	−0.8	−0.8	−0.6	−0.6
−0.8	−0.8	−0.8	−0.8	−0.8	−0.8	−0.8	−0.6	−0.6	−0.6	−0.6	−0.6
−0.6	−0.6	−0.6	−0.6	−0.6	−0.4	−0.4	−0.4	−0.4	−0.4	−0.2	−0.2
−0.4	−0.2	0	0	0	0	0	0	0.2	0.2	0.2	0.2
−0.2	−0.4	−0.4	−0.4	−0.4	−0.4	−0.4	0	0	0	0	0
0	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	−0.2	0	0	0	0
0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.4	0.4
0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.6	0.6	0.6	0.6	0.6
0.6	0.6	0.6	0.4	0.4	0.4	0.4	0.8	0.8	0.8	0.8	0.8
0.8	0.6	0.6	0.6	0.6	0.6	0.6	0.8	0.8	0.8	0.8	0.8
1.0	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.8	0.8	0.8

The structure of a fuzzy expert system is similar to that of the fuzzy logic controller. Its configuration is shown in Fig. 3.13. As in the fuzzy logic controller, there can be fuzzification interface, knowledge base, and inference engine (decision making logic). Instead of the defuzzification module, there is the linguistic approximation module.

Fuzzification Interface

This module deals user's request, and thus we have to determine the fuzzification strategy. If we want to make the fuzzy expert system receive linguistic terms, this module has to have an ability to handle such fuzzy information. The fuzzification strategy, if necessary, is similar to that of the fuzzy logic controller.

In contradiction to the fuzzy logic controller, it is not needed to consider the discretization or normalization. But the fuzzy partition and assigning fuzzy linguistic terms to each sub region are necessary.

The expert's knowledge may be represented in the form of "IF-THEN" by using fuzzy linguistic terms. Each rule can have its certainty factor which represents the certainty level of the rule. This certainty factor is used in the aggregation of the results from each rule.

Inference Engine (Decision Making Logic)

The fuzzy expert systems can use the inference methods of the fuzzy logic controller. The system does not deal with a machine or process, and thus it is difficult to have a fuzzy set with monotonic membership function in the consequent part of a rule. Therefore especially, Mamdani method and Larsen method are often used.

Linguistic Approximation

As we stated before, a fuzzy expert system does not control a machine nor a process, and thus, in general, the defuzzification is not necessary. Instead of the defuzzification module, sometimes we need a linguistic approximation module.

This module finds a linguistic term which is closest to the obtained fuzzy set. To do it, we may use a measuring technique of distance between fuzzy sets.

Scheduler

This module controls all the processes in the fuzzy expert system. It determines the rules to be executed and sequence of their executions. It may also provide an explanation function for the result. For example, it can show the reason how the result was obtained.

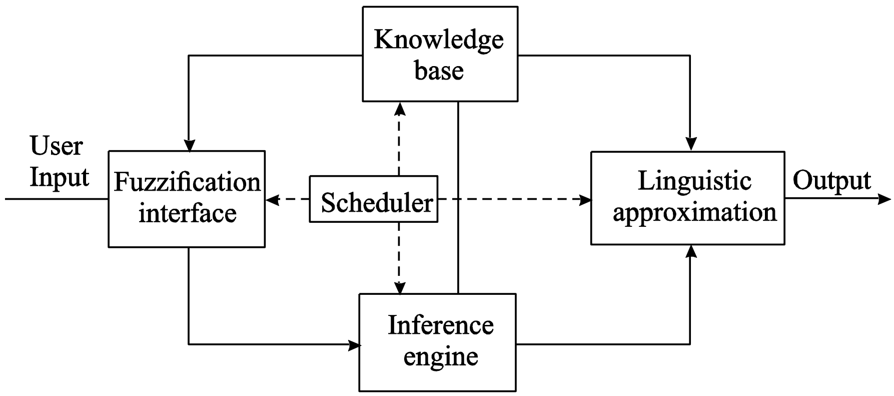


Figure 3.13: Configuration of fuzzy expert system

3.3. FUZZY MODELLING

Modelling, in a general sense, refers to the establishment of a description of a system (a plant, a process, etc.) in mathematical terms, which characterizes the input-output behaviour of the underlying system.

To describe a physical system, such as a circuit or a microprocessor, we have to use a mathematical formula or equation that can represent the system both qualitatively and quantitatively. Such a formulation is a mathematical representation, called a mathematical model, of the physical system.

Most physical systems, particularly those complex ones, are extremely difficult to model by an accurate and precise mathematical formula or equation due to the complexity of the system structure, nonlinearity, uncertainty, randomness, etc. Therefore, approximate modelling is often necessary and practical in real world applications.

Intuitively, approximate modelling is always possible. However, the key questions are what kind of approximation is good, where the sense of "goodness" has to be first defined, of course, and how to formulate such a good approximation in modelling a system such that it is mathematically rigorous and can produce satisfactory results in both theory and applications.

From the detailed studies in the last two chapters, it is clear that interval mathematics and fuzzy logic together can provide a promising alternative to mathematical modelling for many physical systems that are too vague or too complicated to be described by simple and crisp mathematical formulas or equations. When interval mathematics and fuzzy logic are employed, the interval of confidence and the fuzzy membership functions are used as approximation measures, leading to the so-called fuzzy systems modelling.

Following the traditional classification in the field of control systems, a system that describes the input-output behaviour in a way similar to a mathematical mapping without involving a differential operator or equation is called a static system. In contrast, a system described by a differential operator or equation is called a dynamic system. In this chapter, static fuzzy systems modeling is first discussed, including its stability analysis, and the dynamic fuzzy systems modeling will then follow, along with its stability and controllability analyses.

Basic Concepts of Systems Modelling

The fundamental concept of systems modelling can be illustrated as follows:

Suppose that we have an unknown system ("black box"), for which only a set of its inputs $x_1, x_2, ..., x_n$ and outputs $y_1, y_2, ..., y_m$ can be measured (or observed) and so these data are available. Here, both inputs and outputs can be either discrete-time series or continuous signals. We want to find a mathematical description to qualitatively and quantitatively characterize this unknown system, in the sense that by inputting $x_1, x_2, ..., x_n$ into the mathematical description we can always obtain the corresponding outputs $y_1, y_2, ..., y_m$. Establishing such a mathematical description is called mathematical modelling for the unknown system (Fig. 3.14). As usual, the mathematical description can be a mathematical formula, such as a mapping or a functional that relates the inputs to the outputs in the form

$$\begin{cases} y_1 = f_1(x_1, x_2, ..., x_n), \\ . \\ . \\ y_m = f_1(x_1, x_2, ..., x_n), \end{cases} \tag{3.3.1}$$

or a set of differential equations (assuming proper conditions) in the form

$$\begin{cases} y_1 = f_1(x_1, ..., x_n, \dot{x}_1, ..., \dot{x}_n), \\ . \\ . \\ y_m = f_1(x_1, ..., x_n, \dot{x}_1, ..., \dot{x}_n), \end{cases} \tag{3.3.2}$$

or a logical linguistic statement, which can be quantified mathematically, in the form

IF (input x_1) AND ... AND (input x_n) THEN (output y_1) AND ... AND (output y_m). (3.3.3)

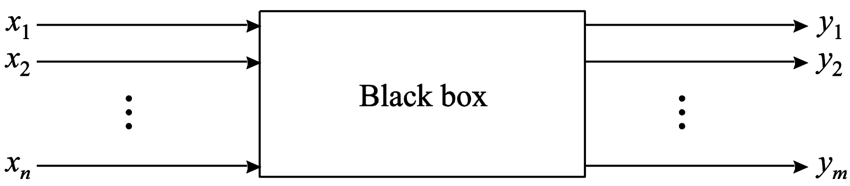


Figure 3.14: An unknown system as a "black box"

Fuzzy system modelling is to quantify the logical linguistic form (3.4.3) by using fuzzy logic and the mathematical functional model (3.4.1), or by using fuzzy logic together with the differential equation model (3.4.2). The result of the former is called a static fuzzy system and the latter, a dynamic fuzzy system.

3.4 MODELLING OF STATIC FUZZY SYSTEMS

Return to the "black box" shown in Fig. 3.14. Since all the inputs $x_1, x_2, ..., x_n$ and outputs $y_1, y_2, ..., y_m$ are assumed to be available the logical linguistic statement (3.3.3) has actually described the unknown system on the basis of the available data. Yet this is not the ultimate purpose of mathematical modelling, since if a new input x_{n+1} comes in, we don't know what the corresponding output should be. The main purpose of mathematical modelling, therefore, is not only to correctly describe the existing input-output relations through the unknown system but also to enable the established model to approximately describe other possible hidden input-output relations of the system. Thus, a general approach is to first quantify the linguistic statement (3.3.3) and then to relay the quantified logical input-output relations by using the mathematical functional (3.3.1), or differential equations (3.3.2).

Recall that a finite fuzzy logic implication statement can always be described by a set of general fuzzy IF-THEN rules containing only the fuzzy logic AND operation, in the following multi-input single-output form:

(1) IF (x_1 is X_{11}) AND ... AND (x_n is X_{1n}) THEN (y_1 is Y_1).

(2) IF (x_1 is X_{21}) AND ... AND (x_n is X_{2n}) THEN (y_1 is Y_2).

(N) IF (x_1 is X_{N1}) AND ... AND (x_n is X_{Nn}) THEN (y_1 is Y_N).

Here, we should recall that the phrase "x is X" is an abbreviation of the complete statement "x belongs to the fuzzy subset X with a corresponding membership value $\mu_X(x)$." We should now restrict our discussion on closed intervals for the fuzzy subsets $X_{11}, X_{12}, ..., X_{Nn}$ and $Y_1, Y_2, ..., Y_N$, so that interval arithmetic can be applied.

For simplicity of discussion, we first consider the simplest case where N = 1 with only one fuzzy IF-THEN rule:

$$R^1: \text{IF } (x_1 \text{ is } X_1) \text{ AND ... AND } (x_n \text{ is } X_n) \text{ THEN } y \text{ is } Y.$$

An example is the following rule with constants $\{a_0, a_1, ..., a_n\}$:

$$R^1: \text{IF } (x_1 \text{ is } X_1) \text{ AND ... AND } (x_n \text{ is } X_n) \text{ THEN } y = a_0 + a_1 x_1 + ... + a_n x_n.$$

When a set of particular inputs are available:

$$x_1 = x_1^0 \in X_1, ..., x_n = x_n^0 \in X_n \text{ with } \mu_{X_1}(x_1^0), ..., \mu_{X_n}(x_n^0),$$

the output y will assume the value

$$y^0 = a_0 + a_1 x_1^0 + ... + a_n x_n^0, \quad (3.4.1)$$

with membership value given by the following general rule

$$\mu_Y(y^0) = \bigvee_{y^0 = a_0 + a_1 x_1^0 + \ldots + a_n x_n^0} \left\{ \mu_{X_1}(x_1^0) \wedge \ldots \wedge \mu_{X_n}(x_n^0) \right\}. \tag{3.4.2}$$

If there are more than one fuzzy IF-THEN rule:

R^i : IF (x_1 is X_{i1}) AND ... AND (x_n is X_{in})

THEN $y_i = a_{i0} + a_{i1}x_1 + \ldots + a_{in}x_n$, $i = 1, 2, \ldots, N$.

In this case, with the given set of inputs $x_1 = x_1^0 \in X_1, \ldots, x_n = x_n^0 \in X_n$ namely, the same inputs applied to all the different rules, one will have

$$\begin{aligned} y_1^0 &= a_{10} + a_{11}x_1^0 + \ldots + a_{1n}x_n^0, \\ y_2^0 &= a_{20} + a_{21}x_1^0 + \ldots + a_{2n}x_n^0, \\ &. \\ y_N^0 &= a_{N0} + a_{N1}x_1^0 + \ldots + a_{Nn}x_n^0. \end{aligned} \tag{3.4.3}$$

The corresponding membership values for these outputs are given as

$$\mu_Y(y_i^0) = \bigvee_{y^0 = a_{i0} + a_{i1} x_1^0 + \ldots + a_{in} x_n^0} \left\{ \mu_{X_1}(x_1^0) \wedge \ldots \wedge \mu_{X_n}(x_n^0) \right\}, i = 1, \ldots, N. \tag{3.4.4}$$

In a typical modelling approach, the final single output, y, is usually obtained via the following weighted average formula:

$$y = \frac{\sum_{i=1}^{N} \mu_Y(y_i^0) \cdot y_i^0}{\sum_{i=1}^{N} \mu_Y(y_i^0)}, \tag{3.4.5}$$

where "." is the ordinary algebraic multiplication.

For the most general situation, we assume that all the coefficients $\left\{ a_{i0}, a_{i1}, \ldots, a_{in} : i = 1, 2, \ldots, N \right\}$ are uncertain and belong to certain intervals:

$$a_{i0} \in A_0, \ldots, a_{in} \in A_n, i = 1, 2, \ldots, N,$$

where, for example,

$$A_0 = \left[\min\left\{ a_{10}, a_{20}, \ldots, a_{N0} \right\}, \max\left\{ a_{10}, a_{20}, \ldots, a_{N0} \right\} \right],$$

$$A_1 = \left[\min\left\{ a_{11}, a_{21}, \ldots, a_{N1} \right\}, \max\left\{ a_{11}, a_{21}, \ldots, a_{N1} \right\} \right],$$

$$A_n = \left[\min\left\{ a_{1n}, a_{2n}, \ldots, a_{Nn} \right\}, \max\left\{ a_{1n}, a_{2n}, \ldots, a_{Nn} \right\} \right],$$

Thus, with the given inputs $x_1 \in X_1, x_2 \in X_2, ..., x_n \in X_n,$ the output becomes

$$Y = A_0 + A_1 \cdot X_1 + ... + A_n \cdot X_n, \tag{3.4.6}$$

which yields the fuzzy subset (interval) for y with the membership functions given by the general rule as

$$\mu_{Y,i}(y_i) = \bigvee_{y_i = a_{i0} + ... + a_{in} x_n} \left\{ \mu_{X_1}(x_1^0) \wedge ... \wedge \mu_{X_n}(x_n^0) \right\}, \; i = 1, ..., N. \tag{3.4.7}$$

Finally, the output y is computed by the weighted average formula

$$y = \frac{\sum_{i=1}^{N} \mu_{Y,i}(y_i) \cdot y_i}{\sum_{i=1}^{N} \mu_{Y,i}(y_i)} = \sum_{i=1}^{N} \beta_i \cdot y_i, \text{ where } \beta_i = \frac{\mu_{Y,i}(y_i)}{\sum_{i=1}^{N} \mu_{Y,i}(y_i)}. \tag{3.4.8}$$

This is a convex combination of the outputs $y_i, \; i = 1, 2, ..., N.$

The three formulas (3.4.6) to (3.4.8) are sometimes called the input-output algorithm for static fuzzy system modeling under the fuzzy IFTHEN rules $R^i, \; i = 1, 2, ..., N,$ where the former has constant coefficients, while the latter has interval coefficients.

Example 3.4.1: Consider an unknown system with two inputs, $x_1, \; x_2$ and one output, y, as shown in Figure 3.5.1(a). Let inputs x_1 be within the range $X_1 = [0, 20]$ = [0,20] and x_2 in $X_2 = [0,10].$ Suppose that X_1 has two associate membership functions, $\mu_{s_1}(\cdot)$ and $\mu_{L_1}(\cdot)$ describing "small" and "large," respectively, and

Similarly X_2 has $\mu_{s_2}(\cdot)$ and $\mu_{L_2}(\cdot)$ as shown in Fig. 3.15 (b)-(c).

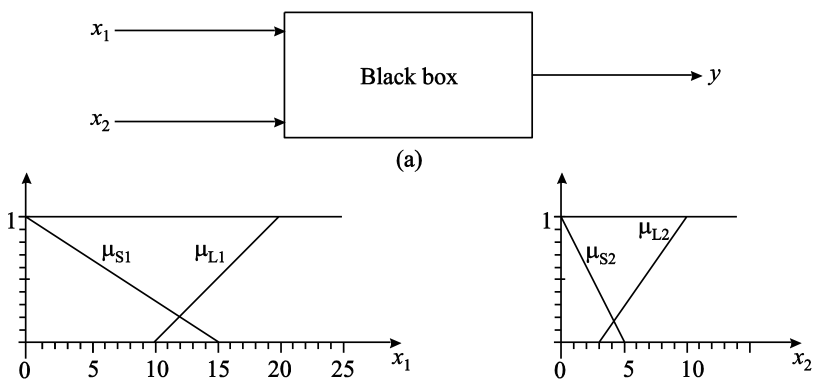


Figure 3.15: The unknown system and its two inputs

Now, suppose that from experiments, the following input-output relations (fuzzy IF-THEN implication rules):

R^1 : IF (x_1 is small) AND (x_2 is small) THEN $y = x_1 + x_2$

R^2 : IF x_1 is large THEN $y = \frac{1}{2} x_1$

R^3 : IF x_2 is large THEN $y = \frac{1}{3} x_2$

Otherwise, $y = 0$.

Here, we assume that x_1 and x_2 will not both be large in this example, just for simplicity of discussion with consistence.

Let $x_1^0 = 13$ and $x_2^0 = 4$ be given, for instance. It can be found from Figures 3.5.1 (b)-(c) that

$$\mu_{S_1}(x_1^0) = 2/15,\ \mu_{L_1}(x_1^0) = 3/10, \mu_{S_2}(x_2^0) = 1/5,\ \mu_{L_2}(x_2^0) = 1/7.$$

Then, applying the input-output algorithm, we compute the following:

(1) The corresponding outputs:

$$y_1^0 = x_1^0 + x_2^0 = 13 + 4 = 17,$$

$$y_2^0 = \frac{1}{2} x_1^0 = \frac{13}{2}, \text{ and } y_3^0 = \frac{1}{3} x_2^0 = \frac{4}{3}.$$

(2) The fuzzy set Y:

$$a_{11} = 1, a_{12} = 1, a_{21} = 1/2, a_{22} = 0, a_{31} = 0,\ a_{32} = 1/3$$

$$A_1 = \left[\min\left\{1, \frac{1}{2}, 0\right\}, \max\left\{1, \frac{1}{2}, 0\right\} \right] = [0, 1],$$

$$A_2 = \left[\min\left\{1, 0, \frac{1}{3}\right\}, \max\left\{1, 0, \frac{1}{3}\right\} \right] = [0, 1],$$

$$Y = A_1 \cdot X_1 + A_1 \cdot X_1 = [0,1]\cdot[0,20] + [0,1]\cdot[0,10] = [0,30].$$

(3) The fuzzy membership values of the outputs:

$$\mu_{Y,1}(y_1^0) = \bigvee_{y_1^0 = x_1^0 + x_2^0} \left\{ \mu_{S_1}(x_1^0) \wedge \mu_{S_2}(x_2^0) \right\}$$

$$\max\left\{ \min\left\{ \mu_{S_1}(x_1^0) \wedge \mu_{S_2}(x_2^0) \right\} \right\} = \max\left\{ \min\left\{ \frac{2}{15}, \frac{1}{5} \right\} \right\} = \frac{2}{15}.$$

$$\mu_{Y,2}(y_2^0) = \bigvee_{y_1^0=\frac{1}{2}x_1^0} \left\{\min\left\{\mu_{L_1}(x_1^0)\right\}\right\} = \max\left\{\frac{3}{10}\right\} = \frac{3}{10}.$$

$$\mu_{Y,3}(y_3^0) = \bigvee_{y_1^0=\frac{1}{3}x_2^0} \left\{\min\left\{\mu_{L_2}(x_2^0)\right\}\right\} = \max\left\{\frac{1}{7}\right\} = \frac{1}{7}.$$

(4) The average value of the final output, corresponding to the input $x_1^0 = 13$ and $x_2^0 = 4$:

$$y^0 = \frac{\sum_{i=1}^{N} \mu_{Y,i}(y_i^0)\cdot y_i^0}{\sum_{i=1}^{N} \mu_{Y,i}(y_i^0)} = \frac{\frac{2}{15}\cdot 17 + \frac{3}{10}\cdot\frac{13}{2} + \frac{1}{7}\cdot\frac{4}{3}}{\frac{2}{15}+\frac{3}{10}+\frac{1}{7}} = 7.649.$$

(5) The three membership functions, one "small" and two "large," for the final output:

$$\mu_S(y) = \begin{cases} -\dfrac{y}{20}+1, & 0 \le y \le 20, \\ 0, & 20 \le y \le 30, \end{cases} \qquad \mu_{L_1}(y) = \begin{cases} 0, & 0 \le y \le 5, \\ \dfrac{y}{5}-1, & 5 \le y \le 10, \\ 1, & 10 \le y \le 30, \end{cases}$$

$$\mu_{L_2}(y) = \begin{cases} 0, & 0 \le y \le 1, \\ \dfrac{3y}{7}-\dfrac{3}{7}, & 1 \le y \le \dfrac{10}{3}, \\ 1, & \dfrac{10}{3} \le y \le 30, \end{cases}$$

However, $\mu_S(y)$ can also be computed from $\mu_{S_1}(y)$ and $\mu_{S_2}(y)$ by the α-cut operations and similarly, $\mu_{L_1}(y)$ from $\mu_{L_1}(x_1^0)$ and $\mu_{L_2}(y)$ from $\mu_{L_2}(x_2^0)$, respectively. Here, to find $\mu_{L_1}(y)$ from $\mu_{L_1}(x_1^0)$, for instance, we use α-cut on the membership function curve $\mu_{L_1}(y): \alpha = (x-10)/10$, and obtain $x_{11} = 10\alpha + 10$, while $x_{12} = 20$ is obtained from Fig. 3.16. Thus, we have $[x_{11}, x_{12}] = [10\alpha + 10, 20]$. It follows from the implication rule $R^2\left(y_2 = \frac{1}{2}x_1\right)$ that the output interval is $[y_{21}, y_{22}] = [5\alpha + 5, 10]$. Then, by setting $y_{21} = 5\alpha + 5$, we have $\alpha = y_{21}/5 - 1$ on [5, 10] (since $\mu_{L_1}(y=5) = 0$ and $\mu_{L_1}(y=10) = 1$). These membership functions are finally extended to the interval Y = [0, 30]. The resulting three output membership functions are shown in Fig. 3.16.

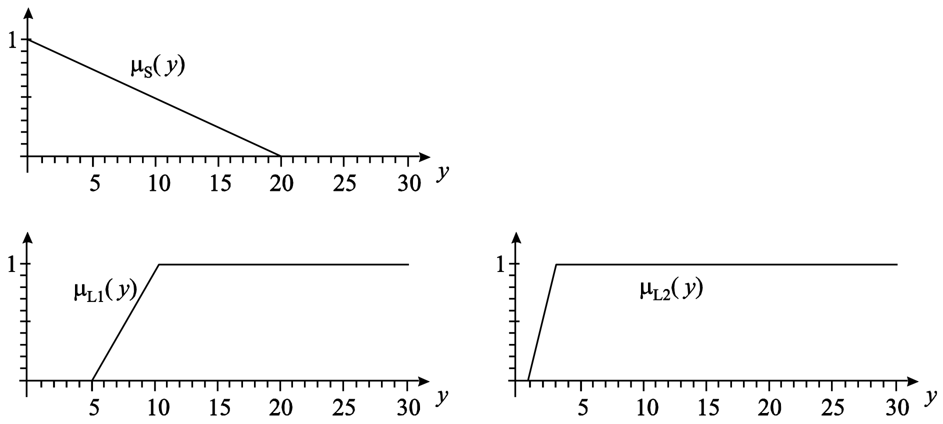


Figure 3.16: The three output membership functions

3.5 MODELLING OF DYNAMICS FUZZY SYSTEMS

A static fuzzy model given by a set of fuzzy IF-THEN rules in the form

$$R^i: \text{IF } (x_1 \text{ is } X_{i1}) \text{ AND ... AND } (x_n \text{ is } X_{in})$$

$$\text{THEN } y_i = a_{i0} + a_{i1}x_1 + \ldots + a_{in}x_n,\ i = 1, 2, \ldots, N, \tag{3.5.1}$$

and the final output is

$$y = \frac{\sum_{i=1}^{N} \mu_{Y,i}(y_i) \cdot y_i}{\sum_{i=1}^{N} \mu_{Y,i}(y_i)},\ i = 1, 2, \ldots, N. \tag{3.5.2}$$

This fuzzy modeling by its nature is static. If the input-output relations are such that

$$x_1 = x(k),$$

$$x_2 = x(k-1),$$

$$x_n = x(k-(n-1)),$$

and

$$y_i = y_i(k),\ i = 1, 2, \ldots, N,$$

so that

$$y_i(k) = a_{i0} + a_{i1}x(k) + a_{i2}x(k-1) + ... + a_{in}x(k-(n-1)),\ i = 1,...,N, \quad (3.5.3)$$

then this new system is considered to be dynamic, since it is described by a set of difference equations rather than a simple input-output mapping. This case is special in that all the inputs $x_1, x_2, ..., x_n$ are related (each state is the delay of its previous one) and the output is given at the last step of the process. (3.5.3) is a typical discrete-time dynamic system in the classical systems theory if all quantities involved are crisp.

Dynamic Fuzzy Systems without Control

It is interesting to study the stability of a discrete-time dynamic (fuzzy) system described by (3.6.3) because that when we are doing system modelling, we would like that the resulting model works well in the sense that it approximates the unknown system closely (preferably, optimal in some sense). It performs its actions stably, particularly for dynamic processes. In other words, stability is one criterion or one of the few most important issues in dynamic system modelling as is in common practice of system engineering.

Definition 3.5.1: A typical single-input/single-output (SISO), discrete-time dynamic fuzzy system is a fuzzy model described by a set of fuzzy IF-THEN rules of the form as follows:

R^i : IF ($x(k)$ is X_{i1}) AND ... AND ($x(k-(n-1))$ is X_{in})

THEN $y_i(k) = a_{i0} + a_{i1}x(k) + ... + a_{in}x(k-(n-1)),\ i = 1, 2, ..., N,$

With $x(k+1) = c_k y(k+1), k = 0, 1, 2, ...,$ where

$$y(k+1) = \frac{\sum_{i=1}^{N} w_i \cdot y_i(k+1)}{\sum_{i=1}^{N} w_i},\ \ k = 0, 1, 2, ...$$

Here, the fuzzy sets consist of intervals $\{X_j : j = 1, ..., n\}$ with the associate fuzzy membership functions $\{\mu_{X_j} : j = 1, ..., n\}$ and $\{w_i : i = 1, ..., N\}$ is a set of weights satisfying $w_i \geq 0,\ i = 1, ..., N,\ \sum_{i=1}^{N} w_i > 0.$

In all the rules $R^i, i = 1, 2, ..., N,$ in this definition X_{ij} share the same fuzzy subset X_j and the same membership function μ_{X_j} for each $j = 1, 2, ..., n.$

It may be noted that, similar to formula (3.6.1), the weights $\{w_i : i = 1, ..., N\}$ usually are chosen to be equal to $\mu_{Y_j}(y_i)$. Moreover, for simplicity of notation and discussion in the following, we will always let $c_k = 1$ for all $k = 0,1,2,...$ and $a_{i0} = 0,\ \forall\ i = 1,2,..., N,$ for all 1,...,N, although this is not necessary in general.

Next, we define some standard concepts and stability results in classical systems theory.

Definition 3.5.2: A multi-input/multi-output (MIMO), nonlinear, discrete-time, dynamic system of the form

$$x(k+1) = f(x(k)),\ \ x(k) \in R^m,\ k = 0,1,2,...,$$

is said to be asymptotically stable about an equilibrium point x_e, or x_e is an asymptotically stable equilibrium point of the system, if

$$x_e = f(x_e)$$

and, starting from any $x(0) \in R^m$, all $x(k)$ are bounded and

$$x(k) \to x_e\ \ (k \to \infty)$$

In this definition, the convergence $x(k) \to x_e\ \ (k \to \infty)$ is usually measured by the l_2-norm (the "length" of a vector), namely,

$\lim\limits_{k\to\infty} \|x(k) - x_e\|_2 = 0,$ where $\left\| [x_1 ... x_m]^T \right\|_2 = \sqrt{x_1^2 + ... + x_m^2}$ is the length vector.

The following stability criterion is well known in classical systems theory.

Theorem 3.5.1: Suppose that the multi-input/multi-output (MIMO), nonlinear, discrete-time, dynamic system of the form

$$x(k+1) = f(x(k)),\ \ x(k) \in R^m,\ k = 0,1,2,...,$$

has an equilibrium point x_e, or x_e is and that there exist a scalar valued function $V(x(k))$ satisfying

(i) $V(0) = 0;$

(ii) $V(x(k)) > 0\ \ for\ \ all\ \ x(k) \neq 0;$

(iii) $V(x(k)) \to \infty\ as\ \|x(k)\|_2 \to \infty$; and

(iv) $V(x(k+1)) - V(x(k)) < 0\ \ for\ \ all\ \ x(k) \neq 0,\ and\ \ all\ \ k = 0,1,2,...$

Then this system is asymptotically stable about the equilibrium point 0.

In theorem 3.5.1, the function V(x(k)), if it exists, is called a Lyapunov function.

Note that this theorem particularly applies to all linear time-invariant state-space systems of the form

$$x(k+1) = A\,x(k),\ \ x(k) \in R^m,\ k = 0,1,2,... \tag{3.5.4}$$

For this linear time-invariant system, however, there is a very simple criterion for the determination of its asymptotic stability.

Theorem 3.5.2: Let $\lambda_j,\ j = 1,2,...,m,$ be Eigen values, counting multiple ones, of the constant matrix A in the linear time-invariant system (3.5.4). Then the system is asymptotically stable about the equilibrium point 0 if and only if

$$\left|\lambda_j\right| < 1, \text{ for all } j = 1,2,...,m.$$

Now, return to the single-input/single-output, linear, discrete-time dynamic fuzzy system described in Definition 3.5.1. To apply the classical stability theorem 3.5.1 or 3.5.2 to it, we first reformulate it in the state-space setting as discussed below:

Let $c_k = 1, k = 0,1,2,...,$ and $a_{i0} = 0,\ i = 1,...,N,$ in the system (for simplicity, as mentioned above). Define

$$\boldsymbol{x}(k) = \left[x(k)\ x(k-1)\ \ x(k-2)...x(k-(n-1))\right]^T,$$

$$A_i = \begin{bmatrix} a_{i1} & a_{i2} & ... & a_{in-1} & a_{in} \\ 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & ... & 1 & 0 \end{bmatrix}.$$

Although this (canonical) formulation has some redundancy (only the first equation is essential), yet it is convenient to use for stability analysis. The system can be rewritten as

$$x(k+1) = \frac{\sum_{i=1}^{N} w_i A_i\, \boldsymbol{x}(k)}{\sum_{i=1}^{N} w_i},\ \ k = 0,1,2,... \tag{3.5.5}$$

Corollary 3.5.1: In the dynamic fuzzy system (3.5.5), let

$$x(k+1) = \sum_{i=1}^{n} \frac{w_i}{\sum_{i=1}^{n} w_i} A_i,$$

and assume that $\{w_i\}$ are constants independent of k. Then, the system is asymptotically stable if and only if all Eigen values of A, λ_i, for all $i = 1, 2, ..., n$ satisfy

$$|\lambda_i| < 1, \text{ for all } j = 1, 2, ..., n.$$

Next, we state a theorem whether proof related to fuzzy system (3.5.5).

Theorem 3.5.3: The discrete-time dynamic fuzzy system (3.5.5) is asymptotically stable about the equilibrium point 0 if there exists a common positive definite matrix P such that

$$A_i^T P A_i - P < 0, \quad \textit{for all } \; i = 1, 2, ..., N.$$

Dynamic Fuzzy Systems with Control

Let us consider, again, the typical single-input/single-output (SISO) linear discrete-time dynamic fuzzy system discussed and defined in the last subsection and taking consideration.

Definition 3.5.3: An SISO, discrete-time, dynamic fuzzy control system is a fuzzy control model described by a set of fuzzy IF-THEN rules of the form

R^i : (IF ($x(k)$ is X_{i1}) AND ... AND ($x(k-(n-1))$ is X_{in})) AND (IF ($u(k)$ is U_{i1}) AND ... AND ($(k-(m-1))$ is U_{im}))

$$\text{THEN} \quad \begin{aligned} & y_i(k) = a_{i0} + a_{i1}x(k) + ... + a_{in}x(k-(n-1)) \\ & + b_{i0} + b_{i1}u(k) + ... + b_{im}x(k-(m-1)), \; i = 1, 2, ..., N, \end{aligned}$$

with $m \le n$ and

$$x(k+1) = c_k y(k+1), \;\; k = 0, 1, 2, ..., \text{ where}$$

$$y(k+1) = \frac{\sum_{i=1}^{N} w_i y_i(k+1)}{\sum_{i=1}^{N} w_i}, \;\; k = 0, 1, 2, ...$$

In this system the fuzzy sets consist of intervals $\{X_j : j = 1,\ldots,n\}$ and $\{U_j : j = 1,\ldots,m\}$, and their associate fuzzy membership functions $\{\mu_{X_j} : j = 1,\ldots,n\}$ and $\{\mu_{X_j} : j = 1,\ldots,m\}$, respectively, and $\{w_i : i = 1,\ldots,N\}$ is a set of weights satisfying $w_i \geq 0,\ i = 1,\ldots,N,\ \sum_{i=1}^{N} w_i > 0.$

Here, we note again that for each rule $R^i : i = 1,2,\ldots N,$ all the X_{ij} share the same fuzzy subset X_j and the same membership function μ_{X_j} for each $j = 1,2,\ldots,n.$ Also, the weights are usually quantified by the output membership functions in the same way as in (3.5.1). For simplicity of notation and discussion, we assume $c_k = 1$ for all $k = 0,1,2,\ldots,$ and $a_{i0} = b_{i0} = 0$ for all $i = 1,2,\ldots N,$ in this model.

It is clear that Definitions 3.5.1 and 3.5.3 have no essential difference if the control inputs $\{u(k)\}$ in Definition 3.5.3 are independent of the states $\{x(k)\}$. In engineering control systems, however, most of the time we would like to have negative state-feedback controllers of the form:

$$
\begin{aligned}
u(k) &= -K_{11}x(k) - K_{12}x(k-1) - \ldots - K_{1n}x(k-(n-1)),\\
u(k-1) &= -K_{22}x(k-1) - \ldots - K_{2n}x(k-(n-1)),\\
&\vdots\\
u(k-m) &= -K_{mm}x(k-m) - \ldots - K_{mn}x(k-(n-1)),
\end{aligned}
$$

where K_{ij} are constant control gains to be determined.

To facilitate our discussion, let us consider a simple example. Suppose that a fuzzy control system is given by

R_S^1: IF x(k) is X_1 AND u(k) is U_1

THEN $x_1(k + 1) = a_1x(k) + b_1u(k)$.

R_S^2: IF x(k) is X_2 AND u(k) is U_2

THEN $x_2(k + 1) = a_2x(k) + b_2u(k)$

Assume also that a state-feedback fuzzy controller is designed to be one described by

R_C^1: IF x(k) is X_1 AND x(k – 1) is X_1

THEN $u_1(k) = -K_{11}x(k) – K_{12}x(k – 1)$

R_C^2: IF x(k) is X_2 AND x(k – 1) is X_2

THEN $u_2(k) = -K_{21}x(k) – K_{22}x(k – 1)$

Then, we can combine all possibilities, where it is important to note that for " $x(k)$ is X_1 and $u(k)$ is U_1" there are two possibilities for the previous control input $u(k-1)$ either "is U_1" or "is U_2." The same is true for "$x(k)$ is X_2 and $u(k)$ is U_1." Thus, we have

R^{11} : (IF x(k) is X_1) AND (IF u(k) is U_1 AND u(k – 1) is U_1)

THEN $x_{11}(k + 1)$: $= a_1x(k) + b_1u_1(k)$

$$= (a_1 - b_1K_{11})\, x(k) - b_1K_{12}x(k-1)$$

R^{12} : (IF x(k) is X_1) AND (IF u(k) is U_1 AND u(k – 1) is U_2)

THEN $x_{12}(k + 1)$: $= a_1x(k) + b_1u_2(k)$

$$= (a_1 - b_1K_{21})\, x(k) - b_1K_{22}x(k-1)$$

R^{21} : (IF x(k) is X_2) AND (IF u(k) is U_2 AND u(k – 1) is U_2)

THEN $x_{21}(k + 1)$: $= a_2x(k) + b_2u_1(k)$

$$= (a_2 - b_2K_{11})\, x(k) - b_2K_{12}x(k-1)$$

R^{22} : (IF x(k) is X_2) AND (IF u(k) is U_2 AND u(k – 1) is U_2)

THEN $x_{22}(k + 1)$: $= a_2x(k) + b_2u_2(k)$

$$= (a_2 - b_2K_{21})\, x(k) - b_2K_{22}x(k-1)$$

On using all state feedback inputs, the fuzzy system, described by Definition 3.6.3 reduces to a new closed-loop dynamic fuzzy system, given by the rules R_{11}, R_{12}, R_{21}, and R_{22}, containing no more control variables $u(k)$. Thus, the stability conditions obtained in Theorem 3.5.3 can be applied, namely, the designer can choose the feedback control gains K_{11}, K_{12}, K_{21} and K_{22} to ensure the asymptotic stability of the overall controlled system.

Example 3.5.3: Consider a fuzzy control system described by the following rule base:

R_S^1: IF x(k) is X_1 AND x(k -1) is X_1

THEN $x_1(k + 1) = 2.178x(k) - 0.588x(k-1) + 0.603u(k)$

R_S^2: IF x(k) is X_2 AND x(k – 1) is X_2

THEN $x_2(k + 1) = 2.256x(k) - 0.361x(k-1) + 1.120u(k)$

Let the fuzzy state-feedback controller be described by

R_C^1: IF x(k) is X_1 AND x(k – 1) is X_1

THEN $u_1(k) = r(k) - K_{11}x(k) - K_{12}x(k-1)$

R_C^2: IF x(k) is X_2 AND x(k – 1) is X_2

THEN $u_2(k) = r(k) - K_{21}x(k) - K_{22}x(k-1)$

where, $\{r(x)\}$ is a reference signal (set-points).

Since the control input $u(k)$ does not appear in the condition parts, we consider its membership values to be identically equal to 1 therein. The resulting closed-loop dynamic fuzzy system is then obtained as follows:

R^{11} : IF x(k) is X_1 AND x(k – 1) is X_1

THEN x_{11} (k + 1) = (2.178 – 0.603K_{11}) x(k)

+ (-0.588 – 0.603 K_{12}) x(k – 1) + 0.603 r(k)

R^{12} : IF x(k) is X_1 AND x(k – 1) is X_2

THEN x_{12} (k + 1) = (2.178 – 0.603 K_{21}) x(k)

+ (-0.588 – 603 K_{22}) x(k – 1) + 0.603 r(k)

R^{21} : IF x(k) is X_2 AND x(k – 1) is X_1

THEN x_{21} (k + 1) = (2.256 – 1.120 K_{11}) x(k)

+ (-0.361 – 1.120 K_{12}) x(k – 1) + 1.120 r(k)

R^{22} : IF x(k) is X_2 AND x(k – 1) is X_2

THEN x_{22} (k + 1) = (2.256 – 1.120 K_{21}) x(k)

+ (-0.361 – 1.120 K_{22}) x(k – 1) + 1.120 r(k)

The overall state output $x(k+1)$ is computed by

$$x(k+1) = \frac{w_1 x_{11}(k+1) + w_2 x_{12}(k+1) + w_3 x_{21}(k+1) + w_4 x_{22}(k+1)}{w_1 + w_2 + w_3 + w_4}$$

for some weights satisfying $w_i \geq 0,\ i = 1, \ldots, N,\ \sum_{i=1}^{N} w_i > 0.$

Then, the weights can be determined by using these membership functions on the same line. For simplicity we assume that the reference signal r(k) = 0 for all k = 0,1,2,..., and that the controller in both R_C^1 and R_C^2 are as given below:

$$u_1(k) = u_2(k) = -K\, x(k),\ \ k = 0, 1, 2, \ldots$$

Then, the closed-loop fuzzy system reduces to the following simple one:

R^1 : IF x(k) is X_1 AND x(k – 1) is X_1

THEN x_1(k + 1) = (2.178 – 0.603K) x(k) – 0.588 x(k – 1)

R^2 : IF x(k) is X_2 AND x(k – 1) is X_2

THEN x_2(k + 1) = (2.256 – 1.120K) x(k) – 0.361 x(k – 1)

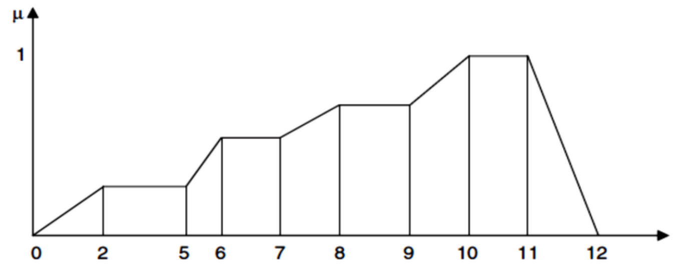
Finally, by applying the above stability theorems, we can verify that this system is stable.

EXERCISE- 3

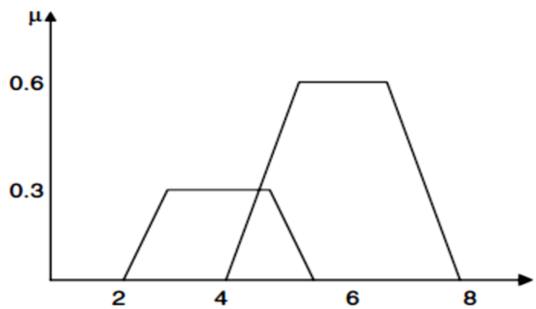
1. Determine crisp $\alpha-$cut relation for $\alpha=0.2;$ for $j=1,2,\ldots,10$ for the following fuzzy relation matrix R:

$$R=\begin{bmatrix}0.3 & 0.8 & 0.7 & 0.9\\ 1 & 0.7 & 0.6 & 0.2\\ 0.1 & 0.7 & 1 & 0.9\\ 0.5 & 0.6 & 0.2 & 0.5\end{bmatrix}.$$

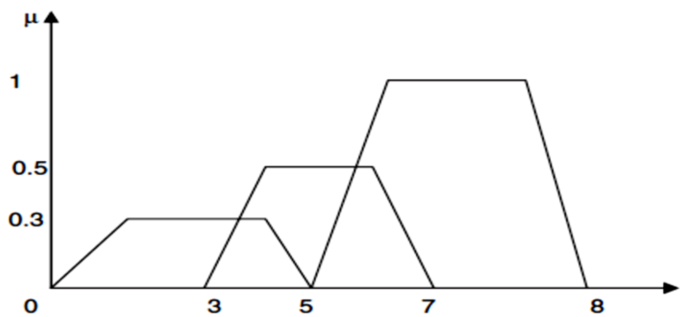
2. By using centroid method of defuzzification convert fuzzy value z to precise value Z^{*} for the following graph.



3. Find the defuzzified value by weighted average method shown in figure.



4. Find the defuzzified values using (a) center of sums methods and (b) center of largest area for the figure shown.



5. In reference to car speeds we have the linguistic variables ‘‘fast’’ and ‘‘slow’’ for speed:

$$\text{Fast} = \left\{ \frac{0}{0} + \frac{0.1}{10} + \frac{0.2}{20} + \frac{0.3}{30} + \frac{0.4}{40} + \frac{0.5}{50} + \frac{0.6}{60} + \frac{0.7}{70} + \frac{0.8}{80} + \frac{0.9}{90} + \frac{1}{100} \right\}$$

$$\text{Slow} = \left\{ \frac{1}{0} + \frac{0.9}{10} + \frac{0.8}{20} + \frac{0.7}{30} + \frac{0.6}{40} + \frac{0.5}{50} + \frac{0.4}{60} + \frac{0.3}{70} + \frac{0.2}{80} + \frac{0.1}{90} + \frac{0}{100} \right\}$$

Using these variables, compute the membership function for the following linguistic terms:

(a) Very fast

(b) Very, very fast

(c) Highly fast (= minus very, very fast)

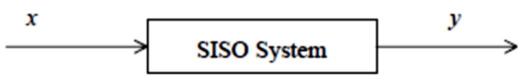
(d) Plus very fast

(e) Fairly fast $\left(= [\text{fast}]^{2/3}\right)$

(f) Not very slow and not very fast

(g) Slow or not very slow

6. Consider an unknown SISO system shown in Figure

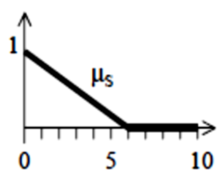


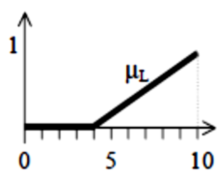
Suppose that this system is described by the following two rules:

$$R^1: \quad IF\ x \text{ is small } THEN\ y = a_1 x + b_1 x^2,$$

$$R^2: \quad IF\ x \text{ is large } THEN\ y = a_2 x + b_2 x^2,$$

where the small and large membership functions are given by Figures, respectively.



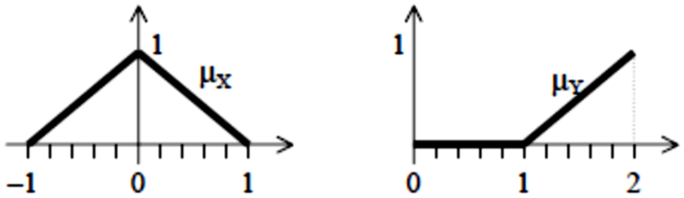


Now, a set of experimental data on the system input-output relation is available:

x	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
y	0.4	0.8	1.5	2.1	2.8	3.4	4.0	4.7	5.5	6.4	7.3

Use the least-squares approach to determine the unknown constant coefficients a_1, a_2, b_1 and b_2.

7. Consider a fuzzy system of the form $z = -x - y$,



   where the fuzzy inputs x and y have membership functions as shown in Figures, respectively. Find the interval Z and membership function μ_z for the fuzzy output z.

8. Consider the discrete-time SISO linear dynamic fuzzy system described by the rules

   $$R^1: \quad IF\ \ x(k-1)\ \text{ is small }\ THEN\ \ x(k+1) = -x(k) - \frac{1}{4}x(k-1),$$

   $$R^2: \quad IF\ \ x(k-1)\ \text{ is large }\ THEN\ \ x(k+1) = x(k) - \frac{1}{4}x(k-1),$$

   where the small and large membership functions for inputs are the same as those given in Exercise Q.6 above, and the weights for the output average are all equal to 1. Suppose that the system initial conditions are $x(-1) = x(0) = 1.0$. Let the weights in the output average be all identically equal to 1. Is this system asymptotically stable? Why? (or why not?)

REFERENCES

1. Chen, G. and Pham, T. T., Introduction to fuzzy systems, Chapman and Hall /CRC, Taylor and Francis, 2006.
2. Clark, D. W. (1992), Computer illogic. . . , Mirage Mag., University of New Mexico Alumni Association, Fall, pp. 12–13.
3. Lee, K. H., First Course on Fuzzy Theory and Applications, Springer, Berlin Heidelberg, NewYork, 2005.
4. Ross, T., Fuzzy logic with engineering applications, McGraw-Hill, New York, 1995.
5. Sivanandam, S. N., Sumathi, S. and Deepa, S. N., Introduction to Fuzzy Logic using MATLAB, Springer, Berlin Heidelberg, New York, 2007.
6. Zadeh, L. A. (1972), A fuzzy-set-theoretic interpretation of linguistic hedges, J. Cybern., vol. 2, pp. 4–34.
7. Zadeh, L. A. (1975), The concept of a linguistic variable and its application to approximate reasoning – I, Inf. Sci., vol. 8, pp. 199–249.
8. Zadeh, L. A. (1975b), The concept of a linguistic variable and its application to approximate reasoning – II, Inf. Sci., vol. 8, pp. 301–357.

Chapter-4

Fuzzy Control

4.1 INTRODUCTION

The "fuzzy wave" reached Europe after success stories in Japanese white goods and home electronics, popular but also industrial interest in fuzzy control. It was not being recognized as a serious discipline, but had been dramatically increasing since 1990. There are still, however, two predominant, extreme positions with respect to the benefits of fuzzy control. On one hand, many proponents of this technology claim that fuzzy control will revolutionize control engineering which promises major breakthrough and will be able to solve complex engineering problems with very little effort. On the other hand, many representatives of the control engineering community still proclaim the philosophy that "everything that can be done in fuzzy control can be done conventionally as well and announce a breakdown of the "fuzzy hype" in the near future.

The insight that neither of the two positions accounts for the real potential of fuzzy control is only gradually increasing. In this preface, we will try and explain that many of the expectations triggered mainly by popular press articles are exaggerated, but the fuzzy control has its advantages in many respects, and that it is here to stay: in most cases as an add-on to conventional technology, in some cases as an enabling technology. The main reason why it will become an integral part, of control engineering is that fuzzy control is a useful technology from an industrial, business oriented point of view, which is the one we are taking here.

When confronted with a control problem for a complicated physical process, a control engineer generally follows a relatively systematic design procedure. A simple example of a control problem is an automobile “cruise control” that provides the automobile with the capability of regulating its own speed at a driver-specified set-point (e.g., 55 mph). One solution to the automotive cruise control problem involves adding an electronic controller that can sense the speed of the vehicle via the speedometer and actuate the throttle position so as to regulate the vehicle speed as close as possible to the driver-specified value (the design objective). Such speed regulation must be accurate even if there are road grade changes, head winds, or variations in the number of passengers or amount of cargo in the automobile. After gaining an intuitive understanding of the plants dynamics and establishing the design objectives, the control engineer typically solves the cruise control problem by doing the following:

1. Developing a model of the automobile dynamics (which may model vehicle and power train dynamics, tire and suspension dynamics, the effect of road grade variations, etc.).
2. Using the mathematical model, or a simplified version of it, to design a controller (e.g., via a linear model, develop a linear controller with techniques from classical control).
3. Using the mathematical model of the closed-loop system and mathematical or simulation-based analysis to study its performance (possibly leading to redesign).
4. Implementing the controller via, for example, a microprocessor, and evaluating the performance of the closed-loop system (again, possibly leading to redesign).

This procedure is concluded when the engineer has demonstrated that the control objectives have been met, and the controller (the "product") is approved for manufacturing and distribution.

In the present chapter we first provide an overview of the standard approach to constructing a control system and identify a wide variety of relevant conventional control ideas and techniques in Section 4.2. We assume that the reader has at least some familiarity with conventional control. Our focus in this chapter is not only on introducing a variety of approaches to fuzzy control but also on comparing these to conventional control approaches to determine when fuzzy control offers advantages over conventional methods. Hence, to understand this chapter reader need to understand several ideas from conventional control e.g. classical control, state-space based design, the linear quadratic regulator, stability analysis, feedback linearization, adaptive control, etc. In section 3, we explain the benefits of fuzzy control from two technology perspectives.

The reader not familiar with conventional control to this extent will still find the chapter quite useful. In fact, we expect to what the appetite of such readers so that they become interested in learning more about conventional control. In Section 4.4 we outline a "philosophy" of fuzzy control where we explain the design methodology for fuzzy controllers, relate this to the conventional control design methodology, and highlight the importance of analysis and verification of the behavior of closed-loop fuzzy control systems. At the end of this chapter we will provide a list of books that can serve to teach such readers about these areas.

Further, we show how the fuzzy control designed methodology can be used to construct fuzzy controllers for challenging real-world applications. As opposed to "conventional" control approaches (e.g., proportional-integral-derivative (PID), lead-lag, and state feedback control) where the focus is on modeling and the use of this model to construct a controller that is described by differential equations, in fuzzy control we focus on gaining an intuitive understanding of how to best control the process, then we load this information directly into the fuzzy controller.

For an example, in the cruise control we may gather rules about how to regulate the vehicle's speed from a human driver. One simple rule that a human driver may provide is "If speed is lower than the set-point, then press down further on the accelerator pedal." Other rules may depend on the rate of the speed error increase or decrease, or may provide ways to adapt the rules when there are significant plant parameter variations (e.g., if there is a significant increase in the mass of the vehicle, tune the rules to press harder on the accelerator pedal). For more challenging applications, control engineers typically have to gain a very good understanding of the plant to specify complex rules that dictate how the controller should react to the plant outputs and reference inputs.

Basically, while differential equations are the language of conventional control, heuristics and "rules" about how to control the plant are the language of fuzzy control. This is not to say that differential equations are not needed in the fuzzy control methodology. Indeed, one of the main focuses of this Chapter will be on how "conventional" the fuzzy control methodology really is and how many ideas from conventional control can be quite useful in the analysis of the new class of control systems.

4.2 CONVENTIONAL CONTROL SYSTEM DESIGN

A basic control system is shown in Figure 1.1. The process (or "plant") is the object to be controlled. Its inputs are u(t), its outputs are y(t), and the reference input is r(t). In the cruise control problem, u(t) is the throttle input, y(t) is the speed of the vehicle, and r(t) is the desired speed that is specified by the driver. The plant is the vehicle itself. The controller is the computer in the vehicle that actuates the throttle based on the speed of the vehicle and the desired speed that was specified. In this section we provide an overview of the steps taken to design the controller shown in Figure 4.1. Basically, these are modeling, controller design, and performance evaluation.

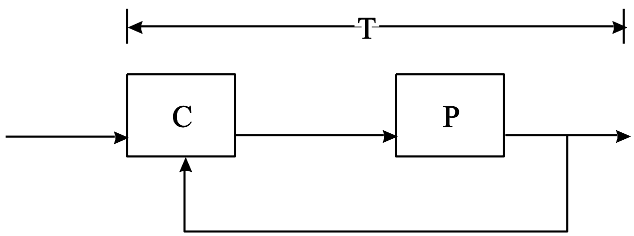


Figure 4.1 Control systems.

4.2.1 Mathematical Modeling

When a control engineer is given a control problem, often one of the first tasks that she or he undertakes is the development of a mathematical model of the process to be controlled, in order to gain a clear understanding of the problem. Basically, there are only a few ways to actually generate the model. We can use first principle of Newton's law in physics (e.g., F = ma) to write down a model. Another way is to perform "system identification" via the use of real plant data to produce a model of the system. Sometimes a combined approach is used where we use physics to write down a general differential equation that we believe represents the plant behavior, and then we perform experiments on the plant to determine certain model parameters or functions.

Often, more than one mathematical model is produced. A "truth model" is one that is developed to be as accurate as possible so that it can be used in simulation based evaluations of control systems. It must be understood, however, that there is never a perfect mathematical model for the plant. The mathematical model is an abstraction and hence cannot perfectly represent all possible dynamics of any physical process (e.g., certain noise characteristics or failure conditions). This is not to say that we cannot produce models that are "accurate enough" to closely represent the behavior of a physical system.

Usually, control engineers keep in mind that for control design they only need to use a model that is accurate enough to be able to design a controller that will work. Then, they often also need a very accurate model to test the controller in simulation (e.g., the truth model) before it is tested in an experimental setting. Hence, lower-order "design models" are also often developed that may satisfy certain assumptions (e.g., linearity or the inclusion of only certain forms of nonlinearities) yet still capture the essential plant behavior. Indeed, it is quite an art (and science) to produce good low-order models that satisfy these constraints. We emphasize that the reason we often need simpler models is that the synthesis techniques for controllers often require that the model of the plant satisfy certain assumptions (e.g., linearity) or these methods generally cannot be used.

Linear models such as the one in Equation (4.1) have been used extensively in the past and the control theory for linear systems is quite mature

$$\dot{x} = Ax + Bu \tag{4.1}$$

$$y = Cx + Du$$

In this case u is the m-dimensional input; x is the n-dimensional state ($\dot{x}$ = differentiation with respect to t); y is the p dimensional output; and A, B, C, and D are matrices of appropriate dimension. Such models, or transfer functions ($G(s) = C(sI - A)^{-1}B + D$ where s is the Laplace variable), are appropriate for use with frequency domain design techniques, the root-locus method, state-space methods, and so on. Sometimes it is assumed that the parameters of the linear model are constant but unknown, or can be perturbed from their nominal values (then techniques for "robust control" or adaptive control are developed).

Much of the current focus in control is on the development of controllers using nonlinear models of the plant of the form

$$\dot{x} = f(x, u) \tag{4.2}$$

$$y = g(x, u)$$

where the variables are defined as for the linear model and f and g are nonlinear functions of their arguments. One form of the nonlinear model that has received significant attention is

$$\dot{x} = f(x) + g(x)u \tag{4.3}$$

since it is possible to exploit the structure of this model to construct nonlinear controllers (e.g., in feedback linearization or nonlinear adaptive control). Of particular interest with both of nonlinear models is the case where f and g are not completely known and subsequent research focuses on robust control of nonlinear systems.

Discrete time versions of the above models are also used, and stochastic effects are often taken into account via the addition of a random input or other stochastic effects. Under certain assumptions we can linearize the nonlinear model in Equation (4.2) to obtain a linear one. In this case we sometimes think of the nonlinear model as the truth model, and the linear models are generated from it as control design models.

There are certain properties of the plant that the control engineer often seeks to identify early in the design process. For instance, the stability of the plant may be analyzed (e.g., to see if certain variables remain bounded). The effects of certain nonlinearities are also studied. The engineer may want to determine if the plant is "controllable" to see, for example, if the control inputs will be able to properly affect the plant; and "observable" to see, for example, if the chosen sensors will allow the controller to observe the critical plant behavior so that it can be compensated for, or if it is "non minimum phase."

The above mentioned properties will have a fundamental impact on our ability to design effective controllers for the system. In addition, the engineer will try to make a general assessment of how the plant behaves under various conditions, how the plant dynamics may change over time, and what random effects are present. Overall, this analysis of the plant's behavior gives the control engineer a fundamental understanding of the plant dynamics. This will be very valuable when it comes time to synthesize a controller.

4.2.3 Performance Objectives and Design Constraints

Controller design entails constructing a controller to meet the specifications. Often the first issue to address is whether to use open- or closed-loop control. If we can achieve our objectives with open-loop control, why turn to feedback control? Often, we need to pay for a sensor for the feedback information and there needs to be justification for this cost. Moreover, feedback can destabilize the system. Since open-loop controller may provide adequate performance, so we do not develop a feedback controller just because we used to developing feedback controllers.

Assuming we use feedback control, the closed-loop specifications (or "performance objectives") can involve the following factors:

Disturbance rejection properties (e.g., for the cruise control problem, that the control system will be able to dampen out the effects of winds or road grade variations). Basically, the need for disturbance rejection creates the need for feedback control over open-loop control; for many systems it is simply impossible to achieve the specifications without feedback (e.g., for the cruise control problem, if we have no measurement of vehicle velocity, how well could we regulate the velocity to the driver's set-point?).

- Insensitivity to plant parameter variations (e.g., for the cruise control problem, that the control system will be able to compensate for changes in the total mass of the vehicle that may result from varying the numbers of passengers or the amount of cargo).
- Stability (e.g., in the cruise control problem, to guarantee that on a level road the actual speed will converge to the desired set-point).
- Rise-time (e.g., in the cruise control problem, a measure of how long it takes for the actual speed to get close to the desired speed when there is a step change in the set-point speed).
- Overshoot (e.g., in the cruise control problem, when there is a step change in the set-point, how much the speed will increase above the set-point).

- Settling time (e.g., in the cruise control problem, how much time it takes for the speed to reach to within 1% of the set-point).
- Steady-state error (e.g., in the cruise control problem, if we have a level road, can the error between the set-point and actual speed actually go to zero; or if there is a long positive road grade, can the cruise controller eventually achieve the set-point).
- **While** these factors are used to characterize the technical conditions that indicate whether or not a control system is performing properly, there are other issues that must be considered that are often of equal or greater importance. These include the following:
- **Cost**: How much money will it take to implement the controller, or how much time will it take to develop the controller?
- **Computational complexity**: How much processor power and memory will it take to implement the controller?
- **Manufacturability**: Does our controller have any extraordinary requirements with regard to manufacturing the hardware that is to implement it?
- **Reliability**: Will the controller always perform properly? What is its "mean time between failures?"
- **Maintainability**: Will it be easy to perform maintenance and routine adjustments to the controller?
- **Adaptability**: Can the same design be adapted to other similar applications so that the cost of later designs can be reduced? In other words, will it be easy to modify the cruise controller to fit on different vehicles so that the development can be done just once?
- **Understandability**: Will the right people be able to understand the approach to control? For example, will the people that implement it or test it be able to fully understand it?
- **Politics**: Is your boss biased against your approach? Can you sell your approach to your colleagues? Is your approach too novel and does it thereby departs too much from standard company practice?

Remarks: Most often not only must a particular approach to control satisfy the basic technical conditions for meeting the performance objectives, but the above issues must also be taken into consideration — and these can often force the control engineer to make some very practical decisions that can significantly affect how, for example, the ultimate cruise controller is designed. It is important then that the engineer has these issues in mind early in the design process.

4.2.4 Controller Design

Conventional control has provided numerous methods for constructing controllers for dynamic systems. Some of these are listed below, and we provide a list of references at the end of this chapter for the reader who is interested in learning more about any one of these topics.

- Proportional-integral-derivative (PID) control: Over 90% of the controllers in operation today are PID controllers (or at least some form of PID controller like a P or PI controller). This approach is often viewed as simple, reliable, and easy to understand. Often, like fuzzy controllers, heuristics are used to tune PID controllers (e.g., the Zeigler-Nichols tuning rules).
- Classical control: Lead-lag compensation, Bode and Nyquist methods, root-locus design, and so on.
- State-space methods: State feedback, observers, and so on.
- Optimal control: Linear quadratic regulator, use of Pontryagin's minimum principle or dynamic programming, and so on.
- Robust control: H2 or H∞ methods, quantitative feedback theory, loop shaping, and so on.
- Nonlinear methods: Feedback linearization, Lyapunov redesign, sliding mode control, back stepping, and so on.
- Adaptive control: Model reference adaptive control, self-tuning regulators, nonlinear adaptive control, and so on.
- Stochastic control: Minimum variance control, linear quadratic gaussian (LQG) control, stochastic adaptive control, and so on.
- Discrete event systems: Petri nets, supervisory control, infinitesimal perturbation analysis, and so on.

Basically, these conventional approaches to control system design offer a variety of ways to utilize information from mathematical models on how to do good control. Sometimes they do not take into account certain heuristic information early in the design process, but use heuristics when the controller is implemented to tune it (tuning is invariably needed since the model used for the controller development is not perfectly accurate). Unfortunately, when using some approaches to conventional control, some engineers become somewhat removed from the control problem (e.g., when they do not fully understand the plant and just take the mathematical model as given), and sometimes this leads to the development of unrealistic control laws. Sometimes in conventional control, useful heuristics are ignored because they do not fit into the proper mathematical framework, and this can cause problems.

4.2.5 Performance Evaluation

The next step in the design process is to perform analysis and performance evaluation. Basically, we need performance evaluation to test that the control system that we design does in fact meet the closed-loop specifications (e.g., for "commissioning" the control system). This can be particularly important in safety-critical applications such as a nuclear power plant control or in aircraft control. However, in some consumer applications such as the control of a washing machine or an electric shaver, it may not be as important in the sense that failures will not imply the loss of life (just the possible embarrassment of the company and cost of warranty expenses), so some of the rigorous evaluation methods can sometimes be ignored. Basically, there are three general ways to verify that a control system is operating properly:

(1) Mathematical analysis based on the use of formal models.

(2) simulation-based analysis that most often uses formal models.

(3) Experimental investigations on the real system.

Next, we discuss these ways one by one

(1) Mathematical Analysis

In mathematical analysis we may seek to prove that the system is stable, that it is controllable, or that other closed-loop specifications such as disturbance rejection, rise-time, overshoot, settling time, and steady state errors have been met. Clearly, however, there are several limitations to mathematical analysis. First, it always relies on the accuracy of the mathematical model, which is never a perfect representation of the plant, so the conclusions that are reached from the analysis are in a sense only as accurate as the model that they were developed from and second, there is a need for the development of analysis techniques for even more sophisticated nonlinear systems since existing theory is somewhat lacking for the analysis of complex nonlinear (e.g., fuzzy) control systems, particularly when there are significant nonlinearities, a large number of inputs and outputs, and stochastic effects. These limitations do not make mathematical analysis useless for all applications, however. Often it can be viewed as one more method to enhance our confidence that the closed-loop system will behave properly, and sometimes it helps to uncover fundamental problems with a control design.

(2) Simulation-Based Analysis

In simulation-based analysis we seek to develop a simulation model of the physical system. This can entail using physics to develop a mathematical model and perhaps real data can be used to specify some of the parameters of the model (e.g., via system identification or direct parameter measurement). The simulation model can often be made quite accurate, and we can even include the effects of implementation considerations such as finite word length restrictions. As discussed above, often the simulation model ("truth model") will be more complex than the model that is used for control design because this "design model" needs to satisfy certain assumptions for the control design methodology to apply (e.g., linearity or linearity in the controls). Often, simulations are developed on digital computers, but there are occasions where an analog computer is still quite useful (particularly for real time simulation of complex systems or in certain laboratory settings).

Regardless of the approach used to develop the simulation, there are always limitations on what can be achieved in simulation-based analysis. First, as with the mathematical analysis, the model that is developed will never be perfectly accurate. Also, some properties simply cannot be fully verified via simulation studies. For instance, it is impossible to verify the asymptotic stability of an ordinary differential equation via simulations since a simulation can only run for a finite amount of time and only a finite number of initial conditions can be tested for these finite length trajectories. Basically, however, simulation-based studies can enhance our confidence that properties of the closed-loop system hold, and can offer valuable insights into how to redesign the control system before you spend time implementing the control system.

(3) Experimental Investigations

To conduct an experimental investigation of the performance of a control system, you implement the control system for the plant and test it under various conditions. Clearly, implementation can require significant resources (e.g., time, hardware), and for some plants you would not even consider doing an implementation until extensive mathematical and simulation-based investigations have been erformed.

However, the experimental evaluation does shed some light on some other issues involved in control system design such as cost of implementation, reliability, and perhaps maintainability. The limitations of experimental evaluations are, first, problems with the repeatability of experiments, and second, variations in physical components, which make the verification only approximate for other plants that, are manufactured at other times. On the other hand, experimental studies can go a long way toward enhancing our confidence that the system will actually work since if you can get the control system to operate, we will see one real example of how it can perform.

4.3 FUZZY CONTROL FROM AN INDUSTRIAL PERSPECTIVE

In order to be able to explain what, the benefits of fuzzy control are, we first will reconsider it from two different technological perspectives.

4.3.1 Fuzzy Control: Real-time Expert Systems or Nonlinear Control Systems?

Out of the many possible ways of looking at fuzzy control, we consider the following two as most appropriate. A fuzzy control system is a real-time expert system, implementing a part of a human operator's or process engineers expertise which does not lend itself to being easily expressed in PID-parameters or differential equations but rather in situation/action rules. This view is introduced in the next sections of this chapter. However, fuzzy control differs from main-stream expert system technology in several aspects. One main feature of fuzzy control systems is their existence at two distinct levels: first, there are if-then rules and qualitative, fuzzy variables and values such as if pressure is high and slightly increasing Then energy supply is medium negative.

These fuzzy values such as "slightly increasing" and fuzzy operators such as "and" are compiled into very elementary numerical objects and algorithms: function tables, interpolation, comparators, etc. The existence of this compiled level is the basis for fast, real-time implementations, as well as for embedding fuzzy control into the essentially numerical environment of conventional control. In artificial intelligence, the field of qualitative physics follows a similar approach, using qualitative variables and so-called "'qualitative differential equations," too. A main technical difference is their usage of crisp intervals corresponding to rectangular fuzzy membership functions for representing qualitative values.
The major difference between fuzzy control and qualitative physics, however, can be found in their relationship to classical engineering discipline a: while the "Qualitative Physics Manifesto" aims at replacing differential equation-based techniques and solving the whole problem with artificial intelligence methods, fuzzy control has right from the beginning been considered an extension to existing technology, seeking hybrid solutions by enhancing control engineering where it is needed and where it makes sense.
In fact, most of the inventors of fuzzy control have a strong control engineering or systems theory background. From their perspective, fuzzy control can be seen as a heuristic, and modular way for defining nonlinear, table-based control systems. Reconsider the rule above: it is nothing but an informal "nonlinear PD-element." A collection of such rules can be used and, in fact, results in the definition of a nonlinear transition function, without the need for defining each entry of the table individually, and without necessarily knowing the closed form representation of that function.
One way to combine fuzzy and PID-control then is to use a linear PID system round the set point, where it does its job, and to "delinearize" the system in other areas by describing the desired behavior or control strategy with fuzzy rules. A representation theorem due to Kosko [120] state that any continuous nonlinear function can be approximated as exactly as needed with a finite set of fuzzy variables, values, and rules. This theorem describes the representational power of fuzzy control in principle, but it does not answer the questions, how many rules are needed and how they can be found, which are of, Course essential to real-world problems and solutions. In many cases, relatively small and simple systems will do, and that is why already several hundreds of real, industrial applications of fuzzy control exist. What exactly are the benefits of using fuzzy control?

4.3.2 The Benefits of Fuzzy Control

Considering the existing applications of fuzzy control, which range from very small, micro-controller based systems in home appliances to large-scale process control systems, the advantages of using fuzzy control usually fall into one of the following four categories:

(1) To implement expert knowledge for a higher degree of automation

In many cases of industrial process control, e.g., in chemical industries, the degree of automation is quite low. There is a variety of basic, conventional control loops, but a human operator is needed during the starting or closing phase, for parametrizing the controllers, or for switching between different control modules. The knowledge of this operator is usually based on experience and does not lend itself to being expressed in differential equations. It is often rather of the type "if the situation is such and such, I should do the following." In this case, fuzzy control offers a method for representing and implementing the expert's knowledge. As an example, consider digester management in paper industries. A Portuguese company has implemented a fuzzy based digester management system on top of its process control software. Some twenty-five rules are used for expressing the core of the operator's control strategy.

The main advantage of the higher degree of automation that has been achieved this way is the resulting consistent control strategy around the clock, yielding a substantial reduction (up to 60%) of the variation of product quality. Kurt her more, subsequent optimizations of the software system led to a significant reduction of energy and base material consumption. All in all, a return on the investment needed for the fuzzy control software package ant) the knowledge base of the actual control system was achieved after only a few months.

(2) Robust nonlinear control

Consider the following problem: a robot arm with several links has to move objects with different masses along a predefined path. Since there are good and exact models of this system available, it is not too big a problem to realize a PID-controller that works pretty well for known masses within a narrow range. Substantial parameter changes, however, or major external disturbances, lead to a sharp decrease in performance. In the presence of such disturbances, **PID-**systems usually are faced with a trade-off between fast reactions with significant overshoot or smooth but slow reactions, or they even run into problems in stabilizing the system at all.
In this case, fuzzy control offers ways in implement simple but robust solutions that cover a wide range of system parameters and that can cope with major disturbances. Its particular case is called fuzzy gliding mode controller and its implementation is discussed in chapter 5. It exhibits similar performance for a given mass only slight variations, but that outperforms the PID-solution as soon as major variations are imposed.
There are two other advantages which can be achieved by using fuzzy or rather hybrid solutions which do not relate t_n the performance of the control system but rather to other aspects - I hat are nevertheless important from a business standpoint.

(3) Reduction of development and maintenance time

In some cases, two different groups of experts are involved in the development of a control system. Field experts know the application problem and how to design appropriate control strategies. Often, however, they are not electronics experts who know the bits and bytes of numerical algorithms and micro-controller implementations. The actual system realization then is carried out by electronics and systems programming engineers, who on the other hand arc not familiar with the application problem. This often results in communication problems between these two groups of people, and hence in delays and extended development times. Fuzzy control, which was pointed out above "lives"[1] at two levels of abstraction, offers languages at both levels of expertise: the symbolic level is appropriate for describing the application engineer?' strategies, while the compiled level is well understood by the electronic engineers. Since there is a well-defined, formal translation between those levels, a fuzzy based approach can help reducing communication problems.

As an example, consider idle-speed control in automotive electronics, where sophisticated control strategies have to be implemented and run in time-critical situations on simple 8-bit micro-controllers. Besides major variations in system parameters due to mass series production and to aging problems, which require extensive system tuning, communication problems between engine and microcontroller specialists lead to very long development times.

We recently had the chance to drive an experimental car with two idle-speed controllers, a conventional PID-system, and a fuzzy control system. The major control goal was to keep a constant idle-speed of 8130 rpm. Independently of disturbances imposed by different road conditions or additional power consumers such as power steering and air conditioning. Practically no differences could be observed concerning the system's behavior, but while almost two man-years went into the development of the conventional solution, development time for the fuzzy solution was around six months only.

(4) Marketing and patents

In Japan, "fuzzy" has become one of the most popular words. In particular in the realm of home appliances, the label "fuzzy-controlled" raises positive associations in the sense of modern, high quality, or user friendly. As a matter of fact, the use of fuzzy control has contributed heavily to the functionality of quite a few home appliances. The "one-touch-button"[1] washing machine, for example, constitutes a major advance with respect to the large control panels of the older washing machines. In this machine, an additional infrared sensor is used to assess the quality and the quantity of dirt in the laundry, and a fuzzy evaluation procedure for these sensor data and subsequent fuzzy rule based inference is used to determine the details of the washing program. In 1990, Japanese manufacturers reported sales of fuzzy controlled home appliances in the range of several billion US dollars.

This and other successful applications of fuzzy control, however, have led to a state where merely using fuzzy control - whether or not it really makes sense from a technological point of view has become a marketing argument. Therefore, the majority of newly introduced home appliances are labeled fuzzy or "neuro-fuzzy." We do not believe that the mere use of fuzzy control, independently of its actual merits, will play a role as a marketing argument for industrial applications. In a few cases, we expect fuzzy control to be used in industrial applications even when it is not actually needed, just to enable companies to demonstrate their technological competence in fuzzy control.

There is yet another mason, however, for using fuzzy control even if it does not improve system performance or reduce development costs. In some business areas, the patent situation is such that it is hard to come up with new solutions using conventional technology without violating or at least interfering with a competitor's patent. Even if the patent turns out to be untenable, this usually leads to significant delays in introducing new products. In some cases, using fuzzy control for a qualitatively equivalent solution will - and actually does -help io by-pass existing patents.

So far, we have listed several success stories of fuzzy control, highlighting the benefits. It should be clear that in most cases, these are no pure fuzzy solutions but rather hybrid solutions, using fuzzy control to augment conventional technology. We should, however, address the question: where are the limits?

4.3.3 The Limits of Fuzzy Control

In some articles in the popular press, examples such as those above are overly generalized to statements about much logic, as a really fabulous technique. Some of these statements are definitely misleading, for some others we simply need a much broader basis for induction in terms of many more applications in order to verify or deny such general statements. So let us reconsider the examples above and their erroneous generalizations as discussed below:

(i) Fuzzy control leads to a higher degree of automation for complex, ill-structured processes

This is true in many cases, but only if there is relevant knowledge about the process and its control available that can be well expressed in terms of fuzzy logic. There are processes for which that kind of knowledge simply is not at all or not to the necessary extent available.

(ii) Fuzzy controllers are more robust than conventional controllers

Again, there are many applications, where the use of fuzzy control - pure or in combination with, say, FID-control - resulted in simple yet highly robust control systems. There are other cases, too. We know of two attempts to control air conditioning system? With fuzzy logic, with only minor differences in structure and knowledge base: one turned out to be highly robust even in the presence of major disturbances, the other one was unstable. It is not yet fully understood for which kinds of control engineering problems fuzzy control really leads to improved robustness and stability, and which are the relevant design choices that affect these properties.

It is not true, however, as opponents of fuzzy control often argue, that there are no stability criteria available for fuzzy control systems. Such criteria are discussed in several different places in this hook. We also have to realize that in this respect, fuzzy control competes with nonlinear conventional control, where stability issues are not as easy to handle as for simple linear systems.

(iii) Fuzzy control reduces development time

Again, yes in many cases it does, and the example above in automotive electronics is a real one. It cannot he expected, however, as claimed in a number of publications, that using fuzzy control will help to solve really complex problems within a couple of weeks. The reduction of development time for results from the fact that there is often knowledge about a process or about control strategies available which ran he naturally expressed in fuzzy control but which is hard to encode in differential equations or conventional control algorithms. This knowledge remains to be acquired, encoded, tested, arid debugged, and for all but the simplest problems, a strong control engineering background will be needed to realize good solutions.

In many ongoing projects in European industries, using fuzzy instead of conventional control will actually increase development time, just because the application engineers working on those projects are not yet familiar with the new technology. The shortage of well-trained personnel in fuzzy control is one of the major bottlenecks that prevent a broader exploitation of this new technology at this time. Fuzzy control is only beginning to be integrated into the academic: curricula, and it will take several years until a sufficient number of engineers familiar with Uv/r/y control will he available from tin: universities. Suite in dust ry cannot wait for this to happen - in particular because the situation in Japan is completely different an increasing number of special seminars and in-house training programs have been started.

(iv) Products using fuzzy control sell better

It is true for home appliances in Japan, however, we do not expect that this phenomenon will carry over to Europe. Marketing arguments will have to focus on user-friendliness, additional functionality, and environmental aspects such as energy savings or say, reduction of fresh water consumption of washing machines. Since fuzzy logic can in many cases contribute to these points, we expect the technology to become positively associated with these advantages, but merely labeling a product "fuzzy-controlled'[1] will not become a major sales argument, in some cases, in particular in home electronics and video equipment, the use of fuzzy control is not being mentioned at all.

A fuzzy auto-focus system, for example, does not seem to be a quality label Therefore; most suppliers of such systems use the term digital or intelligent auto-focus, even if it is implemented in fuzzy logic. In the realm of industrial process control and related fields, marketing strategies differ from those in mass markets such as home appliances. While it is important to companies to demonstrate their technological competence in fuzzy control, the actual decisions for using fuzzy control or any other technique will be made in terms of its contribution to a higher degree of automation, better product quality, cost reduction, or other business-relevant aspects.

4.3.4 On When to Use Fuzzy Control

Having gone through a few examples where fuzzy control is being successfully used, we immediately cautioned about unjustified, too high expectations. But when, then, should fuzzy control be considered?

A satisfactory answer to this question cannot yet be formulated. What we would like to have at our disposal is a systematic decision procedure for analyzing a given problem and inferring whether and if so which variant of fuzzy control should be used based on problem characteristics. At this time, the statistical basis in terms of number of successful applications is not yet broad enough to generate useful generalizations. This should not be a surprise because the industrial and commercial use of fuzzy control did not really start until the late 1980s, compared to the immense number of existing conventional control systems with an accumulated experience of many decades. At tins point, we can only formulate a few guidelines on whether to use fuzzy control or not.

First, it is clear that if there exists already a successful fuzzy control baser) solution to a problem similar to the one at hand, you are on the safe side. In terms of business advantages, however, these are the less interesting cases since somebody else might, already commercially exploit that solution.

Next, if we have a good PlD-solution, where system performance, development and maintenance costs as well as marketing policy are really satisfactory, stick to it.

Finally, if we are not satisfied with our existing solution with respect to any one of these; criteria, or if you have to deal with a problem that could not be solved so far, it is important to analyze the reason why. If there is knowledge available about the system or process that could be used to improve your solution but that is hard to encode in terms of conventional control such as differential equations and PID-parameters and give fuzzy control a try.

In many cases, this knowledge is of a more heuristic and qualitative nature and is easily expressible in terms of fuzzy if-then rules. We know of many cases of exactly this kind.

Just to name one there is no exact and complete mathematical understanding of the processes going on in cooling systems in power generation devices. Hence, algorithmic solutions to infer diagnoses such as system leakage from sensor data exist only to a very limited extent, and the expertise of system operators and engineers is always needed to monitor such cooling systems both the evaluation of data such as pressure and change of pressure and the combination of symptoms to proper situation identification however can be naturally expressed in fuzzy terms such as "normal pressure, but quickly increasing'[1] and corresponding if-then rules. In this particular case, it took only a matter of months to design and implement a fuzzy logic based monitoring system for cooling system leakage that outperforms the existing classical solutions. It is important to note, however, that the fuzzy operators and membership functions being used in this system differ from the very simple min/max-inference and triangular membership functions that are predominant in most of the elementary control applications. We will come back to that issue later on.

4.3.5 The Market Place

So far, public interest in fuzzy control has been raised mainly by llic immense number of fuzzy-controlled Japanese home appliances and the corresponding number of sales, ranging far above one billion US dollars. We expect fuzzy control to penetrate into European and US-American white good markets in 1993, presumably to a lower degree than that which can be observed in Japan. Another main area of current commercial activities in fuzzy control is automotive electronics. Here again, mainly Japanese suppliers have already brought fuzzy controlled anti-skid modules and engine management systems into the stage of series production. We expect this trend to continue, and major European and American car manufacturers have significant ongoing activities in fuzzy control, though we do not know of any product originating from Europe or the US that has already reached the market place.

Focusing on such spectacular products as "one-touch-button washing machines," however, it is often overlooked that, the first real applications, of fuzzy control were realized in the realm of industrial process control, such as cement kiln operation. An increasing number of applications are being reported in this area, and several major suppliers of process control hardware and software are now offering fuzzy control capabilities, for example, in programmable logic controllers or higher level process control systems. In the long run, we see the major application potential in this area, while in terms of market share and commercial benefits, home appliances, automotive electronics, and other microcontroller based applications will dominate the fuzzy control business for the next couple of years. As the technology matures, however, we foresee an increasing number of applications in complex systems engineering, the focus of fuzzy control moving from elementary control problems to higher levels in the system hierarchy such as supervisory control, monitoring and diagnosis, and logistics problems. It is important to note that also one of the major future industries, telecommunications, has started investigating fuzzy control for communication systems, and that several pilot projects have been started concerning, for example, routing and overload handling problems.

An increasing number of market growth studies are trying to estimate and forecast the economic relevance of fuzzy control. While these studies are worthwhile to carry out, it is very hard to delineate exactly the contribution of fuzzy control to the monetary value of a product. Consider the example of a washing machine, where a major part of the electronic control is realized in fuzzy logic. It is certainly not correct to take the price of the whole washing machine, but it also would be misleading to consider just the - usually standard - microcontroller or the number of lines of code realizing the fuzzy part of the system. Our assessment of the contemporary and future market relevance of fuzzy control is the following: we estimate that about 10-15% of all electric and electronic engineering applications will benefit from the use of fuzzy control, being roughly evenly split into fuzzy control as an enhancing and as an enabling technology. Considering the huge market volume of this area, and also its expected further growth, it becomes evident that fuzzy control is a major technology from a business point of view. In addition to that, there is a still small but increasing number of applications of fuzzy logic in other, purely information technology oriented fields such as databases, expert systems in the financial sector, and so on.

We should not, however, commit the same error as was done with "classical" artificial intelligence systems, whereafter some initial success in relatively narrow application areas it was believed that artificial intelligence could be used as a universal stand-alone technology to solve just about every problem in information processing. We expect that for a long time the main application potential of fuzzy logic will stay within the realm of fuzzy control, in an increasingly broader sense of "control." This is not to say that the relevance of fuzzy logic will not expand to other areas, where there are already a few spots such as data compression, but this is still mainly research-oriented work, whereas fuzzy control has already gained a significant role in daily industrial practice and its full potential in this area is by no means yet exploited or even known.

So far, we have discussed the relevance of fuzzy control applications as a business. There is yet another business area in fuzzy control, relating to the technology itself as a product, in terms of special hardware and software. So far, the majority of all existing applications is purely software-based. Although there are a large number of special fuzzy logic processors available and offered by semiconductor firms or small specialized companies, yet we expect that this situation will not change within the next years.

Using special hardware in the sense of general-purpose fuzzy logic processors or coprocessors is and will only be justified in the following situations; the application is extremely time-critical, and the extra costs for an additional fuzzy processor can be recovered either from the sheer number of items (such as in automotive electronics) or from the fact that the system as a whole is large and expensive enough that the costs for a special processor are negligible (this might be

the case when using a fuzzy processor for, say, a pattern recognition task in a complex plant automation system). Therefore, most suppliers of micro-controllers, PLCs, and other control equipment focus on developing specialized fuzzy control software for standard devices, and on integrating these packages into the existing development environments.
Besides major electric/electronic and industrial automation companies, an increasing number of small and specialized software companies is offering fuzzy control development tool kits for a broad range of standard processors. Our advice in general is to use as long as possible pure software solutions, even in higher programming languages such as C. The reason for this is that software offers a much higher degree of flexibility and, in most cases, the solutions are hybrid, i.e., there is always and for a standard processor to implement the "non-fuzzy" parts of the system. Experience so far has shown also that the elementary fuzzy operations such as function table look-up and interpolation, comparators, and basic arithmetic functions are simple enough that fuzzy control algorithms can be optimized to run even very lime-critical applications without specialized hardware. It may be the case that the role of hardware will become more important in the future, when more complex operators will be needed, to adequately express and implement the expertise for control strategies. This might trigger the development of more sophisticated fuzzy processors and also semi- and full-custom integrated hardware designs. The point is, however, that it is not yet obvious which operators will be needed and that existing special hardware does not support any but the simplest fuzzy algorithms.

4.3.6 What Needs to Be Done?

Reconsider the example of the cooling system monitoring problem discussed above. During the system design and knowledge acquisition phase, it turned out that the human expert's usage of, for example, the AND-operation when combining symptoms could not be adequately expressed with the usual fuzzy logic interpretation of AND as a minimum operation. Also the fuzzy membership functions representing values such as "normal pressure" turned out to be more - sophisticated than the simple triangular functions predominant in elementary control applications.

Several conclusions can be inferred from this example: first of all, it was extremely helpful to have a software tool available which supports experimental system implementation, evaluation, and tuning, and which offers the user ways to combine predefined standard operations with his or her own special code. There is an increasing number of such software tools available which - though not yet completely matured provide valuable support. They are being steadily improved and are being integrated into standard software development packages for automatic control.

Secondly, considering the major pact of existing applications of fuzzy control, it can be realized that the "application-proven" part, let us say the "first generation" of fuzzy control, exploits only a very small fragment of fuzzy logic theory. In many cases of more complex, ill-structured problems, this first generation technology is not sufficiently strong to represent and implement the knowledge needed for powerful solutions. It is often possible, though, to improve the technology additively by exploiting a larger part of the readily available theory.

The example above and many others make clear that the development of a mature, more powerful second generation technology still requires basic and experimental research. This research will have to focus on the questions of which parts of the still unused theory will eventually gain practical relevance, where does the theory itself need extensions, and which are useful ways of integrating fuzzy control with conventional and also other modern control algorithms such as neural network based approaches. We are convinced that only application-driven, experimental research can be sufficiently focused and guided in the right directions lo make good progress. It simply does not make any sense to define a new operator "Fuzzy-And. Version 137" and to discuss its mathematical properties without being triggered by the fact that none of the existing operators can be used for a particular application.

Thus, we believe that industrial research - as opposed to pure university-based research - will play a major role. Only industrial research labs have direct access to real-world problems, which is essential for proceeding in this field. This is not to say that the universities are not needed: most importantly, fuzzy control has to become an integral part of educational curricula in electronic and control engineering. Furthermore, academic people can and must contribute to the long-term, fundamental issues in fuzzy control theory. But the major push in fuzzy control within the next years will be application-driven and will necessarily have to be carried out in industrial research labs.

Also considering first generation technology, there are still a few bottle-necks hindering industry from a broader exploitation of the application potential of fuzzy control, in the first place, we need a bettor and more systematic design and analysis methodology for fuzzy control applications, supporting the whole life-cycle from initial decision making all the way through to deployment and maintenance, focusing on the following issue: which are the proper design choices given an analysis of the problem, and how do variations of system parameters affect the system's performance. It should be clear that we cannot expect a universal design and optimization strategy for fuzzy control which is of any practical use. Such a universal theory does not exist for conventional control engineering either. Instead, we have to proceed from the few isolated spots where we already know exactly how to design a fuzzy control algorithm to clusters of problems and related design methodologies.

Besides the necessary paper and experimental work which is needed to advance this methodology, we need a much broader basis of experience in terms of successful and of unsuccessful applications. We expect this methodology to gradually emerge as experience is being accumulated.

The absence of such a coherent and systematic methodology makes another main problem for first-generation fuzzy control even more serious: the lack of well-trained and experienced "fuzzy control engineers." It means that engineers who want to or have to get into this new technology have only limited support and guidance and must by and large rely on their own experience gained through experimental work.

4.4 FUZZY CONTROL SYSTEM DESIGN

What, then, is the motivation for turning to fuzzy control? Basically, the difficult task of modeling and simulating complex real-world systems for control systems development, especially when implementation issues are considered, is well documented. Even if a relatively accurate model of a dynamic system can be developed, it is to be often too complex to use in controller development, especially for many conventional control design procedures that require restrictive assumptions for the plant (e.g., linearity). It is for this reason that in practice conventional controllers are often developed via simple models of the plant behavior that satisfy the necessary assumptions, and via the ad-hoc tuning of relatively simple linear or nonlinear controllers. It is well understood (although sometimes forgotten) that heuristics enter the conventional control design process as long as you are concerned with the actual implementation of the control system. It must be acknowledged, moreover, that conventional control engineering approaches that use appropriate heuristics to tune the design have been relatively successful. You may ask the following questions:

How much of the success can be attributed to the use of the mathematical model and conventional control design approach? And how much should be attributed to the clever heuristic tuning that the control engineer uses upon implementation? And if we exploit the use of heuristic information throughout the entire design process, can we obtain higher performance control systems?

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human's heuristic knowledge about how to control a system. In this section we seek to provide a philosophy of how to approach the design of fuzzy controllers. This will lead us to provide a motivation for, and overview of, the entire book. The fuzzy controller block diagram is given in Figure 4.2, where we show a fuzzy controller embedded in a closed-loop control system. The plant outputs are denoted by y(t), its inputs are denoted by u(t), and the reference input to the fuzzy controller is denoted by r(t).

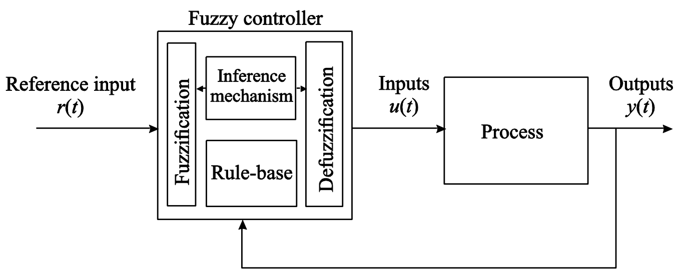


Fig. 4.2 design of fuzzy controllers

The fuzzy controller has four main components:

(1) The "rule-base" holds the knowledge, in the form of a set of rules, of how best to control the system.

(2) The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be.

(3) The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base. And

(4) The defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant.

Basically, we will view the fuzzy controller as an artificial decision maker that operates in a closed-loop system in real time. It gathers plant output data y(t), compares it to the reference input r(t), and then decides what the plant input u(t) should be to ensure that the performance objectives will be met.

To design the fuzzy controller, the control engineer must gather information on how the artificial decision maker should act in the closed-loop system. Sometimes this information can come from a human decision maker who performs the control task, while at other times the control engineer can come to understand the plant dynamics and write down a set of rules about how to control the system without outside help.

These "rules" basically say, "If the plant output and reference input are behaving in a certain manner, then the plant input should be some value." A whole set of such "If-Then" rules is loaded into the rule-base, and an inference strategy is chosen, then the system is ready to be tested to see if the closed-loop specifications are met.

This brief description provides a very high-level overview of how to design a fuzzy control system. Below we will expand on these basic ideas and provide more details on this procedure and its relationship to the conventional control design procedure.

4.4.1 Modeling Issues and Performance Objectives

People working in fuzzy control often say that "a model is not needed to develop a fuzzy controller, and this is the main advantage of the approach." However, will a proper understanding of the plant dynamics be obtained without trying to use first principles of physics to develop a mathematical model? And will a proper understanding of how to control the plant be obtained without simulation-based evaluations that also need a model? We always know roughly what process we are controlling (e.g., we know whether it is a vehicle or a nuclear reactor), and it is often possible to produce at least an approximate model, so why not do this?

For a safety-critical application, if we do not use a formal model, then it is not possible to perform mathematical analysis or simulation-based evaluations. Is it wise to ignore these analytical approaches for such applications? Clearly, there will be some applications where we can simply "hack" together a controller (fuzzy or conventional) and go directly to implementation. In such a situation there is no need for a formal model of the process; however, is this type of control problem really so challenging that fuzzy control is even needed? Could a conventional approach (such as PID control) or a "table look-up" scheme work just as well or better, especially considering implementation complexity?

Overall, when we carefully consider the possibility of ignoring the information that is frequently available in a mathematical model, it is clear that it will often be unwise to do so. Basically, then, the role of modeling in fuzzy control design is quite similar to its role in conventional control system design. In fuzzy control there is a more significant emphasis on the use of heuristics, but in many control approaches (e.g., PID control for process control) there is a similar emphasis. Basically, in fuzzy control there is a focus on the use of rules to represent how to control the plant rather than ordinary differential equations (ODE). This approach can offer some advantages in that the representation of knowledge in rules seems more lucid and natural to some people. For others, though, the use of differential equations is more clear and natural. Basically, there is simply a "language difference" between fuzzy and conventional control: ODEs are the language of conventional control, and rules are the language of fuzzy control.

The performance objectives and design constraints are the same as the ones for conventional control that we summarized above, since we still want to meet the same types of closed-loop specifications. The fundamental limitations that the plant provides affect our ability to achieve high-performance control, and these are still present just as they were for conventional control (e.g., non minimum phase or unstable behavior still present's challenges for fuzzy control).

4.4.2 Fuzzy Controller Design

Fuzzy control system design essentially amounts to

(1) Choosing the fuzzy controller inputs and outputs,

(2) Choosing the preprocessing that is needed for the controller inputs and possibly post processing that is needed for the outputs, and

(3) Designing each of the four components of the fuzzy controller shown in Figure 4.2. As we will see in the next chapter, there are standard choices for the fuzzification and defuzzification interfaces. Moreover, most often the designer settles on an inference mechanism and may use this for many different processes. Hence, the main part of the fuzzy controller that we focus on for design is the rule-base.

The rule-base is constructed so that it represents a human expert "in-the-loop." Hence, the information that we load into the rules in the rule-base may come from an actual human expert who has spent long time learning how best to control the process. In other situations there is no such human expert, and the control engineer will simply study the plant dynamics (perhaps using modeling and simulation) and write down a set of control rules that makes sense. As an example, in the cruise control problem discussed above it is clear that anyone who has experience driving a car can practice regulating the speed about a desired set-point and load this information into a rule-base. For instance, one rule that a human driver may use is "If the speed is lower than the set-point, then press down further on the accelerator pedal."

A rule that would represent even more detailed information about how to regulate the speed would be "If the speed is lower than the set-point AND the speed is approaching the set-point very fast, and then release the accelerator pedal by a small amount." This second rule characterizes our knowledge about how to make sure that we do not overshoot our desired goal (the set-point speed). Generally speaking, if we load very detailed expertise into the rule-base, we enhance our chances of obtaining better performance.

4.4.3 Performance Evaluation of Fuzzy Control

Each and every idea presented in Section 4.2.5 on performance evaluation for conventional controllers applies here as well. The basic reason for this is that a fuzzy controller is a nonlinear controller — so many conventional modeling, analysis (via mathematics, simulation, or experimentation), and design ideas apply directly.

Since fuzzy control is a relatively new technology, it is often quite important to determine what value it has relative to conventional methods. Unfortunately, few have performed detailed comparative analyses between conventional and intelligent control that have taken into account a wide array of available conventional methods (linear, nonlinear, adaptive, etc.); fuzzy control methods (direct, adaptive, supervisory); theoretical, simulation, and experimental analyses; computational issues; and so on.

Moreover, most work in fuzzy control to date has focused only on its advantages and has not taken a critical look at what possible disadvantages there could be to using it (hence the reader should be cautioned about this when reading the literature). For example, the following questions are cause for concern when you employ a strategy of gathering heuristic control knowledge:

- Will the behaviors that are observed by a human expert and used to construct the fuzzy controller include all situations that can occur due to disturbances, noise, or plant parameter variations?
- Can the human expert realistically and reliably foresee problems that could arise from closed-loop system instabilities or limit cycles?
- Will the human expert be able to effectively incorporate stability criteria and performance objectives (e.g., rise-time, overshoot, and tracking specifications) into a rule-base to ensure that reliable operation can be obtained?

These questions may seem even more troublesome in view of the following:

(1) If the control problem involves a safety-critical environment where the failure of the control system to meet performance objectives could lead to loss of human life or an environmental disaster, or

(2) If the human expert's knowledge implemented in the fuzzy controller is somewhat inferior to that of the very experienced specialist we would expect to design the control system (different designers have different levels of expertise)

Clearly, then, for some applications there is a need for a methodology to develop, implement, and evaluate fuzzy controllers to ensure that they are reliable in meeting their performance specifications. This is the basic theme and focus of this book.

4.4.4 Application Areas

Fuzzy systems have been used in a wide variety of applications in engineering, science, business, medicine, psychology, and other fields. For instance, in engineering some potential application areas include the following:

- Aircraft/spacecraft: Flight control, engine control, a vionic systems, failure diagnosis, navigation, and satellite attitude control.
- Automated highway systems: Automatic steering, braking, and throttle control for vehicles.
- Automobiles: Brakes, transmission, suspension, and engine control.
- Autonomous vehicles: Ground and underwater.
- Manufacturing systems: Scheduling and deposition process control.

- Power industry: Motor control, power control/distribution, and load estimation.
- Process control: Temperature, pressure, and level control, failure diagnosis, distillation column control, and desalination processes.
- Robotics: Position control and path planning.

This list is only representative of the range of possible applications for the methods of this book. Others have already been studied, while still others are yet to be identified.

4.5 CONCLUSIONS

In this chapter we have provided an overview of the approaches to conventional and fuzzy control system design and have showed how they are quite similar in many respects. In this chapter our focus will be not only on introducing the basics of fuzzy control, but also on performance evaluation of the resulting closed-loop systems. Moreover, we will pay particular attention to the problem of assessing what advantages fuzzy control methods have over conventional methods. Generally, this must be done by careful comparative analyses involving modeling, mathematical analysis, simulation, implementation, and a full engineering cost-benefit analysis (which involves issues of cost, reliability, maintainability, flexibility, lead-time to production, etc.).

Some of our comparisons will involve many of these dimensions while others will necessarily be more cursory. Although it is not covered in this book, we would expect the reader to have as prerequisite knowledge of the basic ideas in conventional control (at least, those typically covered in a first course on control). After completion of this chapter, the reader should then understand the following:

- The distinction between a "truth model" and a "design model."
- The basic definitions of performance objectives (e.g., stability and overshoot).
- The general procedure used for the design of conventional and fuzzy control systems, which often involves modeling, analysis, and performance evaluation.
- The importance of using modeling information in the design of fuzzy controllers and when such information can be ignored.
- The idea that mathematical analysis provides proofs about the properties of the mathematical model and not the physical control system.
- The importance, roles, and limitations of mathematical analysis, simulation-based analysis, and experimental evaluations of performance for conventional and fuzzy control systems.
- The basic components of the fuzzy controller and fuzzy control system.
- The need to incorporate more sophisticated reasoning strategies in controllers and the subsequent motivation for adaptive and supervisory fuzzy control.

Essentially, this is a checklist for the major topics of this chapter. The reader should be sure to understand each of the above concepts before proceeding to later chapters, where the techniques of fuzzy control are introduced. We find that if we have a solid high-level view of the design process and philosophical issues involved, we will be more effective in developing control systems.

The more that we understand about conventional control, the more we will be able to appreciate some of the finer details of the operation of fuzzy control systems. We realize that all readers may not be familiar with all areas of control, so next we provide a list of books from which the major topics can be learned. There are many good texts on classical control [10, 20, 11, 7, 6, 2]. State-space methods and optimal and multivariable control can be studied in several of these texts and also in [12, 1, 4, 25]. Robust control is treated in [8, 9, 30]. Nonlinear control is covered in [17, 5, 28, 29, 16]; stability analysis in [27, 26]; and adaptive control in [15, 18, 3, 13,]. System identification is treated in [24] (and in the adaptive control texts), and optimal estimation and stochastic control are covered in [19, 23, 22, 14]. A relatively complete treatment of the field of control is in [21].

For more recent work in all these areas, see the proceedings of the IEEE Conference on Decision and Control, the American Control Conference, the European Control Conference, the International Federation on Automatic Control World Congress, and certain conferences in chemical, aeronautical, and mechanical engineering.

Major journals to keep an eye on include the IEEE Transactions on Automatic Control, IEEE Transactions on Control Systems Technology, IEEE Control Systems Magazine, Systems and Control Letters, Automatica, Control Engineering Practice, International Journal of Control, and many others. Extensive lists of references for fuzzy and intelligent control are provided at the end.

EXERCISE -4

1. Describe example of a control problem of an automobile "cruise control" (George Viote).
2. Explain, How the fuzzy control design methodology can be used to construct fuzzy controllers for challenging real-world applications?
3. Write basics of Conventional Control System Design.
4. When a control engineer is given a control problem, how he developed a mathematical model of the process?
5. Explain Performance Objectives and Design Constraints for constructing a controller to meet the specifications.
6. Write numerous methods used for constructing controllers for dynamic systems.
7. How we conduct an experimental investigation of the performance of a control system.
8. Write the Benefits of Fuzzy Control.
9. Write the Limits of Fuzzy Control.
10. Write the Application Areas of Fuzzy Control.

BIBLIOGRAPHY

1. B. D. O. Anderson and J. B. Moore. Optimal Control: Linear Quadratic Methods. Prentice-Hall, Englewood Cliffs, NJ, 1990.
2. K. J. ˚Astr¨om and B. Wittenmark. Computer Controlled Systems: Theory and Design. Prentice-Hall, Englewood Cliffs, NJ, 1984.
3. K. J. ˚Astr¨om and B. Wittenmark. Adaptive Control. Addison-Wesley, Reading, MA, 1995.
4. M. Athans and P. L. Falb. Optimal Control. McGraw-Hill, New York, 1966.
5. D. P. Atherton. Nonlinear Control Engineering: Describing Function Analysis and Design. Van Nostrand Reinhold, Berkshire, England, 1982.
6. J. H. D'Azzo and C. H. Houpis. Linear Control System Analysis and Design:Conventional and Modern. McGraw-Hill, New York, 1995.
7. R. C. Dorf and R. H. Bishop. Modern Control Systems. Addison-Wesley,Reading, MA, 1995.
8. J. C Doyle, B. A. Francis, and A. R. Tannenbaum. Feedback Control Theory. Macmillan, New York, 1992.
9. S. S. Farinwata and G. Vachtsevanos. Stability analysis of the fuzzy controllerdesigned by the phase portrait assignment algorithm. In Proc. of the 2nd IEEE Int. Conf. on Fuzzy Systems, pages 1377–1382, San Francisco, CA, March 1993.
10. G. F. Franklin, J. D. Powell, and A. Emami-Naeini. Feedback Control of Dynamic Systems. Addison-Wesley, Reading, MA, 1994.
11. G. F. Franklin, J. D. Powell, and M. L.Workman. Digital Control of Dynamic Systems. Addison-Wesley, Reading, MA, 1990.
12. B. Friedland. Control System Design. McGraw-Hill, New York, 1986.
13. G. C. Goodwin and K. S. Sin. Adaptive Filtering Prediction and Control. Prentice-Hall, Englewood Cliffs, NJ, 1984.
14. [63] M. J. Grimble and M. A. Johnson. Optimal Control and Stochastic Estimation. Wiley, New York, 1988.
15. P. A. Ioannou and J. Sun. Robust Adaptive Control. Prentice-Hall, Englewood Cliffs, NJ, 1996.
16. A. Isidori. Nonlinear Control Systems. Springer-Verlag, New York, third edition, 1995.
17. H. K. Khalil. Nonlinear Systems. Prentice-Hall, Englewood Cliffs, NJ, 2ndedition, 1996.
18. M. Krstic, I. Kanellakopoulos, and P. Kokotovic. Nonlinear and Adpative Control Design. Wiley, New York, 1995.
19. P. R. Kumar and P. P. Varaiya. Stochastic Systems: Estimation, Identification, and Adaptive Control. Prentice-Hall, Englewood Cliffs, NJ, 1986.
20. B. C. Kuo. Automatic Control Systems. Prentice-Hall, Englewood Cliffs, NJ, 1991.
21. W. Levine, editor. The Control Handbook. CRC Press, Boca Raton, FL, 1996.
22. F. L. Lewis. Optimal Control. Wiley, New York, 1986. 486 BIBLIOGRAPHY
23. F. L. Lewis. Optimal Estimation. Wiley, New York, 1986.
24. L. Ljung. System Identification: Theory for the User. Prentice-Hall, Englewood Cliffs, NJ, 1987.

25. J. M. Maciejowski. Multivariable Feedback Design. Addison-Wesley, Reading, MA, 1989.
26. A. N. Michel and R. K. Miller. Qualitative Analysis of Large Scale Dynamical Systems. Academic Press, New York, 1977.
27. R. K. Miller and A. N. Michel. Ordinary Differential Equations. Academic Press, New York, 1982.
28. [189] J. E. Slotine and W. Li. Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, NJ, 1991.
29. V. I. Utkin. Sliding Modes in Control Optimization. Springer-Verlag, Berlin, 1992.
30. K. Zhou, J. C. Doyle, and K. Glover. Robust and Optimal Control. Prentice- Hall, Englewood Cliffs, NJ, 1996.

Chapter-5

Fuzzy Knowledge Base Control (FKBC)

5.1 INTRODUCTION

The primary goal of control engineering is to distill and apply knowledge about how to control a process so that the resulting control system may reliably and safely achieve high-performance operation. In this chapter we show how fuzzy logic provides a methodology for representation and implementation of our knowledge about how best to control a process.

After study of this chapter, it is expected that the reader should be able to design and simulate a fuzzy control system. This will move the reader a long way toward implementation of fuzzy controllers since we provide pointers on how to overcome certain practical problems encountered in fuzzy control system design and implementation (e.g., coding the fuzzy controller to operate in real-time, even with large rule-bases).

In Section 5.2 we begin with a "gentle" (tutorial) introduction via a simple inverted pendulum control problem. We focus on the construction and basic mechanics of operation of a two-input one-output fuzzy controller with the most commonly used fuzzy operations. In section 5.3 we explain the difference between fuzzy and human control, building on our understanding of the two-input one-output fuzzy controller. Sections 5.4 and 5.5 cover the Fuzzy Rule for a mathematical characterization of general fuzzy system with two inputs and outputs. Fuzzy quantifications such as membership function and general fuzzification, inference, and defuzzification strategies for Fuzzy Control are discussed in section 5.6. In Section 5.7 fusion of fuzzy and PID controller are given for Supervision of conventional controllers, Correction of conventional controllers, and Coordination of control loop set-points. In section 5.8 Inversion of fuzzy systems has been explained. In section 5.9 we illustrate some typical steps in the fuzzy control design process and explain how to write a computer program that will simulate the actions of a fuzzy controller discussed in Section 5.5.We explain various issues encountered in implementing fuzzy controllers in Section 5.10. In the end designing some problems as an exercise is given.

5.2 CLASSICAL FUZZY CONTROL IN INVERTED PENDULUM

Basically Inverted pendulum is used to control Voltammeter Ammeter. Here, we want to design and analyze a fuzzy controller for the simplified version of the inverted pendulum system as shown in figure 5.1. The differential equation describing the system is

$$-ml^2\, d^2\theta/dt^2 + (m\, l\, g) \sin(\theta) = \tau = u(t), \qquad \text{....(5.1)}$$

where m is the mass of the pole located at the tip point of the pendulum, I is the length of the pendulum, θ is the deviation angle from vertical in the clockwise direction, $\tau = u\ (t)$ is the torque applied to the pole in the counterclockwise direction (u (t) is the control action), m time, and g is the acceleration due to gravity.
Assuming $x_1 = \theta$ and $x_2 = d\theta/dt$ to be the state variables, the state-space representation for the non-linear system defined by Eq. (5.1) is given by

$$dx_1/dt = x_2, \qquad dx_2/dt = (g/1) \sin (x_1) - (1/ml^2)\, u\ (t)$$

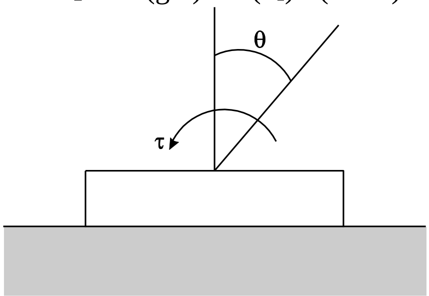


Fig. 5.1: Inverted pendulum control problem

For very small rotationθ, we have sin (θ), where θ is measured in radians. We get

$$dx_1/dt = x_2$$
$$dx_2/ dt = (g/1)\, x_1 - (1/ml^2)\, u\ (t)$$

If x_1 is measured in degrees and x_2 is measured in degrees per second, by choosing l = g and $m = 180 / (\pi\ .\ g^2)$, the linearized and discrete time state-space equation can be represented as matrix difference equations,

$$x_1 (k + 1) = x_1(k) + x_2(k)$$
$$x_2 (k + 1) = x_1(k) + x_2(k) - u(k)$$

For this problem we assume the universe of discourse for the two variables to be

$- 2° \le x_1 \le 2°$ and -5 dps $\le x_2 \le 5$ dps (dps = degrees per second).

Step 1: We construct three membership functions for x_1 on its universe, that is, for the values positive (P), zero (Z), and negative (N), as shown in following figure:

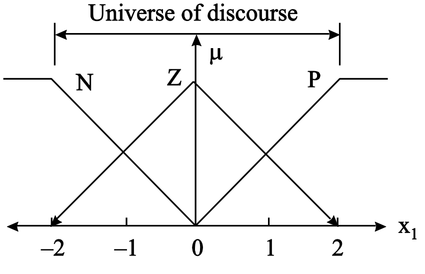


Fig. 5.2

We then construct three membership functions for x_2 on its universe, that is, for the values positive (P), zero (Z) and negative (N), as shown below.

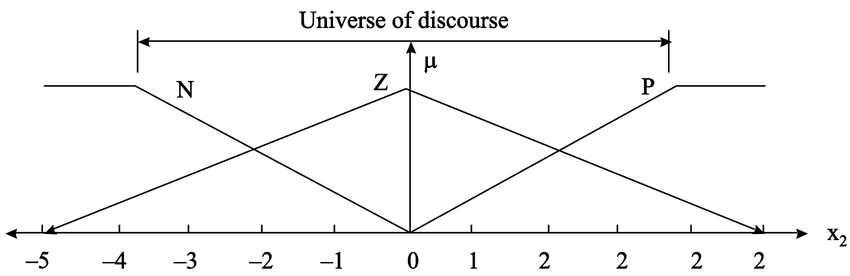


Fig. 5.3: Input x_2 partitioned

Step 2: To partition the control space (output), we will construct five membership functions for u (k) on its universe, which is -24 ≤ u (k) ≤ 24, as shown in figure below.

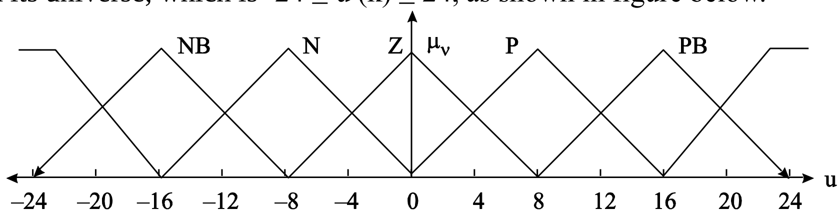


Fig. 5.4: Output, u (k), partitioned into seven partitions (only five used).

Step 3: We then construct nine rules (even though we may not need this many) in a 3 × 3 FAM table, as shown below, for this system, which would involve θ and dθ/dt in order to stabilize the inverted pendulum system. The entries in this table are the control actions, u (k).

Table 5.1 FAM Table

x_2 / x_1	**P**	**Z**	**N**
P	PB	P	Z
Z	P	Z	N
N	Z	N	NB

Step 4: Using the rules expressed in above table, we will now conduct a simulation of this control problem. To start the simulation, we will use the following crisp initial conditions:

$x_1(0) = 1°$ and $x_2(0) = -4$dps

Then we will conduct four cycles of simulation using the matrix difference equations, above, for the discrete steps 0 < k < 3. Each simulation cycle will result in membership functions for the two input variables. The FAM table will produce a membership function for the control action, u (k). We will defuzzify the membership function for the control action using the centroid method and then use the recursive difference equations to solve for new values of x_1 and x_2. Each simulation cycle after k = 0 will begin with the previous values of x_1 and x_2 as the input conditions to the next cycle of the recursive difference equations. Figures 5.5 and 5.6 given below show the initial conditions for x_1 and x_2, respectively. From the FAM table

If (x_1 = P) and (x_2 = Z), then (u = P) min (0.5, 0.2) = 0.2 (P)

If(x_1 = P) and (x_2 = N), then (u = Z) min (0.5, 0.8) = 0.5 (Z)

If (x_1 = Z) and (x_2 = Z), then (u = Z) min (0.5, 0.2) = 0.2 (Z)

If (x_1 = Z) an d(x_2 = N), then(u = N) min (0.5, 0.8) = 0.5 (N)

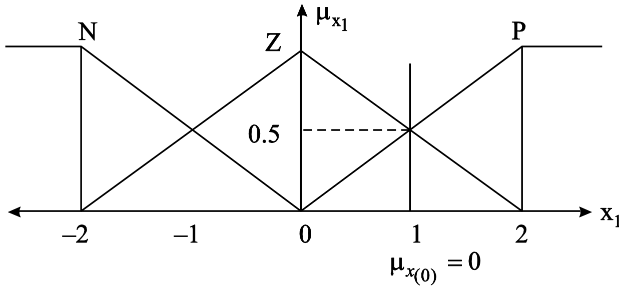


Fig. 5.5: Initial condition for x_1

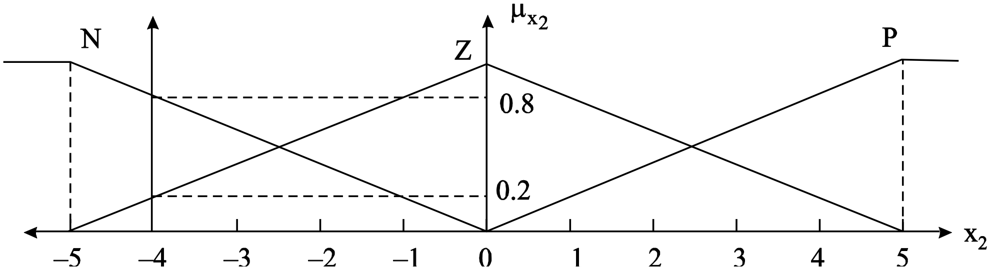


Fig. 5.6: Initial condition for x_2

Figure 5.7 below shows the union of the truncated fuzzy consequents for the control variable, u.

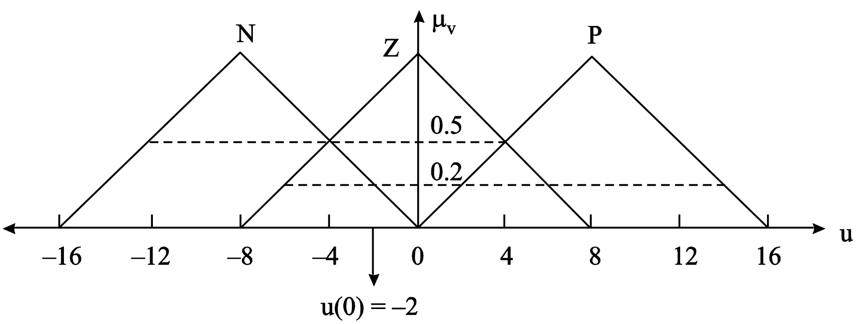


Fig. 5.7: Union of fuzzy consequents fired by rules

The final form with the defuzzified control value is illustrated in the following figure (5.8):

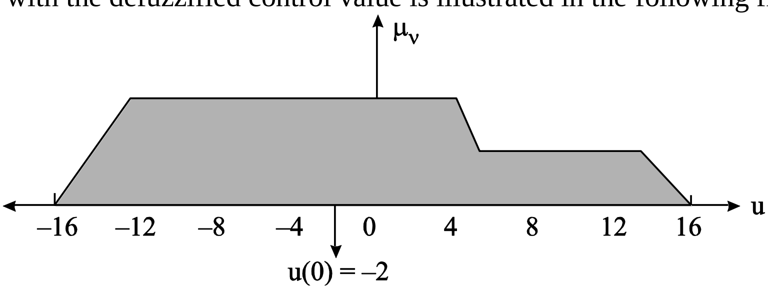


Fig. 5.8: Union of fuzzy consequents and defuzzified value

We have completed the first cycle of the simulation. Now, we take the value of the defuzzified control variable (i.e., u = -2). Using the system equations, we find the initial conditions for next iteration as

$x_1(1) = x_1(0) + x_2(0) = 1 - 4 = -3$ and

$x_2(1) = x_1(0) + x_2(0) - u(0) = 1 - 4 - (-2) = -1$

From this, we get the initial conditions for the second cycle as $x_1(1) = -3$ and $x_2(1) = -1$, which are shown graphically in following figures 5.9 and 5.10:

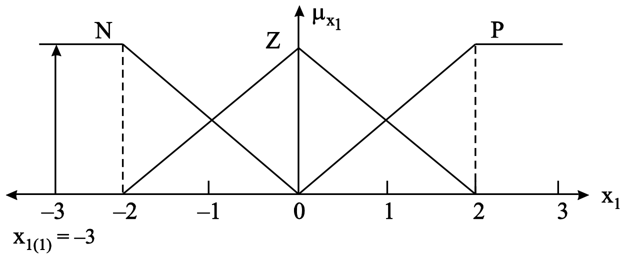


Fig.5.9: Initial condition for second cycle for x_1

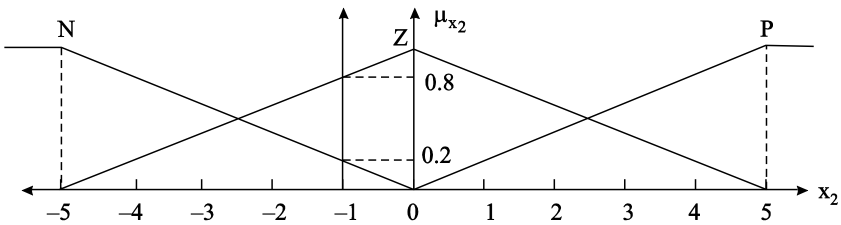


Fig. 5.10: Initial conditions for second cycle for x_2

From the FAM table 5.1, we have

if (x_1 = N) and (x_2 = N), then (u = NB) and min (1, 0.2) = 0.2 (NB)

and

if (x_2 = N) and (x_2 = Z), then (u = N) and min (1, 0.8) = 0.8 (N).

The union of the fuzzy consequents and the resulting defuzzified output are shown below. The defuzzified value is u = - 9.6.

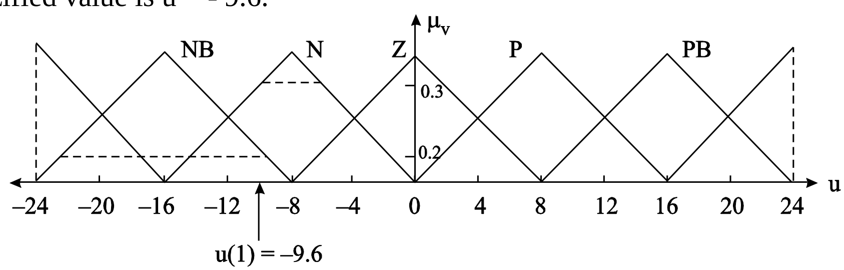


Fig. 5.11: Truncated consequents and defuzzified output for second cycle.

We now use u = - 9.6 to find the initial conditions for third cycle iteration.

$x_1(2) = x_1(1)\ x_2(1) = -3 - 1 = -4$ AND

$x_2(2) = x_1(1)x_2(1) - u(1) = -3 - 1 - (-9.6) = +5.6$

Thus, we get initial conditions $x_1(2) = -4$ and $x_2(2) = 5.6$, which are shown graphically in following figures:

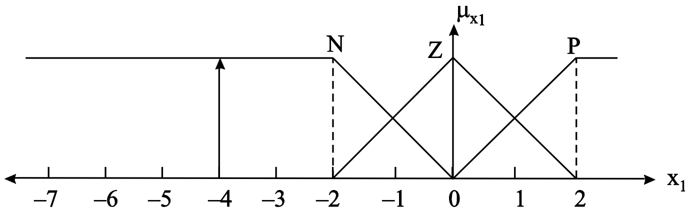


Fig. 5.12: Initial condition for third cycle for x_1

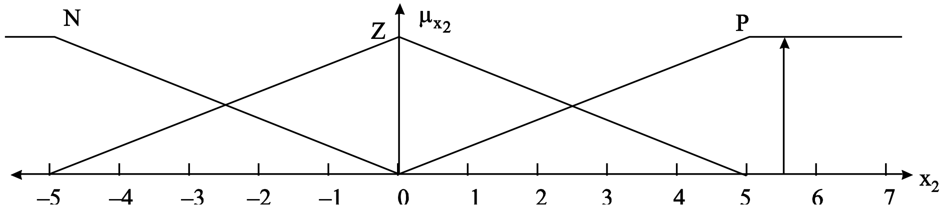


Fig. 5.13: Initial condition for third cycle for x_2

From the FAM table 5.1, if (x_1 = N) and (x_2 = P), then (u = Z) and min (1, 1) = 1(Z).

The resulting consequents and defuzzified control variable, u(z) = 0.0, are shown below

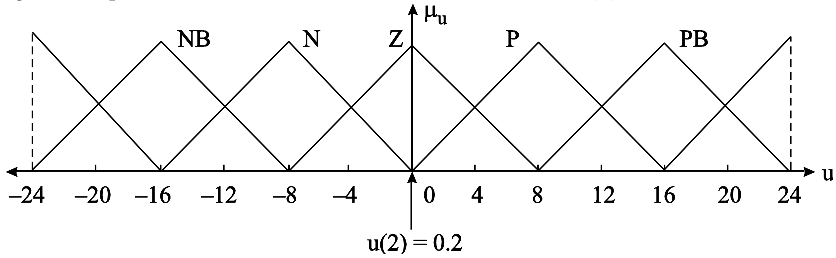


Fig. 5.14: Defuzzified output for third cycle of simulation

For the next iteration

$x_1(3) = x_1(2) + x_2(2) = -4 + 5.6 = 1.6$ and

$x_2(3) = x_1(2) + x_2(2) - u(2) = -4 + 5.6 - (0.0) = 1.6$

Initial conditions $x_1(3) = 1.6$ and $x_2(3) = 1.6$ are shown graphically in figures below:

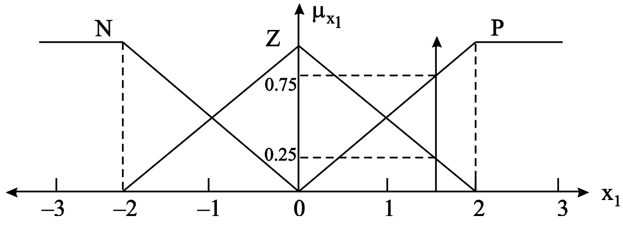


Fig. 5.15: Initial condition for next iteration for x_1

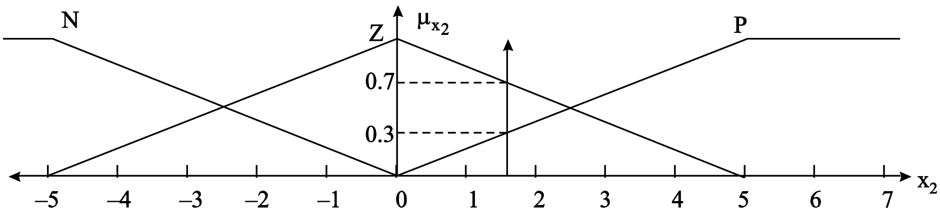


Fig. 5.16: Initial condition for next iteration for x_2

From FAM table 5.1, we get
if (x_1 = Z) and (x_2 = P), then (u = P) and min (0.25, 0.3) = 0.3 (P),
if (x_1 = Z) and (x_2 = Z), then (u = Z) and min (0.25, 0.7) = 0.25 (Z),
if (x_1 = P) and (x_2 = P), then (u = PB) and min (0.75, 0.3) = 0.3 (PB),
and
if (x_1 = P) and (x_2 = Z), then (u = P)and min (0.75, 0.7) = 0.7 (P)
These conditions are shown graphically in following figure and the defuzzified value is u = 5.28.

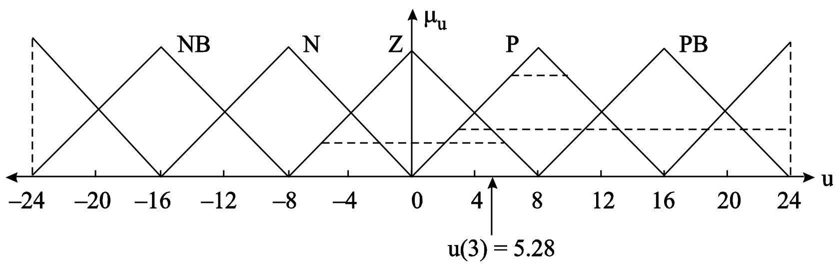


Fig. 5.17: Defuzzified output for next iteration.

5.3 FUZZY AND HUMAN CONTROL AN INTRODUCTION

A block diagram of a fuzzy control system discussed in Chapter 4 is shown in Figure 5.18. Sometimes a fuzzy controller is called a "fuzzy logic controller" (FLC) or even a "fuzzy linguistic controller" since it uses fuzzy logic in the quantification of linguistic descriptions. The fuzzy controller is composed of the following four elements:

1. A rule-base (a set of If-Then rules), which contains a fuzzy logic quantification of the expert's linguistic description of how to achieve good control.
2. An inference mechanism (also called an "inference engine" or "fuzzy inference" module), which emulates the expert's decision making in interpreting and applying knowledge about how best to control the plant.
3. A fuzzification interface, which converts controller inputs into information that the inference mechanism can easily use to activate and apply rules.
4. A defuzzification interface, which converts the conclusions of the inference mechanism into actual inputs for the process.

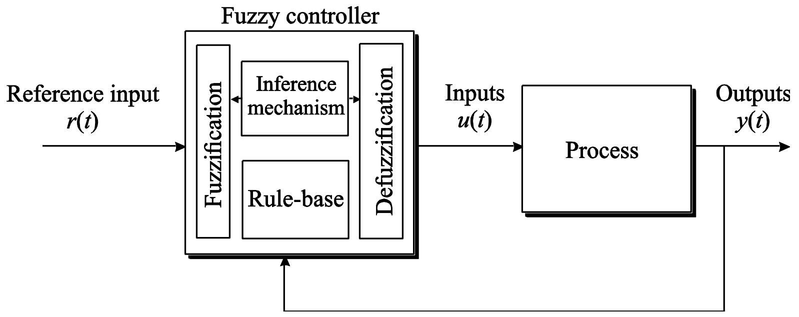


Fig. 5.18 Fuzzy control

We introduce each of the components of the, "inverted pendulum on a cart" by fuzzy and human controller for a simple problem of balancing as shown in Figure 5.19 and 5.20 respectively.

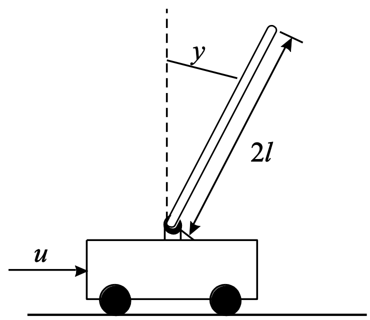


Fig. 5.19 Inverted pendulum Fuzzy control

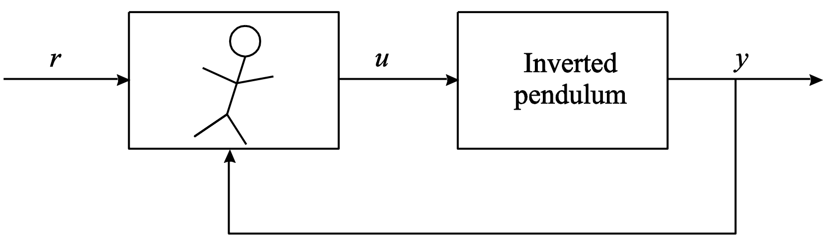


Fig. 5.20 Inverted pendulum human control

Here, y denotes the angle that the pendulum makes with the vertical (in radians), l is the half-pendulum length (in meters), and u is the force input that moves the cart (in Newton's). We will use r to denote the desired angular position of the pendulum. The goal is to balance the pendulum in the upright position (i.e., r = 0) when it initially starts with some nonzero angle off the vertical (i.e., y = 0). This is a very simple and academic nonlinear control problem, and many good techniques already exist for its solution. Indeed, for this standard configuration, a simple PID controller works well even in implementation.

Consider a human-in-the-loop whose responsibility is to control the pendulum, as shown in Figure 5.20. The fuzzy controller is to be designed to automate how a human expert who is successful at this task would control the system. First, the expert tells us (the designers of the fuzzy controller) what information she or he will use as inputs to the decision-making process. Suppose that for the inverted pendulum, the expert (this could be you!) says that she or he will use e (t) = r (t) – y (t) and $\frac{d\,e(t)}{dt}$ as the variables on which to base decisions. Certainly, there are many other choices (e.g., the integral of the error e could also be used) but this choice makes good intuitive sense. Next, we identify the controlled variable. For the inverted pendulum, we are allowed to control only the force that moves the cart, so the choice here is simple.

Once the fuzzy controller inputs and outputs are chosen, we will determine what the reference inputs are. For the inverted pendulum, the choice of the reference input r = 0 is clear. In some situations, r is some nonzero constant to balance the pendulum in the off-vertical position. To do this, the controller must maintain the cart at a constant acceleration so that the pendulum will not fall.

After all the inputs and outputs are defined for the fuzzy controller, we can specify the fuzzy control system. The fuzzy control system for the inverted pendulum, with our choice of inputs and outputs, is shown in Figure 5.21. Now, within this framework we seek to obtain a description of how to control the process.

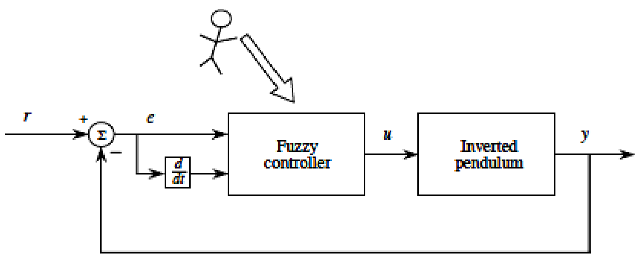


Fig. 5.21 Fuzzy controller for Inverted pendulum

Suppose that the human expert shown in Figure 5.20 provides a description of how best to control the plant in some natural language (e.g., English). We seek to take this "linguistic" description and load it into the fuzzy controller, as indicated by the arrow in Figure 5.21.

The linguistic description provided by the expert can generally be broken into several parts. For the inverted pendulum, "error" describes e (t), "change-in-error" describes $\frac{de(t)}{dt}$ and "force" describes u (t).

Note that we use quotes to emphasize that certain words or phrases are linguistic descriptions. Regardless, the choice of the linguistic variable has no impact on the way that the fuzzy controller operates; it is simply a notation that helps to facilitate the construction of the fuzzy controller via fuzzy logic. Just as e (t) takes on a value of, for example, 0.1 at t = 2 ⇒ e (2) = 0.1, "linguistic values." That is, the values that linguistic variables take on over time change dynamically. Suppose for the pendulum example that "error," "change-in-error," and "force" take on the following values: "Negative Large (NL), Negative Small (NS), Zero (ZE), Positive Small (PS) and Positive Large (PL). For an even shorter description we could use integers: -2, -1, 0, 1 and 2 to represent Negative Large (NL), Negative Small (NS), Zero (Z), Positive Small (PS) and Positive Large (PL) respectively.

This is a particularly appealing choice for the linguistic values since the descriptions are short and nicely represent that the variable we are concerned with has a numeric quality. We shall find the use of this type of linguistic value quite convenient and hence will give it the special name, "linguistic-numeric value."

In the next subsection we shall study how to quantify knowledge about how to control the pendulum using linguistic descriptions is controlled.

For the inverted pendulum each of the following statements quantifies a different configuration of the pendulum (refer back to Figure 5.19):

- The statement "error is PL" can represent the situation where the pendulum is at a significant angle to the left of the vertical.
- The statement "error is NS" can represent the situation where the pendulum is just slightly to the right of the vertical, but not too close to the vertical to justify quantifying it as "zero" and not too far away to justify quantifying it as "NL."
- The statement "error is ZE" can represent the situation where the pendulum is very near the vertical position
- The statement "error is PL **and** change-in-error is PS" can represent the situation where the pendulum is to the left of the vertical and, since $\frac{dy}{dt} < 0$ the pendulum is moving away from the upright position (note that in this case the pendulum is moving counterclockwise).
- The statement "error is NS **and** change-in-error is PS" can represent the situation where the pendulum is slightly to the right of the vertical and, since $\frac{dy}{dt} < 0$, the pendulum is moving toward the upright position (note that in this case the pendulum is also moving counterclockwise).

It is important for the reader to study each of the cases above to understand how the expert's linguistics quantifies the dynamics of the pendulum (actually, each partially quantifies the pendulum's state).

Overall, we see that to quantify the dynamics of the process we need to have a good understanding of the physics of the underlying process we are trying to control. While for the pendulum problem, the task of coming to a good understanding of the dynamics is relatively easy; this is not the case for many physical processes. Quantifying the process dynamics with linguistics is not always easy, and certainly a better understanding of the process dynamics generally leads to a better linguistic quantification. Often, this will naturally lead to a better fuzzy controller provided that we can adequately measure the system dynamics so that the fuzzy controller can make the right decisions at the proper time.

5.4 RULES

Next, we will use the above linguistic quantification to specify a set of rules (a rule-base) that captures the expert's knowledge about how to control the plant.

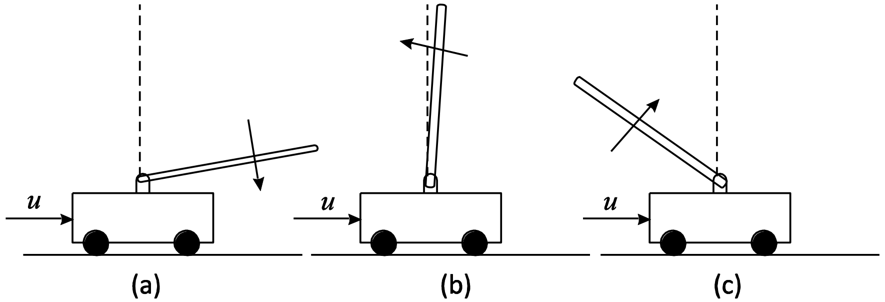


Fig. 5.22 Fuzzy rule base for Inverted pendulum

In particular, for the inverted pendulum in the three positions shown in Figure 5.22, we have the following rules:

Rule	Statement	Description
1	**If** error is NL **and** change-in-error is NL **Then** force is PL.	This rule quantifies the situation in Figure 2.5(a) where the pendulum has a large positive angle and is moving clockwise
2	**If** error is ZE **and** change-in-error is PS **Then** force is NS.	This rule quantifies the situation in Figure 2.5(b) where the pendulum has nearly a zero angle with the vertical (a linguistic quantification of zero does not imply that e(t) = 0 exactly) and is moving counterclockwise
3	**If** error is PL **and** change-in-error is NS **Then** force is NS.	This rule quantifies the situation in Figure 2.5(c) where the pendulum is far to the left of the vertical and is moving clockwise

Action is to be taken to start the pendulum moving in the proper direction in each rule:

Rule 1: we should apply a strong positive force (to the right)

Rule 2: we should apply a small negative force (to the left) (a positive force could result in the pendulum overshooting the desired position).

Rule 3: we should apply a small negative force (to the left) to assist the movement, but not a big one since the pendulum is already moving in the proper direction

Each of the three rules listed above is a "linguistic rule" since it is formed solely from linguistic variables and values.

The general form of the linguistic rules listed above is

If premise **Then** consequent

As we can see from the three rules listed above, the premises (which are sometimes called "antecedents") are associated with the fuzzy controller inputs and are on the left-hand-side of the rules. The consequents (sometimes called "actions") are associated with the fuzzy controller outputs and are on the right-hand-side of the rules.

5.4.1 Rule-Bases

Using the above approach, we could continue to write down rules for the pendulum problem for all possible cases. For above mentioned pendulum problem, with two inputs and five linguistic values there are at most $5^2 = 25$ possible rules (all possible combinations of premise linguistic values for two inputs). A tabular representation of one possible set of rules for the inverted pendulum is shown in the following Table 5.2:

Table 5.2: Possible set of rules

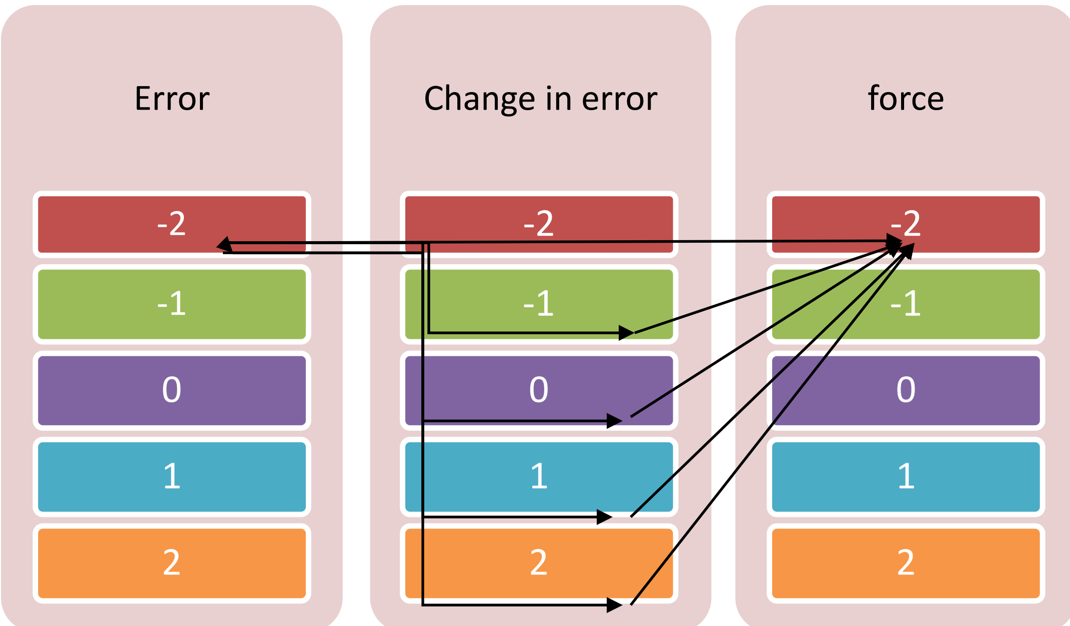
'Force ' u		Change in error e				
		-2	-1	0	1	2
Error e	-2	2	2	2	1	0
	-1	2	2	1	0	-1
	0	2	1	0	-1	-2
	1	1	0	-1	-2	-2
	2	0	-1	-2	-2	-2

Notice that the body of the table lists the linguistic-numeric consequents of the rules, and the left column and top row of the table contain the linguistic-numeric premise terms. For example,

If error is PL (2) **and** change-in-error is NS (-1) **Then** force is NS (-1) which is rule 3 above.

Table 5.2 represents abstract knowledge that the expert has about how to control the pendulum given the error and its derivative as inputs. The pattern of rule consequents that appears in the body of the table:

The diagonal of zeros and viewing the body of the table as a matrix we see that it has a certain symmetry to it. This symmetry that emerges when the rules are tabulated is no accident and is actually a representation of abstract knowledge about how to control the pendulum; it arises due to symmetry in the system's dynamics. We will actually see later that similar patterns will be found when constructing rule-bases for more challenging applications, and we will show how to exploit this symmetry in implementing fuzzy controllers. The each entry output force = -2 (NL) in above table 5.2 is obtained in the following manner. All the remaining entry will be obtain in similar manner.



5.5 FUZZY QUANTIFICATION OF KNOWLEDGE

Up to this point we have only quantified, in an abstract way, the knowledge that the human expert has about how to control the plant. Next, we will show how to use fuzzy logic to fully quantify the meaning of linguistic descriptions so that we may automate, in the fuzzy controller, the control rules specified by the expert.

5.5.1 Membership Functions

First, we quantify the meaning of the linguistic values using "membership functions." For example, this is a plot of a function μ versus e (t) that takes on special meaning as shown in figure 5.23. The function μ quantifies the certainty 2 that e (t) can be classified linguistically as "PS." To understand the way that a membership function works, it is best to perform a case analysis where we show how to interpret it for various values of e (t):

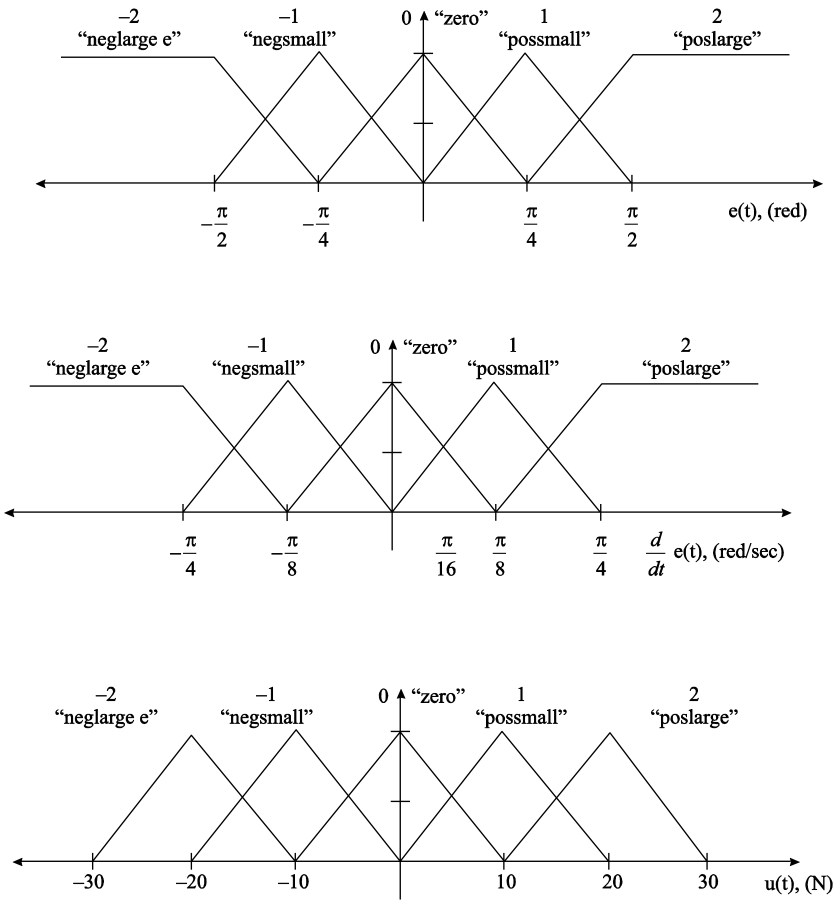


Fig. 5.23 Membership function for Inverted pendulum

5.5.2 Matching: Determination of Rules to be Used

It is actually the case that for most fuzzy controllers the fuzzification will explain the exact operations of the fuzzification process and also explain why it can be simplified and under certain conditions virtually ignored. The fuzzification process as the act of obtaining a value of an input variable (e.g., e (t)) and finding the numeric values of the membership function(s) that are defined for that variable. For example, if e (t) = π/4 and d/ dt e (t) = π/16, the fuzzification process amounts to finding the values of the input membership functions for these.

Next, we seek to explain how the inference mechanism in Figure 5.23 operates. The inference process generally involves two steps:

1. The premises of all the rules are compared to the controller inputs to determine which rules apply to the current situation. This "matching" process involves determining the certainty that each rule applies and we will more strongly take into account the recommendations of rules about which we are more certain to apply to the current situation.
2. The conclusions (what control actions to take) are determined using the rules that have been determined to apply at the current time. The conclusions are characterized with a fuzzy set (or sets) that represent the certainty that the input to the plant should take on various values.

5.5.3 Premise Quantification via Fuzzy Logic

To perform inference we must first quantify each of the rules with fuzzy logic. To do this we first quantify the meaning of the premises of the rules that are composed of several terms, each of which involves a fuzzy controller input. Consider Figure 5.24 in where we list two terms from the premise of the rule

If error is ZE **and** change-in-error is PS, **then** force is NS. Thus, we had quantified the meaning of the linguistic terms "error is ZE" and "change-in-error is PS" via the membership functions shown in Figure 5.24. Now we seek to quantify the linguistic premise "error is ZE **and** change-in-error is PS." Hence, the main item to focus on is how to quantify the logical "and" operation that combines the meaning of two linguistic terms. While we could use standard Boolean logic to combine these linguistic terms, since we have quantified them more precisely with fuzzy sets (i.e., the membership functions), we can use these.

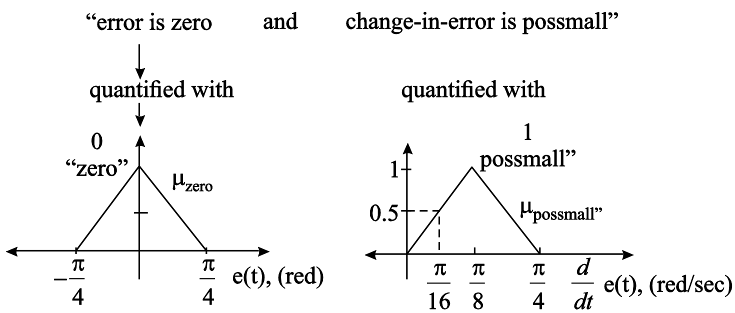


Fig. 5.24

5.5.4 Determining Which Rules Are On

Determining the applicability of each rule is called "matching." We say that a rule is "on at time t" if its premise membership function $\boldsymbol{\mu}_{\text{premise}}$ (e (t), d/dt e(t)) > 0.

Hence, the inference mechanism seeks to determine which rules are on to find out which rules are relevant to the current situation. In the next step, the inference mechanism will seek to combine the recommendations of all the rules to come up with a single conclusion. Consider, for the inverted pendulum example, how we compute the rules that are on. Suppose that

$$e(t) = 0$$

and

$$d/dt\ e(t) = \pi/8 - \pi/32\ (= 0.294)$$

Figure 5.25 shows the membership functions for the inputs and the values above for e(t) and d/ dt e(t) are indicated with thick black vertical lines

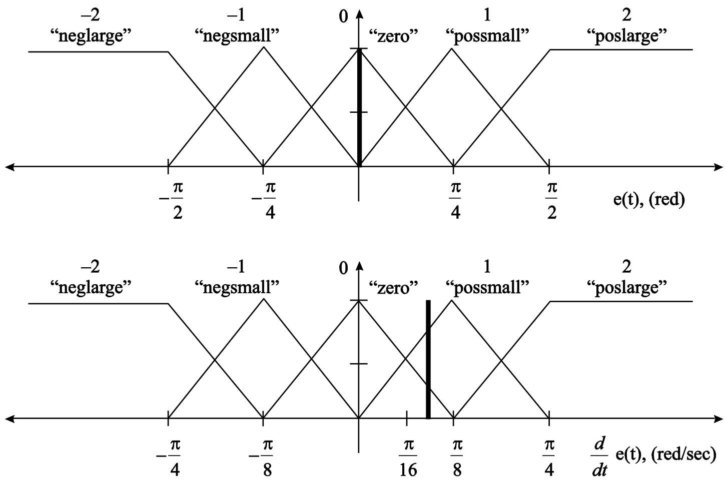


Fig. 5.25

5.5.5 Inference Step: Determining Conclusions

Next, we consider how to determine which conclusions should be reached when the rules that are on are applied to deciding what the force input to the cart carrying the inverted pendulum should be. To do this, we will first consider the recommendations of each rule independently. Then, we will combine all the recommendations from all the rules to determine the force input to the cart.

We see that $\mu_{(1)}(u)$ is in general a time-varying function that quantifies how certain rule (1) ,i.e., the force input u should take on certain values. It is most certain that the force input should lie in a region around zero (see Figure 5.26 (b)), and it indicates that it is certain that the force input should not be too large in either the positive or negative direction—this makes sense if we consider the linguistic meaning of the rule. The membership function $\mu_{(1)}(u)$ quantifies the conclusion reached by only rule (1) and only for the current e(t) and d/dt e(t). It is important that the reader be able to picture how the shape of the implied fuzzy set changes as the rule's premise certainty changes over time.

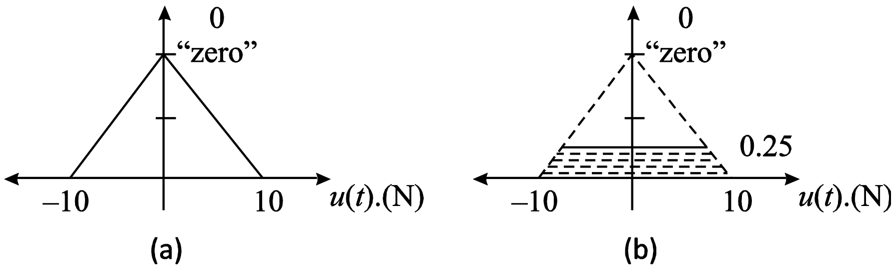


Fig. 5.26

5.5.6 Converting Decisions into Actions

Next, we consider the defuzzification operation, which is the final component of the fuzzy controller shown in Figure 5.18. Defuzzification operates on the implied fuzzy sets produced by the inference mechanism and combines their effects to provide the "most certain" controller output (plant input). Some think of defuzzification as "decoding" the fuzzy set information produced by the inference process (i.e., the implied fuzzy sets) into numeric fuzzy controller outputs.

To understand defuzzification, it is best to first draw all the implied fuzzy sets on one axis as shown in Figure 5.27. We want to find the one output, which we denote by "u^{crisp}," that best represents the conclusions of the fuzzy controller that are represented with the implied fuzzy sets. There are actually many approaches to defuzzification.

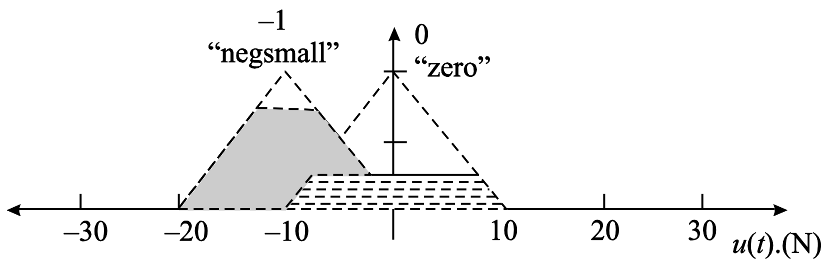


Fig. 5.27

5.5.7 Graphical Depiction of Fuzzy Decision Making

For convenience, we summarize the procedure that the fuzzy controller uses to compute its outputs given its inputs in Figure 5.28. Here, we use the minimum operator to represent the "and" in the premise and the implication and COG defuzzification. The reader is advised to study each step in this diagram to gain a fuller understanding of the operation of the fuzzy controller. To do this, develop a similar diagram for the case where the product operator is used to represent the "and" in the premise and the implication, and choose values of e (t) and d/dt e (t) that will result in four rules being on. Then, repeat the process when center-average defuzzification is used with either minimum or product used for the premise. Also, learn how to picture in your mind how the parameters of this graphical representation of the fuzzy controller operations change as the fuzzy controller inputs change.

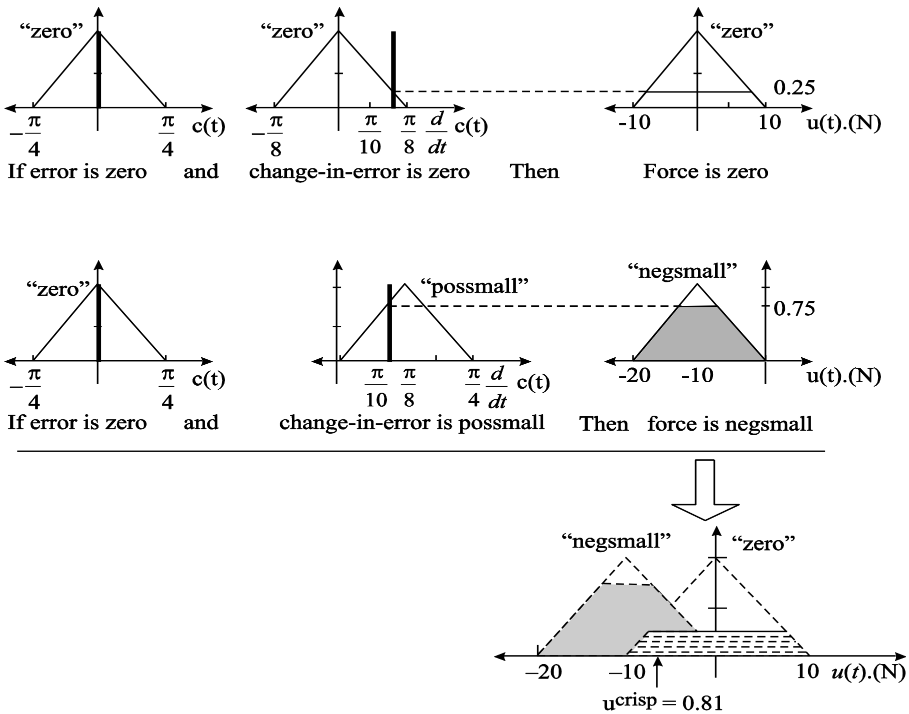


Fig. 5.28

This completes the description of the operation of a simple fuzzy controller. You will find that while we will treat the fully general fuzzy controller in the next section, there will be little that is conceptually different from this simple example. We simply show how to handle the case where there are more inputs and outputs and show a fuller range of choices that you can make for the various components of the fuzzy controller.

As evidenced by the different values obtained by using the minimum, product, and defuzzification operations, there are many ways to choose the parameters of the fuzzy controller that make sense. This presents a problem since it is almost always difficult to know how to first design a fuzzy controller. Basically, the choice of all the components for the fuzzy controller is somewhat ad hoc. What are the best membership functions? How many linguistic values and rules should there be? Should the minimum or product be used to represent the "and" in the premise—and which should be used to represent the implication? What defuzzification method should be chosen? These are all questions that must be addressed if we want to design a fuzzy controller.

5.6. FUSION OF FUZZY AND PID

A typical example is the conventional proportional-integral-derivative (PID) controller design, where the first- or second-order linear plant transfer function has to be first given. We will review and discuss these conventional PID controllers in Chapter 6. Here, to design a fuzzy logic controller, not necessarily of PID-type, for set point tracking, we suppose that the mathematical formulation of the plant is completely unknown.

Most of fuzzy control research focuses on set point regulation problem, hence, fuzzy control is often viewed as a form of nonlinear PID control and comparisons of fuzzy control vs. fuzzy control are frequent in literature.

Conventional PID control is, however, well established and can satisfy the performance requirements of most set point regulation problems at minimal cost.

Often, performance improvement offered by fuzzy control cannot compensate the increased complexity in computation and tuning. Consequently, fuzzy PID control is rare in commercial applications; commercial applications of fuzzy control are largely focused on high-level, task-oriented control that fall outside the domain of conventional control methods Chiu [8].

i) Supervision of conventional controllers

In this configuration, the high level strategy is used for adjustments of the parameters of the conventional control loops. A common problem with linear PID controllers used for control of highly nonlinear processes is that the set of controller parameters produces satisfactory performance only when the process is within a small operational window. Outside this window, different PID controller parameters are necessary, and these adjustments may be done automatically by a higher level controller (Fig. 5.29).

The control system consists of a conventional discrete PID controller of which the proportional, integral and derivative gains (K_p, K_i, K_d) are changed by a fuzzy supervisor each sampling time (thus fuzzy controller is regarded a gain scheduler). The inputs of the supervisor might be error and the change of error Nauta Lemke and Krijgsman [9] Zhao et. al.[10] the supervisor is used to tune the scaling factors of the underlying fuzzy PD controller depending on overshoot, reaching time and amplitude of oscillation.

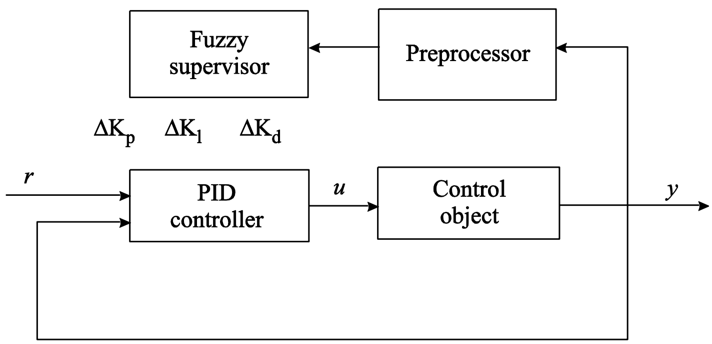


Fig. 5.29 Supervisory control system for advanced PID control

Alternative gain scheduling scheme may be implemented through the use of a homogeneous 1st order TS system with the following rule format Jang and Gulley [11]:

IF e is A_{1r} AND Δe is A_{2r} THEN $y_r = p_{1r}\, e + p_{2r}\Delta e$

Each such rule represents a local PD controller (with suitable variable selection other control laws can be implemented) that are combined together by means of fuzzy logic. Thus it is possible to specify separate control law for each region of variable space that is determined by the selection of input MFs. Low transparency measure of the controller would be useful.

ii) Correction of conventional controllers

Normally, conventional control systems, which are based on PID controllers, are capable of controlling the process when the operation is smooth and close to normal conditions. However, if sudden changes occur or if the process enters abnormal situations, then a configuration that is capable of bringing the process back to normal operation as fast as possible may be useful. This idea can be implemented using the parallel connection of fuzzy and PID controllers (Fig. 5.30). For normal operation, the addition to the output of PID controller is zero, whereas in abnormal situations fuzzy controller develops nonzero output that restores the normal state.

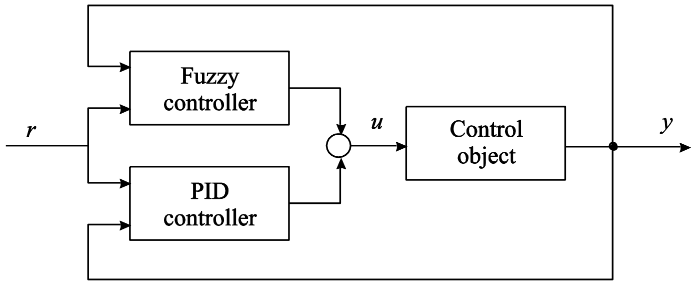


Fig. 5.30 Parallel connection of fuzzy and PID control

iii) Coordination of control loop set-points

Fuzzy logic can be applied not only for calculation of control variable (i.e. direct control) but for modification of control strategy in general, by using a fuzzy supervisor that makes adjustments of the controller set point (Fig 5.31).

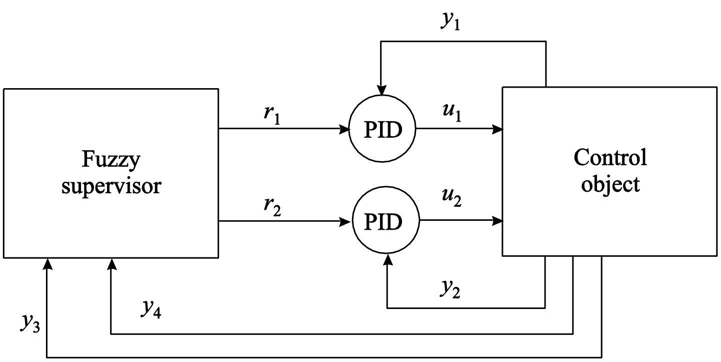


Fig. 5.31 Supervisory control system for modifying the control strategy

History of such fuzzy hierarchical control configuration goes back to the first industrial application of fuzzy control where human supervisor of the cement kiln was replaced. Such utilization of fuzzy logic can be very effective because high-level control strategy is often formulated using natural language and the translation into the language of fuzzy systems may be rather straightforward. Setpoint control, on the other hand, is often more effective if implemented with conventional tools for reasons pointed out earlier. The applications described in chapter 6 use this type of hierarchical control architecture exclusively.

5.7 INVERSION OF FUZZY SYSTEMS

The design process for fuzzy controllers that makes use of heuristic information originating from human experts has many successful applications. Supervised learning of fuzzy controllers has also found use. Both approaches have, however, several shortcomings, as already described in the introduction. It may be difficult to perform the initial synthesis of the controller or to maintain required performance as some time passes (because of significant and unpredictable drift of plant parameters or the presence of noise or some other type of disturbance).

The solution is to use self-learning fuzzy controllers that can adapt to different plant conditions, i.e. adaptive fuzzy control.

On the other hand, it can be postulated that the ultimate goal of the controller design is to derive the inverse model of the process. In theory, the use of an inverse model possesses the advantages of open-loop control, i.e. inherent stability and perfect control with zero error. The major concern is that if the inverse configuration actually exists or if it is physically realizable. Global inversion of the system, where all states become outputs of the inverted model and the output of the original system becomes the state variable (Fig. 5.32, center) has normally non-unique solution and must be given by a family of solutions. In case of partial inversion, only one of the states of the original system becomes an output of the inverted model and other states together with the original output are the inputs of the inverted model (Fig. 5.32, right).

Partially inverted model can be also more easily embedded into the control system than the global inversion.

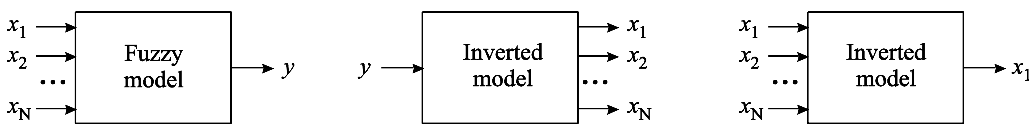


Fig. 5.32 Global (center) and Partial (right) inversion of the original fuzzy model(left)

The partial inversion has often a unique solution but not necessarily (the original model must be strictly monotone in respect to the inverted state to be invertible).

5.7.1 Numerical inversion of fuzzy systems

The most intuitive way to obtain the inversion is to reverse the input and output data and train an inverse model of a system or process as shown in (Fig. 5.33, left). For the sake of simplicity only two states are brought in with $x_1 = u\ (k)$, $x_2 = y\ (k)$, $y = y\ (k + 1)$. This type of training has been long practiced with neural networks. Two major drawbacks are characteristic to this approach - first, if several values of u are possible for the same output of the process (many-to-one mapping, Fig.5.33. right), and a least-squares approach is used, the identification algorithm maps y to the mean of all u, which can lead to a quite meaningless inverse model.

Secondly, it is difficult to pick the appropriate signal u for inverse learning, as it is supposed to work over a wide range of amplitudes of y and for a large bandwidth.

The second problem can be solved with online learning scheme depicted in Fig. 5.34., successful application of which is reported in Jang (1995). Here, the controller is the copy of inverse model and instead process output u we need to pick r that is the set point for the controlled process that we usually know.

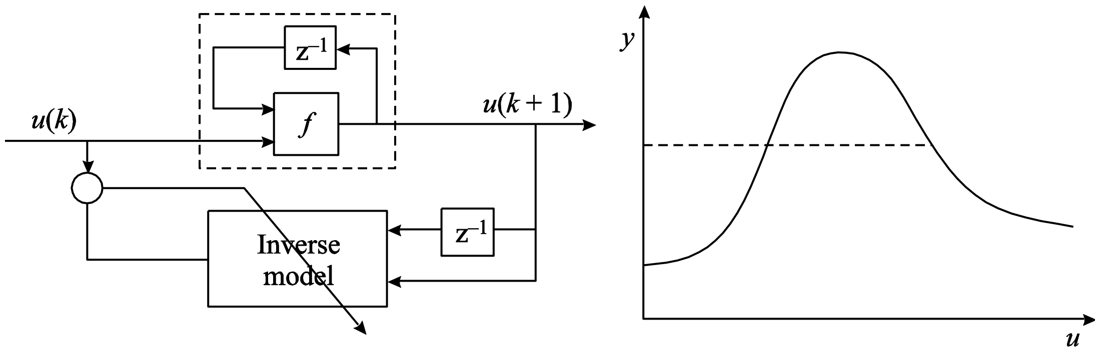


Fig. 5.33 Training of the inverse process model (left) non-invertible function (right)

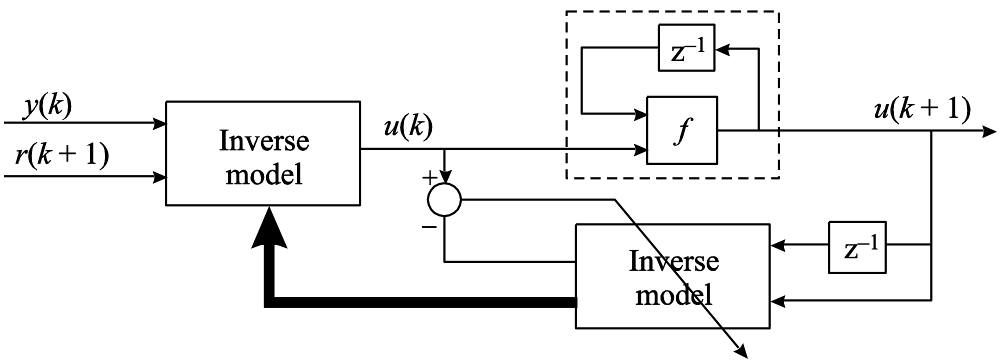


Fig. 5.34 Online inverse training

Another numerical inversion technique that is generally associated with neural networks is back propagation through time (temporal back propagation) that was invented by several researchers simultaneously in 1980s. One of the most famous applications of this technique is the truck backer-upper simulation.

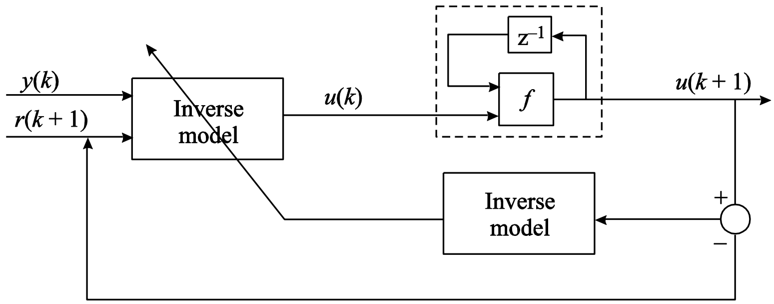


Fig. 5.35 Back propagation through time

The goal of training is to minimize the cost function (5.2) by changing the parameter set θof the controller

$$K[\theta(k)] = \frac{1}{2}(r(k) - y(k))^2 . \tag{5.2}$$

We apply gradient descent method using the chain rule:

$$\frac{\partial J[\theta(k)]}{\partial \theta(k)} = \frac{\partial J[\theta(k)]}{\partial y(k)} \frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \theta(k)} \tag{5.3}$$

Note that the error is back propagated through the model (though the parameters of the model are not updated by gradient descent). The model of the system can be a neural emulator, neuro-fuzzy system or even a set of mathematical equations as demonstrated in Jang [12], where the method implemented with ANFIS is used to balance the inverted pendulum.

5.7.2 Non-numerical inversion of fuzzy systems

The numerical identification of the inverse model may become computationally expensive, requiring many training epochs and many training samples. The issue of invertibility is also of importance and not very well handled with automatic generation of the inverted model. The techniques for training the inverted fuzzy model have become known basically through neural network research. Fuzzy systems, however, are principally different from neural networks because: (a) they can be interpreted in linguistic terms; (b) if transparent, their parameters can be interpreted in terms of their influence to the input-output relationship. First feature allows linguistic inversion Second feature allows exact analytical inversion.

Linguistic inversion (causality inversion) is obtained through the exchange of antecedent and consequent variables in fuzzy rules if a symmetrical operator (t norm) represents the if-then relation.

Consider a three-input fuzzy system from Baranyi et. al.(1998) as given given in the following Table 5.3:

Table 5.3 Rule base of the original model

U_2, U_3	**U_1 is small**	**U_1 is medium**	**U_1 is large**
U_2 is low AND U_3 is low	zero	low	medium
U_2 is low AND U_3 is high	low	medium	high
U_2 is high AND U_3 is low	low	medium	high
U_2 is high AND U_3 is high	medium	medium	high

The inversion procedure may have three possible results marked in table 5.4:

i) The input configuration is unique. This is the ideal case.

ii) The input configuration is non-unique that means that the rule base is non-invertible. The approximate solution is to choose the input configuration with the lowest control energy (in linguistic sense).

iii) There are no inputs that allow one-step transition to the desired output.

The reason why such situation occurs is two-fold. First, the number of MFs given for U_1 and V is not equal, i.e. there are 16 rules in the inverted model whereas the number of rules in the original model is 12.

The second reason is that the original system may not simply allow one-step transition to the desired output from the given state. The approximate solution is to choose the "nearest" output (again in linguistic sense).

Table 5.4 Inverted rule base

U_2, U_3	**V is zero**	**V is low**	**V is medium**	**V is high**
U_2 is low AND U_3 is low	small $^{(i)}$	medium $^{(i)}$	large $^{(i)}$	large $^{(iii)}$
U_2 is low AND U_3 is high	small $^{(iii)}$	small $^{(i)}$	medium $^{(i)}$	large $^{(i)}$
U_2 is high AND U_3 is low	small $^{(iii)}$	small $^{(i)}$	medium $^{(i)}$	large $^{(i)}$
U_2 is high AND U_3 is high	small $^{(iii)}$	small $^{(iii)}$	small $^{(ii)}$	large $^{(i)}$

One must pay careful attention to MFs of the inverted model. Input and output MFs may be of different type (e.g. triangular input and singleton output MFs) and therefore conversion of MFs is in order. The conversion algorithms are rather straightforward. This is similarly valid even if linguistic inversion is performed on standard fuzzy systems with triangular MFs (unless they are uniformly distributed on all variables) because of the contradictive nature of input and output transparency constraints.

If the inversion goal is fixed (let us for instance assume that the desired state of y is under label of medium), it is possible to perform local inversion of the model. The difference between partial linguistic inversion and local linguistic inversion is that while inverted input

(U_1) becomes output, inverted output (V) is never used in the final inverted model. From the rules that have linguistically identical premises in terms of U_2 and U_3 (i.e. rows in table 5.3) only one that Transparent fuzzy systems: modeling and control 120 results in the desired V is selected. Thus, indirect inversion of the model inTable 5.3 would appear as

IF U_2 is low AND U_3 is low THEN U_1 is large

IF U_2 is low AND U_3 is high THEN U_1 is medium

IF U_2 is high AND U_3 is low THEN U_1 is medium

IF U_2 is high AND U_3 is high THEN U_1 is small$^{(ii)}$

There are the following advantages of this approach:

i) The number of rules of the inverted model is S_y times smaller than in the case of linguistic inversion, where S_y is the number of linguistic labels of y.

ii) Type of consequent MFs does not matter, as they are never inverted. For instance, models with incremental rule bases would be extremely difficult to invert linguistically, because of the large number of consequent MFs. With indirect inversion this is not a problem, rather the opposite.

iii) Less problems with $^{(ii)}$ and $^{(iii)}$ type rules.

The basic disadvantage of local inversion is that the inverted model is valid only for the fixed inversion goal and in case of multiple goals model must be re-inverted each time.

5.7.3 Example of Control

Hereby we present a comparison of selected control methods described above. The controlled object depicted in Fig. 5.36 is the water tank with a pipe flowing in and pipe flowing out (MATLAB demo). The outflow rate F_o depends on the diameter of the outflow pipe (which is constant) and the pressure in the tank (the latter varies with the water level $h \in [0, 2]$). Thus the system has some very nonlinear characteristics. Our task is to maintain some predetermined water level (we choose the set-point that changes between 0.5 and 1.5) in the tank by changing the valve that lets the water to flow in with Inflow rate $F \in [-1, 1]$. The generic control system that suits for most of the designed controllers is depicted in Fig. 5.37.

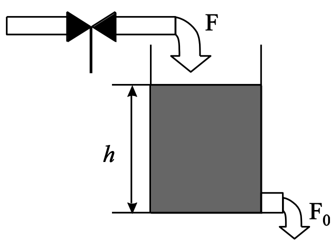


Fig. 5.36 Control object

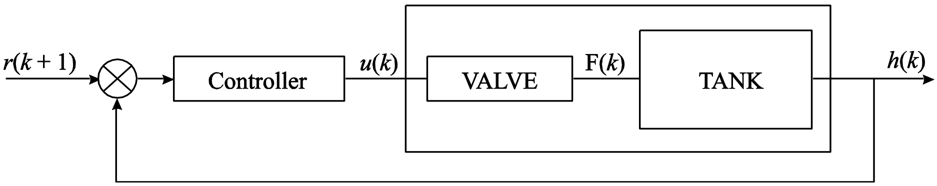


Fig. 5.37 Control system

Conventional control

There is not much to comment on conventional setpoint control. The nearoptimal tuning parameters are not difficult to find. In case of P control K_p = 100; in case of PD control K_p = 7 and K_d = 10; application of PI or PID control does not make much sense as there is no performance improvement with these control laws. The problem with P control is that it does not remove oscillations (Fig. 5.38). The problem with PD control is that while it results in superb performance in noise-free environment, it is very sensitive to output-additive white noise even if the level of noise is not very high (presently 1%) as can be seen from Fig. 5.39. PD control would supposedly benefit from internal model control scheme (Dutton et. al. 1996) that can cancel the influence of output additive noise.

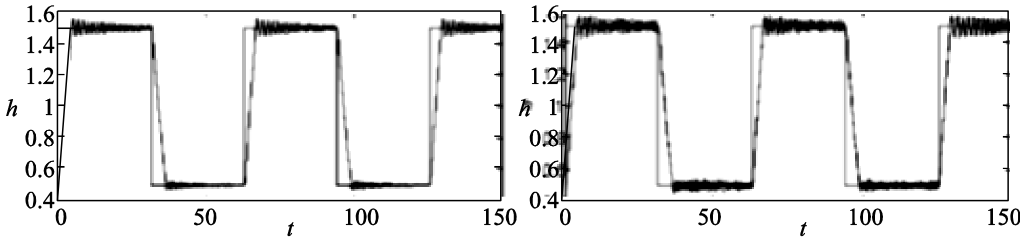


Fig. 5.38 P control Noise-free (left), with noisy data (right)

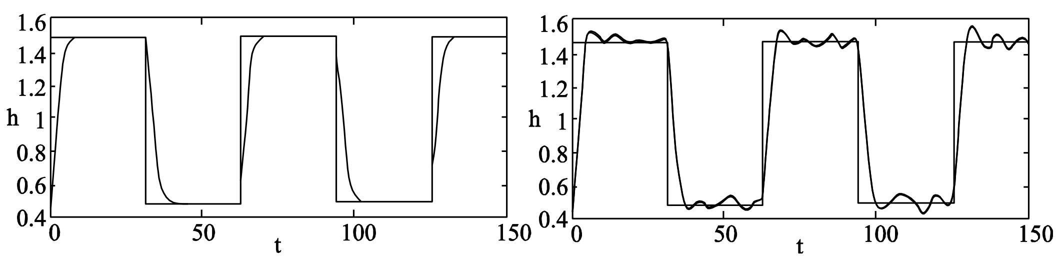


Fig. 5.39 PD control Noise-free (left), with noisy data (right)

MATLAB fuzzy controller

The controller that is developed for MATLAB demo is quite interesting. It takes e (k) and dh(k)/dt as its inputs, whereas the rule base of the controller is nontraditional and consists of only the following 5 rules:

IF e (k) is Z	THEN u (k) is Z
IF e (k) is P	THEN u (k) is PB
IF e (k) is N	THEN u (k) is NB
IF e (k) is Z AND dh (k) is P	THEN u (k) is NS
IF e (k) is Z AND dh (k) is P	THEN u (k) is PS

First three rules are responsible for general control strategy (relay-type logic), whereas last two ensure smooth action around the set point. It is interesting to note that the controller is non-transparent even though the MFs do not violate any of transparency conditions. The first rule is always fired whenever e (k) falls into Z, thus controller output cannot achieve NS and PS values even if the respective rules are fully activated but falls somewhere in between Z and those MFs. Nevertheless, the control performance does not necessarily suffer from non-transparency as has been shown many times. The controller is also surprisingly insensitive to noise (Fig. 5.40).

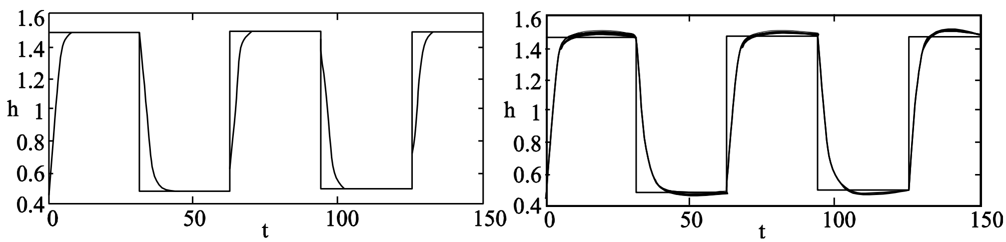


Fig. 5.40 Fuzzy control (MATLAB demo) Noise-free (left), with noisy data (right)

Fuzzy PD controller

Fuzzy PD controller is constructed according the guidelines of section 5.2 with 5 fuzzy sets on each input variable, thus consisting of 25 fuzzy rules. Input MFs have uneven partition as in Fig. 5.41, left and in the final tuning phase, scaling factors $k_e = 0.8$, $k_{de} = 5$ and $k_u = 1.5$, are specified. This way, rather good control performance can be obtained (Fig. 5.41), whereas fuzzy PD control is less sensitive to noise, as can be seen from Fig. 5.41, right.

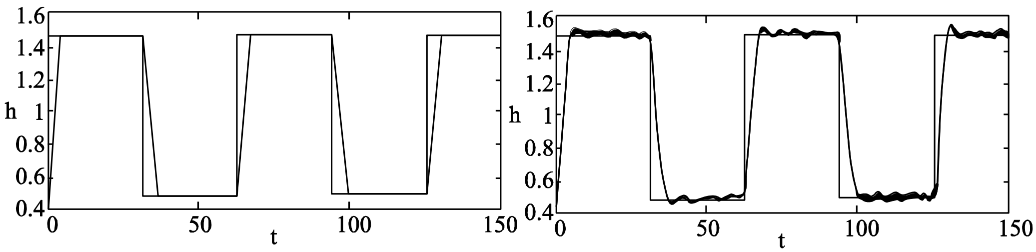


Fig. 5.41 Fuzzy PD control. Noise-free (left), with noisy data (right)

Control using analytical inversion

In this approach fuzzy (transparent) model of the controlled process must be identified first. We collect 1000 samples of system response (Fig. 5.42, right) generated with input noise (Fig. 5.42, left).

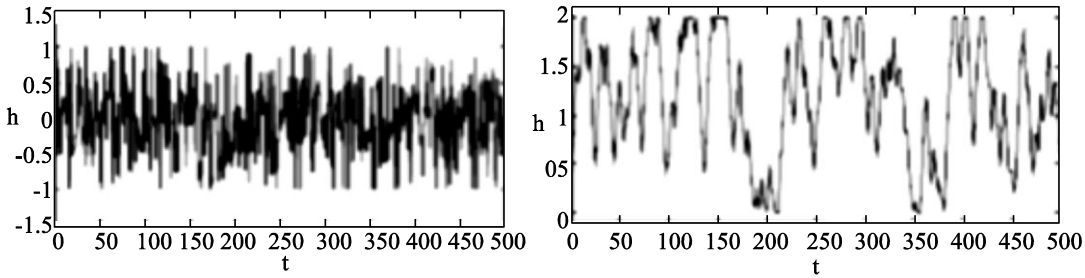


Fig. 5.42 Collected data Input u (left), output h (right) Sampling interval is 0.5 s.

Based on this collection, modeling data set is constructed to the format

$$h\,(k + 1) = f(h(k), h(k - 1), u(k)) \qquad (5.4)$$

The problem with this method is that it is not always simple to choose the appropriate u algorithmically if the rule base of the model is non-invertible. We choose 2 MFs for each input variable. This way, non-invertibility is avoided and the modeling error also seems low enough (RMSE = 0.0261).

More serious problem with (5.4) is, however, that the inverted controller u (k) = f(h(k), h(k - 1), r(k + 1)) pays no attention to system inertia. It always chooses positive u whenever error is positive (and vice versa). It never slows down near the set point and thus performs as (poorly tuned) P controller and is shown in the following Fig. 5.43:

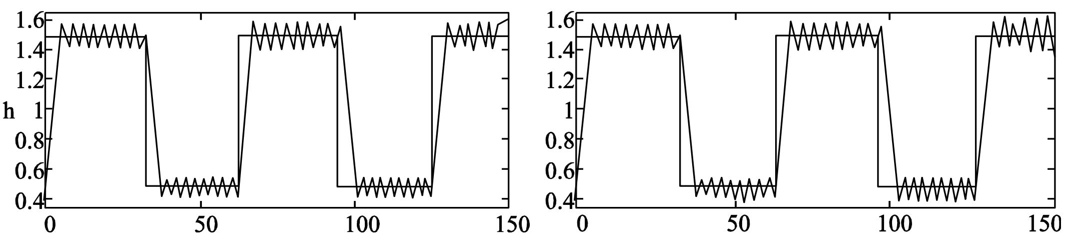


Fig. 5.43 Inverted model based control. Noise-free (left), with noisy data (right)

To give the controller some predictive power (derivative gain, so to speak) we use the model structure.

$$h\,(k + 2) = f(h(k + 1), h(k), u(k)), \qquad (5.5)$$

in inverted mode

$$u\,(k) = f\,(h\,(k), h(k + 1), r(k + 2)), \qquad (5.6)$$

Where unknown h (k + 1) is predicted recursively using the fuzzy model (5.27).

$$h\,(k + 1) = f(h(k), h(k - 1), u(k - 1)). \qquad (5.7)$$

The resulting controller has much better performance (Fig. 5.44). Presumably, with a more complex (and accurate) model, even better results could be obtained.

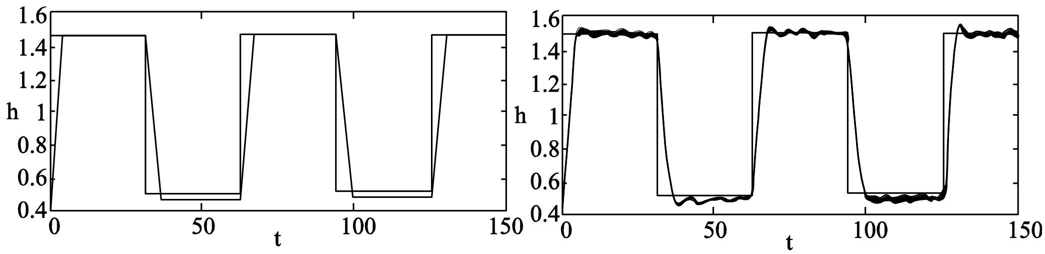


Fig. 5.44 Inverted model based control. Ahead inverted model Noise-free (left), with output-additive noise (right).

Inverted control is not very sensitive to noise and, in fact, in this case, internal model control (Fig. 5.45) is directly applicable as fuzzy model of the process has already been identified.

Control using linguistic inversion

With model configuration h(k + 1) = f(h(k), u(k)), output is highly correlated to h(k) and while this way low average approximation error can be obtained, it is very likely that important information about u is not present and cannot be extracted. One possibility to cope with this problem is to use fuzzy variational model. More serious problem with linguistic inversion (as already pointed out in section 5.5.1) is a large amount of type [(ii)] and [(iii)] rules, moreover, because of model-plant mismatch there are also contradictory rules that in summary means the majority of the rules must be derived from our knowledge about the controlled process and that too much depends on critical judgment – which, on the other hand, is a classical example of knowledge-based control (already implemented with fuzzy PD controller). This is also the reason why there are very few successful reports of linguistic inversion based control and allows us to conclude that control based on linguistic inversion is not very well suited for set point control.

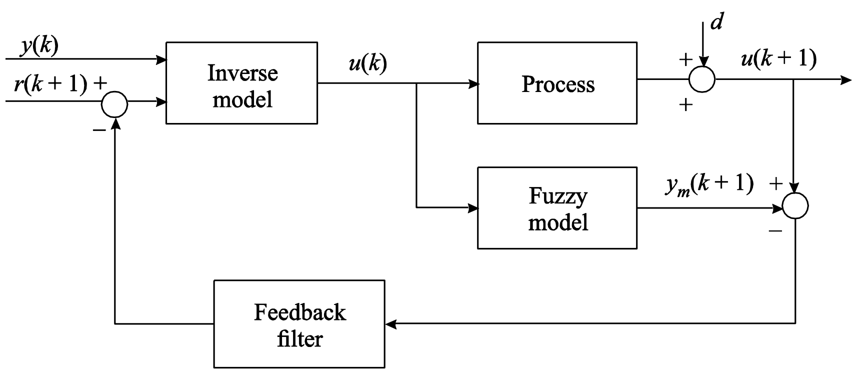


Fig. 5.45 Internal model control scheme

The compared controllers, respective control laws and performance measures IAE (integrated average error) are given in Table 5.5. Note: IAE is not the best possible performance criterion as it favors fast rise times and oscillating (e.g. P control) to slower but much smoother response (e.g. PD control) but it produces the numbers that are somewhat representative.

Table 5.5 Control results.

Controller	Control law	IAE	IAE (noisy data)
P	u = f(e)	0.0818	0.0787
PD	u = f(e, de/dt)	0.0940	0.1250
MATLAB Fuzzy	u = f(e, dh/dt)	0.0994	0.0859
Fuzzy PD	u = f(e(k), Δe(k))	0.0828	0.0899
Inverted model 1	u(k) = f(h(k – 1), h(k), r(k + 1))	0.1429	0.1346
Inverted model 2	u(k) = f(h(k), h(k + 1), r(k + 2)	0.1163	0.1129

Fuzzy approaches seem to be less sensitive to noise but that does not necessarily mean that fuzzy control is more robust per se. This sensitivity rather depends on how the inputs of the controller are processed. I.e. controllers that have continuous derivative component (as PD) experience most serious performance drop, whereas controllers where the derivative is obtained by discrete approximation (Fuzzy PD) or there is no derivative component at all (P, Inverted model based control) are much more robust. The level of noise depends on application and very often conventional control is more reliable and almost always less costly than fuzzy control.

5.8 SUMMARY AND CONCLUSIONS

During the last few decades, fuzzy control has been successfully applied to many control problems its applications involve a wide range of products ranging from simple consumer products to control systems of complex technological systems.

The main strengths of fuzzy logic control have arguably been:

1. It can be used in systems which cannot be easily modeled mathematically. In particular, systems with non-linear responses that are difficult to analyze may respond to a fuzzy control approach.
2. As a rule-based approach to control, fuzzy control can be used to efficiently represent an expert's knowledge about a problem.
3. Continuous variables may be represented by linguistic constructs that are easier to understand, making the controller easier to implement and modify.
4. Fuzzy controllers may be less susceptible to system noise and parameter changes, thus making them more robust.
5. Complex processes can often be controlled by relatively few logic rules, allowing a more understandable controller design and faster computation for real-time applications.

A fuzzy controller is constructed to make decisions about what the control input to the plant should be given processed versions of the plant outputs and reference input. It is a form of artificial (i.e., non-biological) decision-making system. Decision making systems find wide applications in many areas in addition to those have been traditionally studied in control systems. For instance, the machine scheduling case study of the previous section shows a nontraditional application of feedback control where a fuzzy system can play a useful role as a decision-making system.

There are many other areas in which fuzzy decision-making systems can be used including the following:

- Manufacturing: Scheduling and planning materials flow, resource allocation, routing, and machine and equipment design.
- Traffic systems: Routing and signal switching.
- Robotics: Path planning, task scheduling, navigation, and mission planning.
- Computers: Memory allocation, task scheduling, and hardware design.
- Process industries: Monitoring, performance assessment, and failure diagnosis.
- Science and medicine: Medical diagnostic systems, health monitoring, and automated interpretation of experimental data.
- Business: Finance, credit evaluation, and stock market analysis.

This list is by no means exhaustive. Virtually any computer decision-making system has the potential to benefit from the application of fuzzy logic to provide for “soft” decisions when there is the need for decision making under uncertainty. At last we focus on the design of fuzzy decision-making systems for problems other than feedback control. We begin by showing how to construct fuzzy systems that provide warnings for the spread of an infectious disease. Then we show how to construct a fuzzy decision making system that will act as a failure warning system in an aircraft.

Its main weaknesses include

1. Lack of systematic approach in knowledge acquisition and tuning of the controller
2. Lack of reliable means for the analysis of optimality and stability of the controller.
3. Combinatorial explosion of rules in case of multitude of inputs.

Most existing fuzzy logic controllers since Mamdani's application Mamdani and Assilian [7] are set point controllers that respond to proportional and derivative (PD) or proportional and integral (PI) terms (section 5.6).

EXERCISE 5 (PROBLEM DESIGNING)

Design1: Study of a biological system where a fuzzy decision-making system is used as a warning system to produce alarm information.

To model a form of biological growth, using one of Volterra's population equations a simple model representing the spread of a disease in a given population is given by

$$\frac{dx1(t)}{dt} = -ax_1(t) + bx_1(t)x2(t) \quad (5.8)$$

$$\frac{dx2(t)}{dt} = -bx_1(t)x_2(t) \quad (5.9)$$

where $x_1(t)$ is the density of the infected individuals, $x_2(t)$ is the density of the noninfected individuals, $a > 0$, and $b > 0$. These equations are only valid for $x_1(t) \geq 0$ and $x_2(t) \geq 0$. The initial conditions $x_1(0) \geq 0$ and $x_2(0) \geq 0$ must also be specified. Equation (5.9) intuitively means that the noninfected individuals become infected at a rate proportional to $x_1(t)x_2(t)$. This term is a measure of the interaction between the two groups. The term $-ax_1(t)$ in Equation (5.8) represents the rate at which individuals die from disease or survive and become forever immune. The term $bx_1(t)x_2(t)$ in Equation (5.1) represents the rate at which previously noninfected individuals become infected.

Problem 5.1: Design a fuzzy system to produce alarms if certain conditions occur in the diseased population—that is, a simple warning system. The fuzzy system uses $x_1(t)$ and $x_2(t)$ as inputs, and its output is an indication of what type of warning condition occurred along with the certainty that this warning condition has occurred.

Design 2 Failure Warning System for an Aircraft

Consumer and governmental demands have provided the impetus for an extraordinary increase in the complexity of the systems that we use. For instance, in automotive and aircraft systems, governmental demands have called for

(1) Highly accurate air-to-fuel-ratio control in automobiles to meet pollution standards,

(2) Highly technological aircraft capable of achieving frequent flights with very little maintenance downtime.

Similarly, consumer demands have driven

(1) The development of antiskid braking systems for increased stop ability, steer ability, and stability in driving and

(2) The need for increased frequency of commercial flights such that travel must occur under all weather conditions in a timely manner.

While engineers have, in general, been able to meet these demands by enhancing the functionality of high-technology systems, this has been done at the risk of significant failures (it is generally agreed that "the more complex a system is the more likely it is to fail in some way").

For automotive and aircraft systems, some of the failures that are of growing concern include the following:

- Failures and/or degradation of performance of the emissions control systems (failures or degradation leads to a significant increase in the level of pollutants).
- "Cannot duplicate" failures where a failure is detected while the aircraft is in flight that cannot be duplicated during maintenance, which lengthens the downtime.
- Actuator, sensor, and other failures in aircraft systems that cause commercial aircraft crashes in adverse weather conditions.
- A system failure in an integrated vehicle handling, braking and traction control system, which can lead to a loss of control by the driver.

Automotive and aircraft systems provide excellent examples of how failures in high-technology systems can result in catastrophic failures. In addition, the effect of undetected system faults can lead to costly downtime or catastrophic failures in manufacturing systems, nuclear power plants, and process control problems. As history indicates, the probability of some of the system failures listed above is sometimes high. There is then the need for detecting, identifying, and providing appropriate warnings about failures that occur on automobiles, aircraft, and other systems so that corrective actions can be taken before there is a loss of life or other undesirable consequences**.**

Problem 5.2: Suppose the aircraft's input vector u has two components, the elevator δe (deg), and thrust δt (deg).The output vector y has three components, pitch rate q (deg/sec), pitch angle θ (deg), and load factor ηz (g).

Consider four aircraft failure modes (To define the modes, take the same approach as in the previous section and define decision regions using conventional logic and inequalities. Later, soften the decision boundaries and define the fuzzy decision-making system.)

To define the decision boundaries, each input and output is discretized into five regions with four boundaries associated with the real number line. For example, the elevator δe is discretized as follows:

- Region R_1: $\delta e \leq \delta R_1$
- Region Y_1: $\delta R1 < \delta e \leq \delta Y_1$
- Region G: $\delta Y1 < \delta e \leq \delta Y_2$
- Region Y_2: $\delta Y_2 < \delta e \leq \delta R_2$
- Region R_2: $\delta e \geq \delta R_2$

The G (for Green) region denotes an area of safe operation, the Y_1 and Y_2 (for Yellow) regions denote areas of warning, and the R_1 and R_2 (for Red) regions denote areas of unsafe operation.

Using the defined regions for the aircraft inputs and outputs, explain four failure modes for the aircraft identified as follows:

1. Load factor is in region R_2.
2. Load factor is in region Y_2.
3. Load factor is in region Y_2 and elevator is in region Y_1
4. (Pitch rate is in Y_1 and Pitch angle is in Y_1) or (Pitch rate is in Y_2 and Pitch angle is in Y_2).

BIBLIOGRAPHY

1. C. R. Asfahl. Robots and Manufacturing Automation. New York, NY: Wiley ,1992.
2. D. Drianker, H. Hellendoorn, and M. Reinfrank. An Introduction to Fuzzy Control. Berlin, Germany: Springer-Verlag ,1993.
3. K. Tanaka. “Design of model-based fuzzy control using Lyapunov’s stability approach and its applications to trajectory stabilization of a model car.” Theoretical Aspects of Fuzzy Control (ed. H. T. Nguyen, M. Sugeno, R. Tong, and R. R. Yager), pp 31-63, New York, NY: Wiley ,1995.
4. L. X. Wang. Adaptive Fuzzy Systems and Control. Englewood Cliffs, NJ: Prentice-Hall (1994).
5. J. W. Webb and R. A. Reis. Programmable Logic Controllers: Principles and Applications. Englewood Cliffs, NJ: Prentice-Hall ,1995.
6. L. A. Zade (1994).. “The role of fuzzy logic in modeling, identification, and control.” Modeling, Identification, and Control, Vol. 15, No. 3, pp 191-203
7. W. J. M. Kickert and E. H. Mamdani (1978). Analysis of a fuzzy logic controller. Fuzzy Sets and Systems, 1:29–44.
8. S. Chiu, S. Chand, D. Moore, and A. Chaudhary (1991). Fuzzy logic for control of roll and moment for a flexible wing aircraft. IEEE Control Systems Magazine,11(4):42–48 .
9. H. R. van Nauta Lemke and D.-Z. Wang. Fuzzy PID supervisor. In Proc. of the IEEE Conf. on Decision and Control, pages 602–608, Fort Lauderdale, FL, December 1985.
10. J. Zhao, R. Gorez, and V. Wertz. Synthesis of fuzzy control systems based on linear Takagi-Sugeno fuzzy models. In R. Murray-Smith and T. A. Johansen, editors, Multiple Model Approaches to Nonlinear Modeling and Control, pages 311–340. Taylor and Francis, UK, 1996.
11. J.-S. R. Jang. ANFIS(1993).Adaptive-Network-Based Fuzzy Inference System IEEE Trans. on Systems, Man, and Cybernetics, 23(3):665–685.

Chapter-6

Adaptive Fuzzy Control

6.1 INTRODUCTION

Adaptive control is a method of designing a controller with some adjustable parameters and an embedded mechanism for adjusting these parameters. Adaptive controllers have been used mainly to improve the controller's performance online. For each control cycle, the adaptive algorithm is normally implemented in three basic steps:

(i) observable data are collected to calculate the controller's performance,
(ii) the controller's performance is used as a guidance to calculate the adjustment to a set of controller parameters and
(iii) the controller's parameters are then adjusted to improve the performance of the controller in the next cycle.

Normally, an adaptive controller is designed based on one of the available techniques. Each technique is originally designed for a specific class of dynamic systems. The controller is then adjusted as data are collected during run time to extend its effectiveness to control a larger class of dynamic systems. Actually, 90% of the controllers in operation today are PID controllers (or at least some form of PID controller like a P or PI controller). This approach is often viewed as simple, reliable, and easy to understand.

Conventional (classical) proportional-integral-derivative (PID) controllers are perhaps the most well-known and most widely used controllers in modern industries: statistics has shown that more than 90% controllers used in industries are PID or PID-type of controllers. Generally, PID controllers have the merits of being simple, reliable and effective. These consume lower cost but are very easy to operate. Besides, for lower-order linear systems (plants, processes), PID controllers have remarkable set-point tracking performance and guaranteed stability. Therefore, PID controllers are very popular in real-world applications.

In this chapter, we first give a brief review on the conventional PID controllers and introduce the design and stability analysis of a new type of fuzzy PID controller. We also compare these two classes of PID controllers and discuss their advantages as well as limitations. It is seen that the fuzzy PID controllers are generally superior to the conventional ones, particularly for higher-order, time-delayed, nonlinear systems, and for those systems that have only vague mathematical models which the conventional PID controllers are difficult, if not impossible, to handle. For lower-order linear systems, it will be seen that both conventional and fuzzy

PID controllers work equally well, so in these simple cases conventional PID controllers are recommended due to their simple structures.

The price to pay for the success of the fuzzy PID controllers is that their design methods are slightly more advanced and their resulting formulas are somewhat more complicated (e.g., containing variable control gains in contrast to the conventional PID controllers where the control gains are constant). It may be seen that although the fuzzy PID controllers are designed by fuzzy mathematics, their final form as controllers are conventional controllers. As such, those can be used to directly replace the conventional ones in applications. Moreover, their variable control gains are self-tuned formulas that have adaptive capability to handle time-delay effects, nonlinearities, and uncertainties of the given system.

6.2. CONVENTIONAL PID CONTROLLERS

In this section, we provide a brief review on several basic types of conventional PID controllers and their configuration, design methods and stability analysis, which are needed to introduce the fuzzy PID controller later, where the fuzzy PID controllers are natural extensions of the Conventional ones: they have the same structure but are defined based on fuzzy mathematics and fuzzy control strategies. Individually, the conventional proportional (P), integral (I), and derivative (D) controllers for controlling a given system (plant, process) have the structures shown in Figure 6.1(a), (b), and (c), respectively, where r = r(t) is the reference input (set-point), y = y(t) is the controlled system's output, e = e(t) = r(t) – y(t) is the set-point tracking error, and u = u(t) is the control action (output of the controller) which is used as the input to the system. For Fuzzy controller we consider two domains the frequency domain and time domain.

In the time domain, there are the following solutions:

(i) P-controller $u(t) = K_P\, e(t)$;

(ii) I-controller $u(t) = K_I \; K_I \int_0^e \tau(t)d\tau$

(iii) D-controller $u(t) = K_D \dfrac{d}{dt} e(t)$

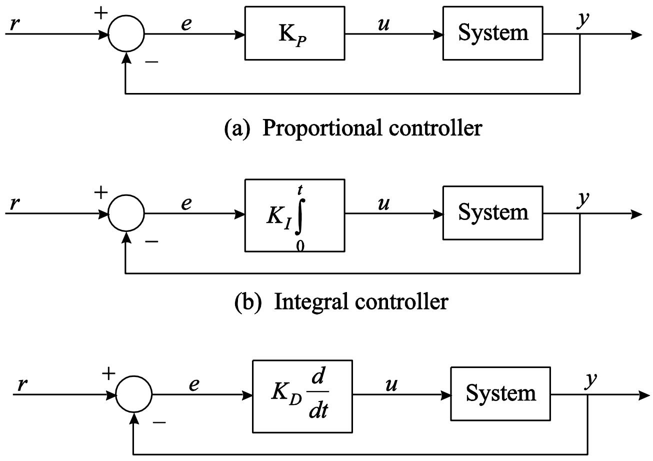
In frequency domain for the individual P, I, and D controllers:

(i) P-controller $U(s) = K_P\, E(s)$

(ii) I-controller $U(s) = (K_I / s)\, E(s)$ and

(iii) D-controller $U(s) = K_D\, s\, E(s)$.

Here L is the Laplace transform L{.} of a continuous-time signal, or the z-transform Z{.} of a discrete-time signal. Thus, U(s) = L{ u(t) } and E(s) = L{ e(t) }, with zero initial conditions.

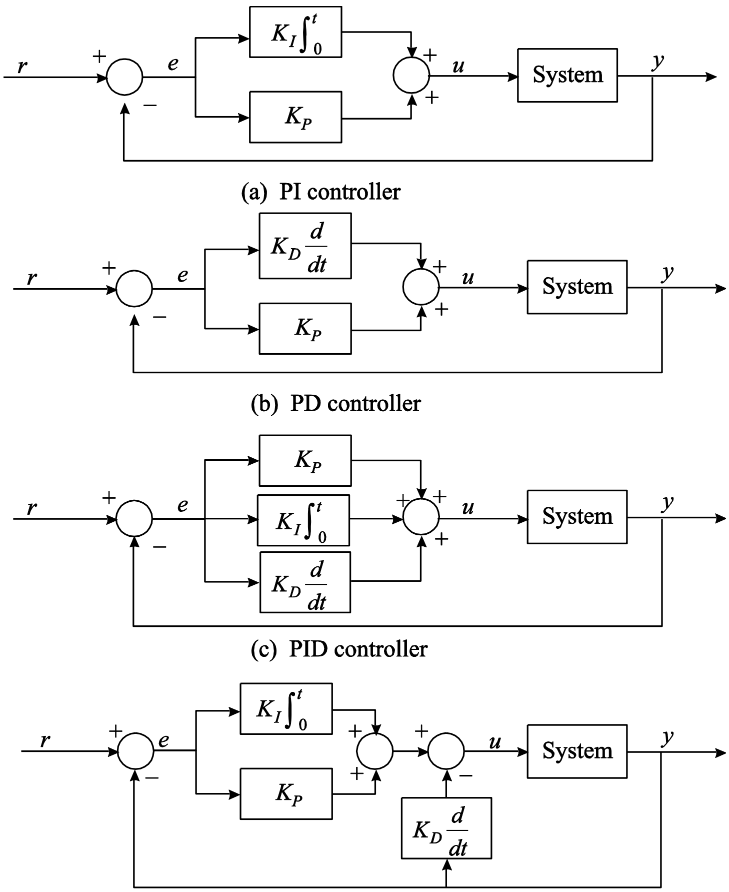


(a) Proportional controller

(b) Integral controller

(c) Derivative controller

Figure6.1. Proportional-integral-derivative controllers.



(a) PI controller

(b) PD controller

(c) PID controller

Figure 6.2 Some typical combinations of P, I and D controllers.

There are some basic and typical combinations of P, I, and D controllers

In the time domain:

(iv) PI-controller $u(t) = K_P e(t) + K_I \int_0^t e(\tau)dt$

(v) PD-controller $u(t) = K_P e(t) + K_D \frac{d}{dt} e(t)$

(vi) PID-controller $u(t) = K_P e(t) + K_I \int_0^t e(\tau)dt + K_D \frac{d}{dx} e(t)$

(vii) The PI+D-controller $u(t) = K_P e(t) + K_I \int_0^t e(\tau)dt - K_D \frac{d}{dt} y(t)$

In the frequency domain:

(iv) PI-controller $U(s) = K_P E(s) + (s / K_I) E(s)$;

(v) PD-controller $U(s) = K_P E(s) + K_D s E(s)$;

(vi) PID-controller $U(s) = K_P E(s) + s K_I E(s) + K_D s E(s)$;

(vii) PI+D-controller $U(s) = K_P E(s) + (s/K_I) E(s) – K_D s Y(s)$,

where $Y(s) = L\{ y(t) \} = L\{ r(t) – e(t) \} = R(s) – E(s)$.

It may be noted that the PID controller shown in Figure 6.2(c) is not a good combination of the three controllers in practice, since if the error signal e(t) has discontinuities, then the D-controller produces very bad (even unbounded) responses. A practical combination of the three controllers is the PI+D controller has been shown in Figure 6.2(d), where the system output signal y(t) is usually smoother than the error signal

To show how a PID-type of controller works and how to design such a controller to perform set-point tracking with guaranteed stability, we study simple Greg Viot's Fuzzy Cruise controller examples given below and explain PI, PD, PID and PI+D control in frequency domain and time domain.

Example 6.2.1: Greg Viot's Fuzzy Cruise Controller:
G-V controller is used to maintain a vehicle at a desired speed. This system consists of two fuzzy inputs, namely speed difference and acceleration and one fuzzy output, namely throttle control. It is shown in the following Figure 6.3

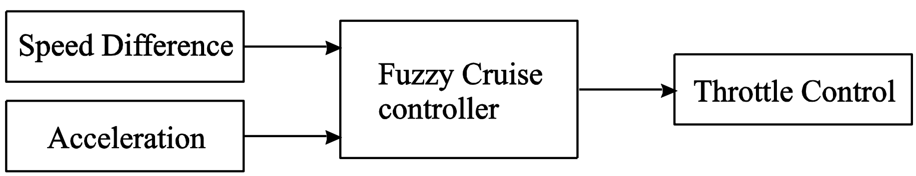


Fig. 6.3 : Fuzzy Cruise Controller

The fuzzy sets which characterize the inputs and outputs are given in the following figures: (i)Speed difference in Fig. (6.4 – a), (ii) Acceleration in Fig. (6.4 – b) and (iii) Throttle control in Fig. (6.4 – c), where NL - Negative large; PM - Positive medium; NS - Negative small; ZE- Zero; PL - Positive large; NM - Negative medium; PS- Positive small;

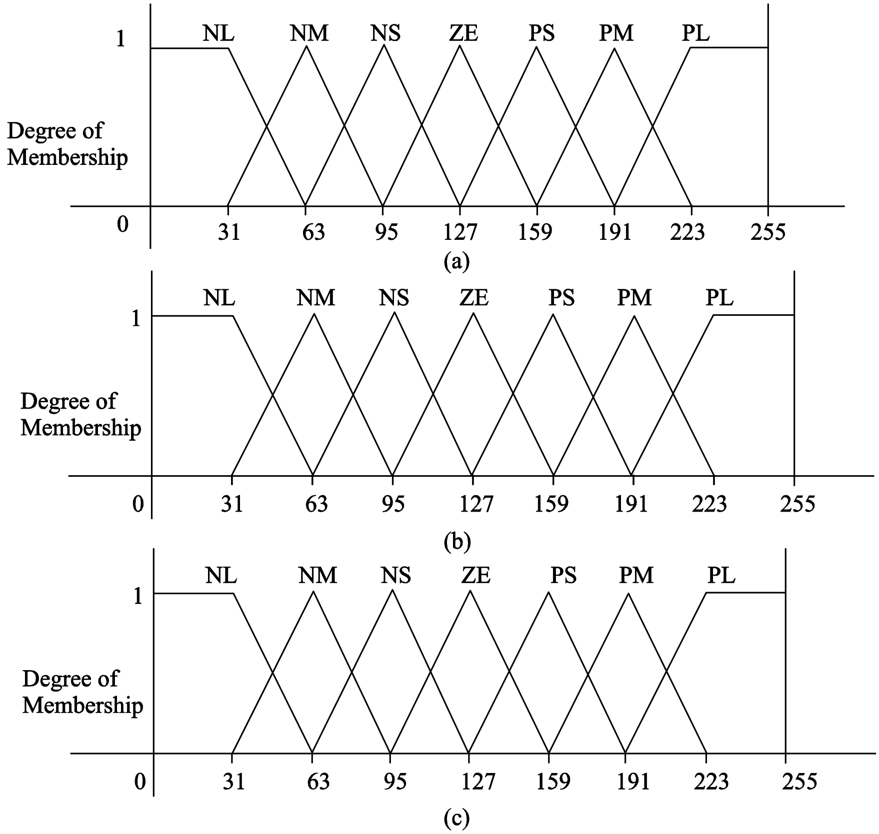


Fig.6.4 (a), (b) and (c) Speed difference, Acceleration, and Throttle control

A sample fuzzy rule base R we consider for governing the cruise control is given in the following table 6.1:

Table 6.1 Sample Cruise Control Rule Base

Rule 1	If (speed difference is NL) and (acceleration is ZE) then (throttle control is PL)
Rule 2	If (speed difference is ZE) and (acceleration is NL) then (throttle control is PL)
Rule 3	If (speed difference is NM) and (acceleration is ZE) then (throttle control is PM)
Rule 4	If (speed difference is NS) and (acceleration is PS) then (throttle control is PS)
Rule 5	If (speed difference is PS) and (acceleration is NS) then (throttle control is NS)
Rule 6	If (speed difference is PL) and (acceleration is ZE) then (throttle control is NL)
Rule 7	If (speed difference is ZE) and (acceleration is NS) then (throttle control is PS)
Rule 8	If (speed difference is ZE) and (acceleration is NM) then (throttle control is PM)

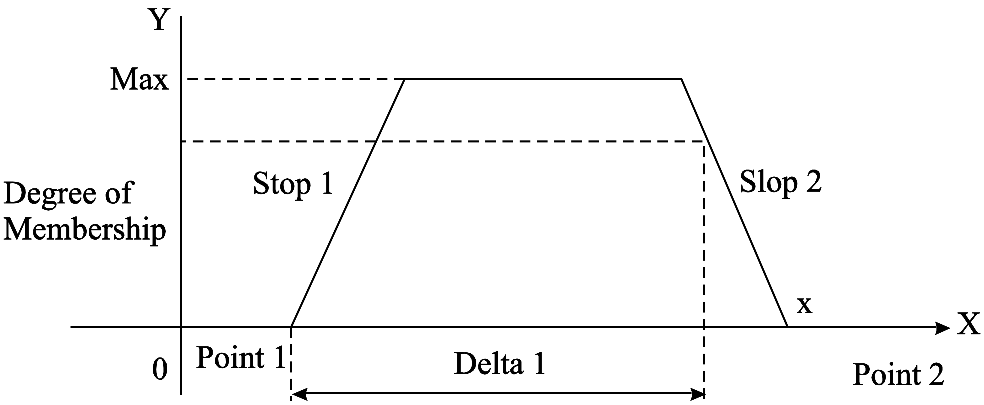


Fig. 6.5

For the fuzzification of inputs, i.e., to compute the membership for the antecedents, we use the Figure 6.5 and compute the followings:

delta 1 = x - point 1 and delta 2 = point 2 – x

$$\text{Degree of membership}\begin{cases} =0 & \text{If } (\text{Delta}\,1<0) \text{ or } (\text{Delta} \leq 0), \\ =\min\begin{pmatrix}\text{Delta } 1*\text{slope}1 \\ \text{Delta } 2*\text{ slope}\,2,\text{Max}\end{pmatrix} & \text{Otherwise}\end{cases}$$

Here, x is the system input and has its membership function values computed for all fuzzy sets. For a measured value of the speed difference the membership function of each of the set is computed using the above formula. Similarly, for each of the other system inputs (acceleration), the fuzzy membership function values are recorded.

Let the measured normalized speed difference be 100 and the normalized acceleration be70, then the computation of the fuzzy membership values i.e. fuzzified inputs are shown in the following Figure 6.6:

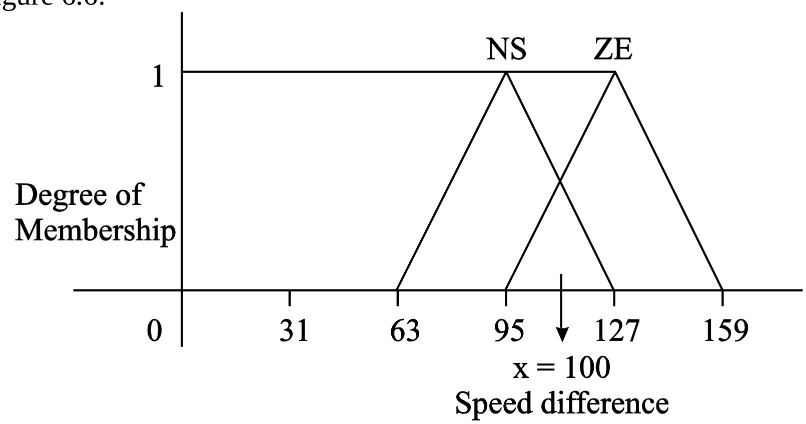


Fig 6.6: Fuzzification of speed difference = 100

The computations of the fuzzy membership values for the given inputs speed difference (x = 100), using the qualifying fuzzy sets shown in the above figure 8.4. Fuzzy membership function of x for NS, where

Delta 1 = 100 - 63 = 37; Delta 2 = 127 - 100 = 27 .,
Slope 1 = 1/32 = 0.03125; Slope 2 = 1/32 = 0.03125
The degree of the membership function is given by

$$\text{Degree of membership}\begin{cases} =0 & \text{If (Delta}\,1<0)\text{ or (Delta }\leq 0), \\ =\min & \begin{pmatrix}\text{Delta }1*\text{slope}1 \\ \text{Delta }2*\text{ slope}\,2,\text{Max}\end{pmatrix}\text{Else}\end{cases}$$

$$\mu_{NS}(x) = \min\begin{pmatrix}37*0.03125 \\ 27*0.03125\end{pmatrix} = 0.8438$$

Fuzzy membership function of x for ZE, where
Delta 1 = 100 - 95 = 5; Delta 2 = 159 -100 = 59
Slope 1 = 1/32 = 0.03125; Slope 2 = 1/32 = 0.03125

$$\mu_{NS}(x) = \min\begin{pmatrix}5*0.03125 \\ 59*0.03125\end{pmatrix} = 0.1563$$

The membership function of x with the remaining fuzzy sets, namely NL,......, NM, PS,......is zero.
Similarly for acceleration (x = 70) the qualifying fuzzy sets are shown in the following Figure 6.7.

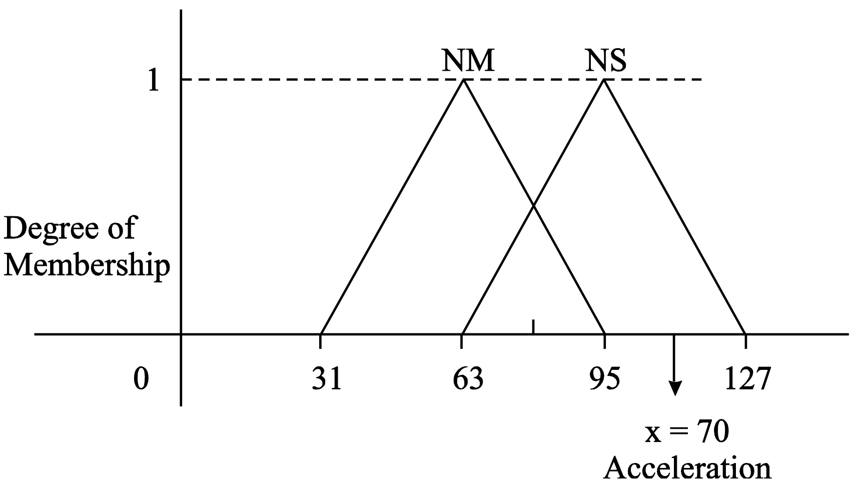


Fig 6.7: Fuzzification of acceleration.

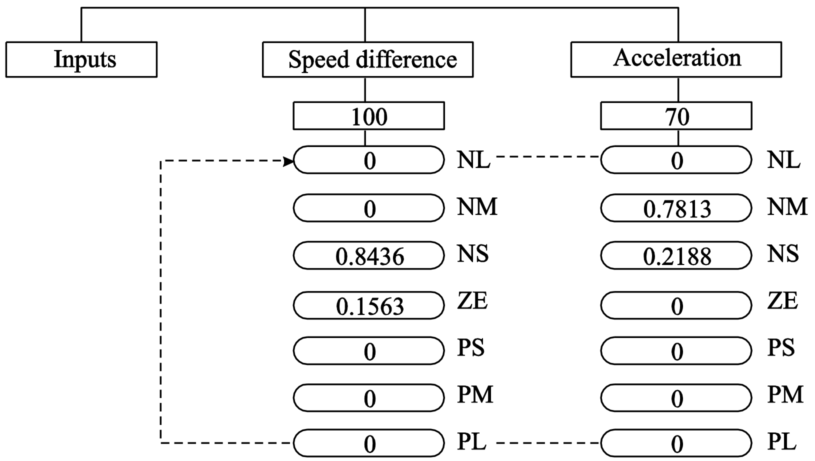


Fig 6.8 Fuzzy membership values for speed difference = 100 and acceleration = 70

The fuzzy membership function of x = 70 for NM is
μ_{NM} (x) = 0.7813 and for NS is μ_{NS}(x) = 0.2188.
Thus, using the fuzzy rule base, the rule strengths are given by the following table:

Table 6.2: Sample Cruise Control Rule Base

Rule l	If (speed difference is NL) and (acceleration is ZE) then (throttle control is PL)
Rule 2	If (speed difference is ZE) and (acceleration is NL) then (throttle control is PL)
Rule 3	If (speed difference is NM) and (acceleration is ZE) then (throttle control is PM)
Rule 4	If (speed difference is NS) and (acceleration is PS) then (throttle control is PS)
Rule 5	If (speed difference is PS) and (acceleration is NS) then (throttle control is NS)
Rule 6	If (speed difference is PL) and (acceleration is ZE) then (throttle control is NL)
Rule 7	If (speed difference is ZE) and (acceleration is NS) then (throttle control is PS)
Rule 8	If (speed difference is ZE) and (acceleration is NM) then (throttle control is PM)

Rule 1: min (0, 0) = 0
Rule 2: min (0.1563, 0) = 0
Rule 3: min (0, 0) = 0
Rule 4: min (0.8438, 0) = 0
Rule 5: min (0, 0.2188) = 0
Rule 6: min (0,0) = 0
Rule 7: min (0.1563, 0.2188) = 0.1563
Rule 8: min (0.1563, 0.7813) = 0.1563 . :

The fuzzy output of the system is the 'fuzzy OR' of all the fuzzy outputs of the rules with the non-zero rule strengths. In the given rule base R, the computing fuzzy outputs are those of rules 7 and 8 with strengths of 0.1563 each.
To defuzzify the output, we use the centre of gravity method. The computation of CG for the two competing outputs of rules 7 and 8 with strengths of 0.1563 each.

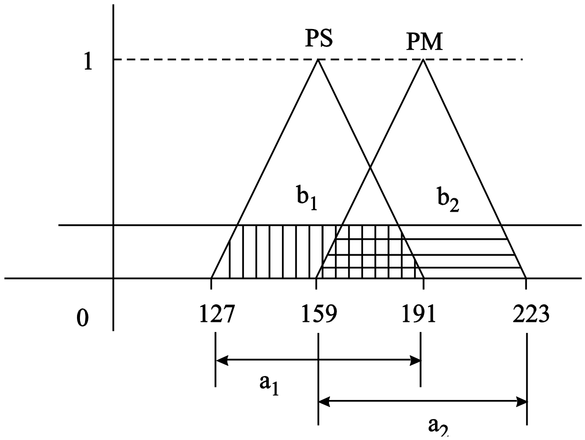


Fig 6.9: Computation of CG for fuzzy cruise control system

For the Fuzzy Set PS

X-axis centroid point = 159
Rule strength applied to determine output area = 0.1563
Shaded area = 1/2 h (a_1 + a_2) = 1/2(0.1563) (64 + 63.82) = 9.99

For the Fuzzy Set PM

X-axis centroid point = 191
Rule strength applied to determine output area = 0.1563
Shaded area = 1/2 h (a_1 + a_2) = 1/2(0.1563) (64 + 63.82) = 9.99
Therefore, weighted average (CG) = (9.99 x 159 +9.99 x 191)/ 19.98 = 175.
In crisp term, the throttle control is to be set as 175.

6.3 DESIGN OF THE PI CONTROLLER IN FREQUENCY DOMAIN

Consider the PI-control system shown in the frequency domain by Figure 6.10 where the given linear system has a first-order transfer function, 1/(as + b), with known constants a > 0 and b. The reference r is a given constant (set-point).

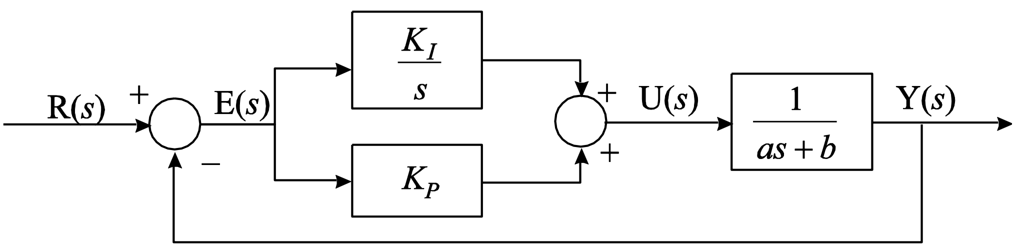


Fig. 6.10: PI-control system

In GV cruise control problem the design of the PI controller is to determine the two constant control gains, K_P (Speed Difference) and K_I (Acceleration) such that the system output Throttle Control = y(t) can track the reference: y(t) $\to$ r as t $\to\infty$, while the entire feedback control system is stable even if the given transfer function 1/(as + b) is unstable.

We first observe from Figure 6.10 that

$$U(s) = K_p E(s) + \frac{s}{K_I} E(s)$$

$$E(s) = R(s) - Y(s),$$

$$Y(s) = \frac{1}{(as+b)} U(s)$$

By combining these relations together, we can obtain the following overall system input-output relation:

$$Y(s) = H(s)R(s) \frac{K_p s + K_I}{(as+b)} R(s) \tag{6.1}$$

where H(s) is the transfer function of the overall feedback control system. It is easy to see that this transfer function has two poles:

$$S_{1,2} = \frac{-(b+K_P) \pm \sqrt{(b+K_P)^2 - 4aK_I}}{2a} \tag{6.2}$$

Thus, in our design, if we choose (note: $a > 0$)

$$K_P > -b, \tag{6.3}$$

then we can guarantee that these two poles have negative real parts, so that the overall controlled system is stable.

What we want for set-point tracking is

$$\lim_{t \to \infty} e(t) = 0\,. \tag{6.4}$$

It follows from the Terminal-Value Theorem of Laplace transforms that

$$\lim_{t \to \infty} e(t) = \lim_{|z| \to 0} s\,E(s)$$

$$= \lim_{|z| \to 0} s[R(s) - Y(s)]$$

$$= \lim_{|z| \to 0} s[1 - H(s)]R(s)$$

$$= \lim_{|z| \to 0} s . \frac{as^2 + bs}{as^2 + (b+K_P)s + K_I} . \frac{r}{s}$$

$$= 0$$

It may appear that the set-point tracking task can always be done no matter how we choose the control gain KI, provided that the other control gain, K_P, satisfies the condition (6.3), and this is true for any given constants a and b in the given system (plant). This observation is correct for this example.

However, in so doing it often happens that the system output y(t) tracks the set-point r with higher-frequency oscillations caused by the pure imaginary parts of the two poles given in (6.2). Hence, to eliminate such undesirable oscillations so as to obtain better tracking performance, we can select the

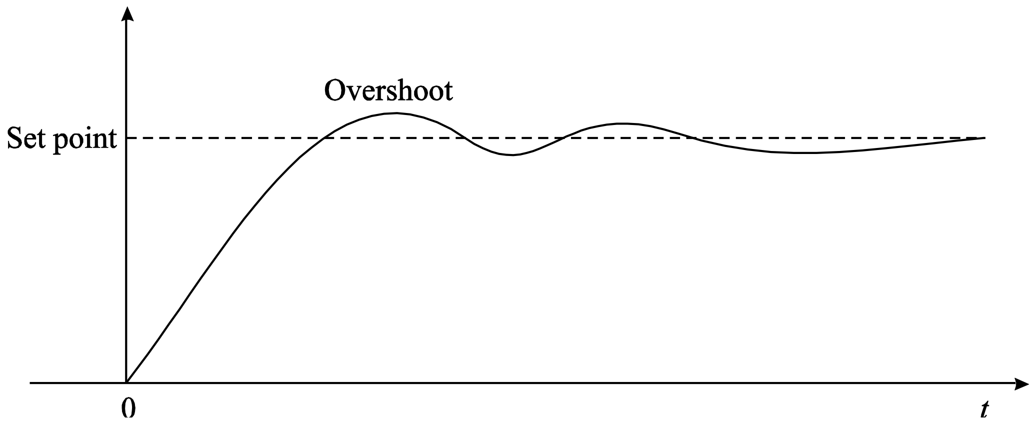


Figure 6.11 A typical tracking performance of the PI control system.

control gain K_I to zero out the pure imaginary parts of the poles s_1 and s_2.

Namely, we can force

$$(b + K_P)^2 - 4\, a\, K_I = 0,$$

which yields

$$K_I = \frac{(b + K_P)^2}{4a},$$

where K_P has been determined by (6.3-b). Thus we complete the design of the PI controller for the given plant 1/(as + b), which can be originally unstable (i.e., $b < 0$). We thus obtain an overall stable feedback control system whose output can track the set-point without oscillations, at least in theory. Its output y(t) generally has the shape as shown in Figure 6.11.

As seen, a PI controller so designed can completely eliminate both the steady-state tracking error and the transient oscillations, but may not be able to reduce the maximum overshoot in the output. On the contrary, a PD controller can generally improve the tracking performance by reducing the maximum overshoot of the output, but may not be able to eliminate the steady-state tracking error, as can be seen from the next example.

Example 6.3.1. Consider the PD-control system shown in the frequency domain by Figure 6.12, where the given linear system has a second-order transfer function, $1/(as^2 + bs + c)$, with known constants $a > 0$, b, and c. The design of the PD controller is again to determine the two constant control gains, K_P and K_D (Input), such that the system output can track the set-point while the entire feedback control system is stable even if the given transfer function $1/(as^2 + bs + c)$ is unstable.

It follows from Figure 6.12 that $U(s) = K_P E(s) + K_D s E(s)$,

$$E(s) = R(s) - Y(s),$$

$$Y(s) = \frac{1}{as^2 + bs + c} U(s),$$

so that

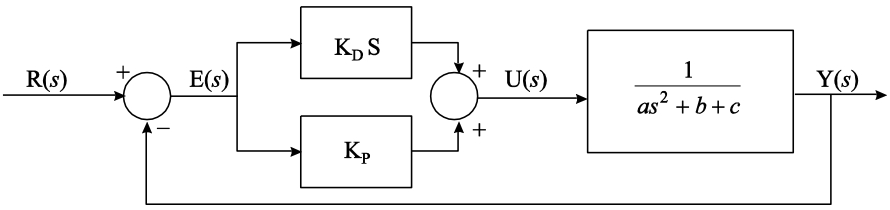


Figure 6.12 A typical tracking performance of the PI controller.

$$Y(s) = H(s)R(s) = \frac{K_P + K_D s}{as^2 + (b + K_D)s + (c + K_P)} R(s).$$

The transfer function H(s) of the overall feedback control system has two poles:

$$S_{1,2} = \frac{-(b + K_D) \pm \sqrt{(b + K_D)^2 - 4a(c + K_P)}}{2a}, \qquad (6.5)$$

and so the selection (note $a > 0$)

$$K_D > -b \qquad (6.6)$$

and

$$K_P = \frac{(b + K_D)^2}{4a} - c \qquad (6.7)$$

can guarantee the controlled system be stable and have no oscillations on the output trajectory during the set-point tracking process. However, the asymptotic tracking error for this PD controlled system is

$$\lim_{t\to\infty} e(t) = \lim_{|z|\to 0} sE(s)$$

$$= \lim_{|z|\to 0} s[I - H(s)]R(s)$$

$$= \lim_{|z|\to 0} s.\frac{as^2 + bs + c}{as^2 + (b + K_D)s + (c + K_P)}.\frac{r}{s}$$

$$= \frac{cr}{c + K_P},$$

which is non-zero if $c \neq 0$ and $r \neq 0$. This implies that the PD controller generally cannot eliminate the steady-state error in set-point tracking.

The PD controller, as compared to the PI controller, has its advantages: it can produce smaller maximum overshoot and is more sensitive (easier to tune) in general. A typical set-point tracking performance of the PD control is similar to that of the PI controller shown in Figure 6.11.

To implement a PI or PD controller on a computer, we need the digital version of the analog one discussed above. To digitize an analog controller, the following three discretization formulas can be used:

(i) The forward divided difference: $s = \dfrac{z-1}{T}$;

(ii) The backward divided difference: $s = \dfrac{1-z^{-1}}{T}$;

(iii) The trapezoidal formula: $s = \dfrac{2}{T}\dfrac{z-1}{z+1}$,

where $T > 0$ is the sampling period. These formulas transform the analog controller and the controlled system from the continuous-time frequency domain (in the Laplace-transform s-variable) to the discrete-time frequency domain (in the z-transform z-variable). In particular, the last formula maps the entire left-half (right-half) s-plane into the entire inner (outer) unit circle in the z-plane in such a way that the mapping is one-to-one and the entire imaginary axis corresponds to the entire unit circle. Hence, this formula preserves all basic properties, particularly the stability, of the original controller and controlled system, and hence is the one used most frequently. This trapezoidal formula is also known as the bilinear transform in engineering and the conformal mapping in mathematics.

Using the bilinear transform

$$s = \frac{2}{T}\frac{z-1}{z+1} \tag{6.8}$$

in the continuous-time PI controller, we obtain (see Figure 6.10):

$$\frac{K_I}{s} \to \frac{K_I T}{2}\frac{z-1}{z+1} = \frac{K_I T}{2}\frac{2-(1-z^{-1})}{1-z^{-1}} = \frac{K_I T}{2} + \frac{K_I T}{1-z^{-1}},$$

$$U(s) = (K_P + \frac{K_I}{s})E(s) \to \tilde{U}(z) = (K_P - \frac{K_I T}{2} + \frac{K_I T}{1-z^{-1}})\tilde{E}(z).$$

Let

$$\tilde{K}_P = K_P + \frac{K_I}{s} \qquad \text{and} \qquad \tilde{K}_I = K_I T. \tag{6.9}$$

Then we have

$$(1-z^{-1})\tilde{U}(z) = \tilde{K}_P(1-z^{-1})\tilde{E}(z) + \tilde{K}_I\tilde{E}(z).$$

It then follows from the inverse z-transform that

$$u(nT) - u(nT-T) = \tilde{K}_P\,[e(nT) - e(nT-T)] + \tilde{K}_I\, e(nT),$$

so that

$$\frac{u(nT)-u(nT-T)}{T} = \tilde{K}_P\frac{e(nT)-e(nT-T)}{T} + \frac{\tilde{K}_I}{T}e(nT)$$

Let, furthermore,

$$\Delta u(nT) = \frac{u(nT)-u(nT-T)}{T} \tag{6.10}$$

be the incremental control and

$$v(nT) = \frac{e(nT)-e(nT-T)}{T} \tag{6.11}$$

We arrive at

$$u(nT) = u(nT-T) + T\Delta u(nT) \tag{6.12}$$

$$\Delta u(nT) = \tilde{K}_P\, v(nT) + \frac{\tilde{K}_I}{T}e(nT) \tag{6.13}$$

which can be implemented as shown in Figure 6.13.

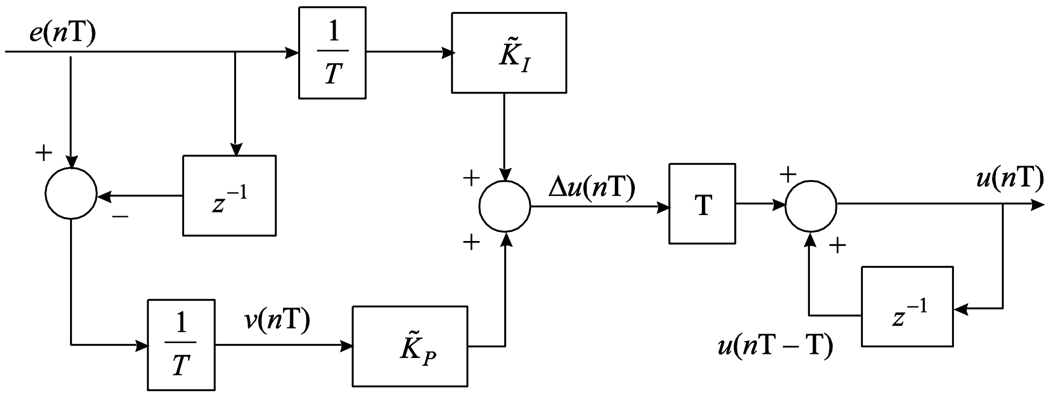


Figure 6.13 The digital PI controller

Similarly, using the bilinear transform (6.8) in the continuous-time PD controller (see Figure 6.12), we obtain

$$sK_D \to \frac{2}{T}\frac{z-1}{z+1}K_D = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}K_D,$$

$$U(s) = (K_P + sK_D)\,E(s) \to \tilde{U}(z) = (K_P + \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}K_D)\tilde{E}(z)$$

Let $\tilde{K}_P = K_P$ and $\tilde{K}_D = \frac{2}{T}K_D$ (6.14)

Then we have

$$(1+z^{-1})\tilde{U}(z) = (1+z^{-1})\tilde{K}_P\tilde{E}(z) + \tilde{K}_D(1+z^{-1})\tilde{E}(z),$$

so that the inverse z-transform gives

$$\Delta u(nT) = \tilde{K}_P d(nT) + \tilde{K}_D v(nT), \qquad (6.15)$$

$$d(nT) = \frac{e(nT) + e(nT-T)}{T},$$

$$v(nT) = \frac{e(nT) - e(nT-T)}{T}$$

This digital PD controller, described by (6.15) can be implemented as shown in the following figure 6.14.

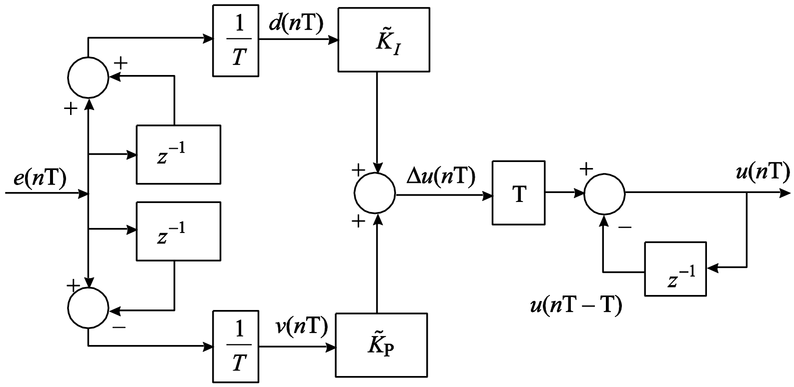


Figure 6.14 The digital PD controller.

We remark that formula (6.15) can be rewritten as

$$u(nT) = -u(nT - T) + \tilde{K}_P\,[e(nT) + e(nT - T)] + \tilde{K}_D\,[e(nT) - e(nT - T)],$$

so that

$$u(nT - T) = -u(nT - 2T) + \tilde{K}_P\,[e(nT - T) + e(nT - 2T)] + \tilde{K}_D\,[e(nT - T) - e(nT - 2T)].$$

Thus, by substituting u(nT – T) into u(nT), we obtain

$$u(nT) = u(nT - 2T) + \tilde{K}_P\,[e(nT) - e(nT - 2T)] + \tilde{K}_D\,[e(nT) - 2e(nT - T) + e(nT - 2T)]$$

$$= u(nT - 2T) + \tilde{K}_P T\,\frac{e(nT) - e(nT - 2T)}{T}$$

$$+\tilde{K}_D T^2\,\frac{e(nT) - 2e(nT - T) + e(nT - 2T)}{T^2}$$

where the two finite divided differences are the discretization of $\dot{e}(t)$ and $\ddot{e}(t)$,respectively of the continuous-time error signal e(t), under the bilinear transform. This formula shows clearly that the digital PD controller implicitly uses the discretization of both $\dot{e}(t)$ and $\ddot{e}(t)$ in conventional control theory.

The fuzzy PI, PD, and PI+D controllers to be introduced below employ the digital PI controller (6.12) and the digital PD controller (6.15), whose implementations are shown in Figures 6.13 and 6.14, respectively.

6.4 FUZZY PID CONTROLLER

Now we introduce the fuzzy PID controllers and discuss their design methods, performance evaluation, and stability analysis.

We first study the fuzzy PD controller design in detail, in which all the basic ideas, design principles, and step-by-step derivation and calculations are discussed. The fuzzy PI controller design is discussed briefly and followed by the fuzzy PI+D controller design. Having this background, many other types of fuzzy PID controllers can be designed by following similar procedures. The stability analysis of these fuzzy PID controllers will be investigated in the last section of the chapter.

Although it is possible to design a fuzzy logic type of PID controller by a simple modification of the conventional ones, via inserting some meaningful fuzzy logic IF-THEN rules into the control system (e.g., to self-tune the PID control gains with the help of a "look-up" table), yet these approaches in general complicate the overall design and do not come up with new fuzzy PID controllers that capture the essential characteristics and nature of the conventional PID controllers. Besides, they generally do not have analytic formulas to use for control specification and stability analysis.

The fuzzy PD, PI, and PI+D controllers to be introduced below are natural extensions of their conventional versions, which preserve the linear structures of the PID controllers, with simple and conventional analytical formulas as the final results of the design. Thus, they can directly replace the conventional PID controllers in any operating control systems (plants, processes). The main difference is that these fuzzy PID controllers are designed by employing fuzzy logic control principles and techniques, which are very similar to the conventional digital PID controllers. After the design is completed, all the fuzzy logic IF-THEN rules, membership functions, defuzzification formulas, etc. are not required any more in applications. What one can see is a conventional controller with a few simple formulas similar to the familiar PID controllers. Thus, in operations the controllers do not use any "look-up" table at any step, and so can be operated in real time. A control engineer who doesn't have any knowledge about fuzzy logic and/or fuzzy control systems can use them just like the conventional ones, particularly for higher-order, time-delayed, and nonlinear systems, and for those systems that have only vague mathematical models or contain significant uncertainties. The key reason, which is the price to pay, for such success is that these fuzzy PID controllers are slightly more complicated than the conventional ones, in the sense that they have variable control gains in their linear structures. These variable gains are nonlinear functions of the errors and changing rates of the error signals. The main contribution of these variable gains in improving the control performance is that they are self-tuned gains and can adapt to the rapid changes of the errors and the (changing) rates of the error signals caused by the time-delayed effects, nonlinearities, and uncertainties of the underlying system (plant, process).

6.5. FUZZY PD CONTROLLER

The overall fuzzy PD set-point tracking control system is shown in Figure 6.15, where the process under control is a discrete-time system (or a discretized continuous-time system), and r(nT) is the reference signal which can be a constant (set-point). The fuzzy PD controller inside the dashed box differs from the conventional digital PD controller (shown in Figure 6.14) as there is an extra "fuzzy controller" in the path of the incremental control signal Δu(nT). Moreover, a constant multiplication block has been changed from the sampling period T to an adjustable constant control gain Ku in order to enable the new controller one more degree of freedom in the control process (but this is not necessary). In this fuzzy PD controller, the "fuzzy controller" block is the key that improves the conventional digital PD controller's capabilities and performance.

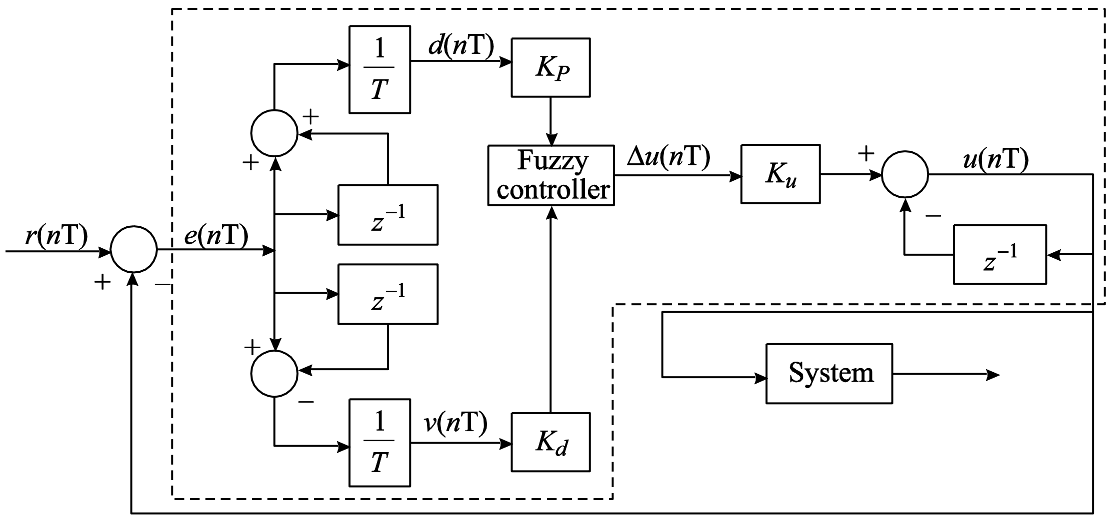


Figure 6.15 The fuzzy PD controller.

To simplify the notation, we have let the adjustable control gains be

$K_P = \tilde{K}_P$ and $K_d = \tilde{K}_D$ in Figure 6.15 (compared to Figure 6.14) and similarly let $K_i = \tilde{K}_I$ when the fuzzy PI controller is discussed later.

To illustrate how the "fuzzy controller" block works, we first introduce a constant parameter (an adjustable scalar) L > 0, and decompose the plane by L as twenty input-combination regions (IC1-IC20) as shown in Figure 6.16, where the horizontal axis is the input signal $K_pd(nT)$, and the vertical axis the input signal $K_dv(nT)$, to the "fuzzy controller." Then, according to which region the input signals ($K_pd(nT)$, $K_dv(nT)$) belong, the "fuzzy controller" block produces the following incremental outputs:

$$\Delta u(nT) = \frac{L[K_p d(nT) - K_d v(nT)]}{2(2L - K_p\,|\,d(nT)\,|)}, \qquad \text{in IC1, IC2, IC5, IC6,} \qquad (6.16)$$

$$= \frac{L[K_p d(nT) - K_d v(nT)]}{2(2L - K_d\,|\,v(nT)\,|)}, \qquad \text{in IC3, IC4, IC7, IC8,} \qquad (6.17)$$

$$= \frac{1}{2}[L - K_d v(nT)], \qquad \text{in IC9, IC10} \qquad (6.18)$$

$$= \frac{1}{2}[-L + K_p v(nT)], \qquad \text{in IC11, IC12} \qquad (6.19)$$

$$= \frac{1}{2}[-L - K_d v(nT)], \qquad \text{in IC13, IC14} \qquad (6.20)$$

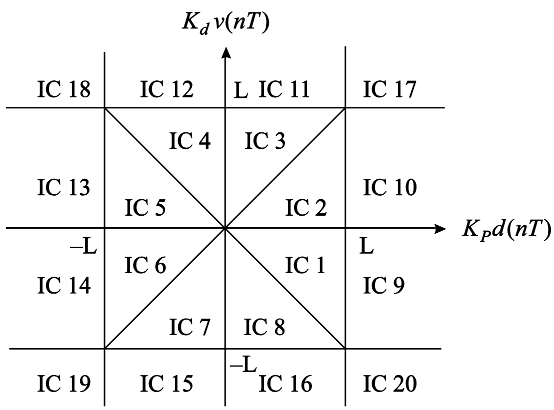


Figure 6.16 Regions of the "fuzzy controller" input-combination values.

$$= \frac{1}{2}[L + K_p v(nT)], \qquad \text{in IC15, IC16,} \qquad (6.21)$$

$$= 0, \qquad \text{in IC17, IC19,} \qquad (6.22)$$

$$= -L, \qquad \text{in IC18,} \qquad (6.23)$$

$$= L, \qquad \text{in IC20,} \qquad (6.24)$$

Here, if the input signals ($K_p d(nT)$, $K_d v(nT)$) belong to a boundary line, then either of the two neighboring regions can be used, since they will be the same (namely, all these control functions are continuous on the boundaries).

This completes the description for implementation of the fuzzy PD controller in a set-point tracking system, regardless of any knowledge about the system (plant, process) under control. The initial conditions for this control system are the following nature ones:

$$y(0) = 0,\ \Delta u(0) = 0,\ e(0) = r,\ v(0) = 0. \tag{6.25}$$

Before we discuss how to derive these formulas for the "fuzzy controller" block and in what sense this design is good, a few remarks on the above formulas are in order.

First, we note that the above nine pieces of formulas are all conventional (crisp) analytical formulas, from which one does not see any fuzzy contents (membership functions, fuzzy logic IF-THEN rules, etc.). Therefore, the "fuzzy controller" block, as well as the entire fuzzy PD controller shown in Figure 6.15, are conventional controllers: the overall control system works in the conventional manner despite the name "fuzzy." As a result, this fuzzy PD controller can be used to replace the conventional PD controller anywhere, and a control engineer can operate this fuzzy PD controller in a way completely analogous to the conventional one: what he needs to do is to tune the control gains and parameters, K_p, K_d, K_u, and L, without the need of knowledge of the fuzzy mathematics, fuzzy logic, and fuzzy control theory.

Second, the above nine pieces of formulas are continuously connected as a whole (on the boundaries between different regions shown in Figure 6.16). This can be verified by direct calculation of any two adjacent formulas on a boundary. Therefore, the control formula switching process does not have any jumps.

Third, since as usual all the control gains (K_p, K_d, and K_u) are positive real numbers, the above nine formulas can be computed from the two inputs

$$K_p d(nT) \text{ and } K_d v(nT),$$

where we have

$$K_p \mid d(nT) \mid = \mid K_p v(nT) \mid \text{ and } K_d \mid v(nT) \mid = \mid K_d v(nT) \mid.$$

Finally, but most importantly, we should point out that the first formula (6.16) preserves the linear structure of the conventional PD controller (see formula (6.15)):

$$\Delta u(nT) = \left[\frac{LK_p}{2(2L - K_p \mid d(nT) \mid)}\right] d(nT) + \left[\frac{-LK_d}{2(2L - K_p \mid d(nT) \mid)}\right] v(nT),$$

except that the two control gains here are variable gains: they are nonlinear functions of the signal $d(nT) = \frac{1}{T}[e(nT) + e(nT-T)]$. Similarly, the second formula (6.17) has the same linear structure but with two variable gains that are nonlinear functions of the signal $v(nT) = \frac{1}{T}[e(nT) - e(nT-T)]$. It is very important to note that it is exactly these variable control gains that improve the performance of the controller , for instance, when the error signal e(nT) increasesthe signal d(nT) increases, so that the control gain $L/2(2L-K_p|d(nT)|)$ also increases automatically. This means that the controller takes larger actions accordingly and hence, has certain self-tuning and adaptive capabilities in the control process. Here, it is also important to note that $K_p|d(nT)|$ does not exceed the constant L in the denominator of the control gain according to the definition of the input-combination (IC) regions shown in Figure 6.16; otherwise, the formula will switch to another one of the nine formulas.

We next discuss the design principle of the fuzzy PD controller described above, and give detailed derivations to the resulting formulas (6.16)-(6.24).

We first recall that in a standard procedure, a fuzzy controller design consists of three components: (i) fuzzification (ii) rule base establishment and (iii) defuzzification. Our design of the fuzzy PD controller follows this procedure.

In the fuzzification step, we employ two inputs: the error signal e(nT) and the rate of change of the error signal v(nT) with only one control output u(nT) (to be fed to the system under control). The input to the fuzzy PD controller, namely, the "error" and the "rate" signals, have to be fuzzified before being fed into the controller. The membership functions for the two inputs (error and rate) and the output of the controller that is used in our design are shown in Figure 6.17, which are likely the simplest possible functions to use for this purpose. Both the error and the rate have two membership values: positive and negative, while the output has three (singleton functions): positive, negative,

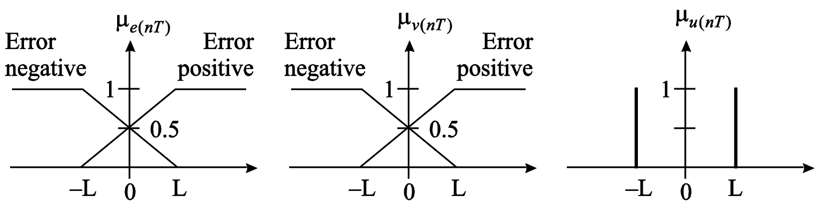


Figure 6.17 The membership functions of e(nT), v(nT), and u(nT).

and zero. The constant L > 0 used in the definition of the membership functions is chosen by the designer according to the value ranges of the error, rate, and output, which is used as a tunable parameter but can also be fixed after being determined. Note that the constant L used in these three membership functions can be different in general (according to the physical meaning of the signals in the application), but we let them be the same here in order to simplify the design.

Based on these membership functions, the fuzzy control rules that we used are the following:

$R^{(1)}$ IF error = ep AND rate = vp THEN output = oz.

$R^{(2)}$ IF error = ep AND rate = vn THEN output = op.

$R^{(3)}$ IF error = en AND rate = vp THEN output = on.

$R^{(4)}$ IF error = en AND rate = vn THEN output = oz.

Here, "output" is the fuzzy control action $\Delta u(nT)$, "ep" means "error positive," "oz" means "output zero," etc. The "AND" is the logical AND defined by

μ_A AND μ_B = min{ μ_A, μ_B }for any two membership values μ_A and μ_B on the fuzzy subsets A and B, respectively.

The reason for establishing the rules in such formulation can be understood in the same way as that described in Chapter 4, Section II, which is briefly repeated here for clarity. First, it is important to observe that since the error signal is defined to be e = r – y, where r is the reference (set-point) and y is the system output (see Figure 6.11 for the general situation in the continuous-time setting), we have $\dot{e} = \dot{r} - \dot{y} = -\dot{y}$ in the case that the set-point r is constant.

For Rule 1 ($R^{(1)}$): condition ep (the error is positive, e > 0) implies that r > y (the system output is below the set-point) and condition vp ($\dot{e}$ > 0) implies that $\dot{y} < 0$ (the system output is decreasing). In this case, the controller at the previous step is driving the system output, y, to move downward. Hence, the controller should turn around and drive the system output to move upward. It is very important to observe, however, that in our control law (see formula (6.12) and Figure 6.15):

$$u(nT) = -u(nT - T) + K_u \Delta u(nT), \qquad \ldots (6.26)$$

there is a minus sign in front of u(nT–T), which will automatically perform the expected task. For this reason, we set “output = oz” for the incremental control as the first rule, which means $\Delta u(nT) = 0$ at this step. For Rule 2 ($R^{(2)}$): condition ep (e > 0) implies that r > u and condition vn $(\dot{e} < 0)$ implies that $\dot{y} > 0$. In this case, u is below r and the controller at the previous step is driving the system output, y, to move upward. Hence, the controller needs not to take any action. But, in control law (6.26), there is a minus sign with u(nT–T) which will turn the control action to the opposite. To compensate for this, we should let Δu be positive, i.e., “output = op.”

For Rule 3 ($R^{(3)}$): condition en (e < 0) implies that r < y and condition vp ($\dot{e} > 0$) implies that $\dot{y} < 0$. In this case, y is over r and the controller at the previous step is driving the system output to move downward. Therefore, the controller needs not to take any action. Similar to Rule 2, to compensate the minus sign in (6.26), we let Δu be negative, i.e., “output = on.”

For Rule 4 ($R^{(4)}$): condition en (e < 0) implies that r < y and condition vn ($\dot{e} < 0$) implies that $\dot{y} > 0$. In this case, y is over r and the controller at the previous step is driving the system output to move upward. Hence, the controller should turn its output around to drive the system output to move downward. Because of the minus sign in (6.26), similar to Rule 1, the controller needs not to take any action: “output = oz.”

Here, we remark that one may try to let the controller take some action to speed up the system output in cases of Rules 1 and 4, but this will somewhat complicate the design of the controller.

In the defuzzification step, we use the same logical AND as mentioned above, and the membership functions shown in Figure 6.17 for the error e(nT), the rate v(nT), and the output $\Delta u(nT)$ of the “fuzzy controller” block. Because we have two (positive and negative) membership values for the error and rate, the commonly used weighted average formula is used for defuzzification, leading to

$$\Delta u(nT) = \frac{\Sigma(\text{membership value of input} \times \text{corresponding value of output})}{\Sigma(\text{membership value of input})} \qquad (6.27)$$

The defuzzification procedure and its corresponding results are now analyzed and summarized as follows:

First, we observe from (6.16)-(6.24) that, instead of the error signal e(nT), use the average error signal d(nT) and simply call K_pd(nT) and K_dr(nT) the "error signal" and the "rate signal" respectively.

Note that using the average error will not alter the above reasoning for the control rules ($R^{(1)}$)-($R^{(4)}$), at least in principle.

Second, observe that the membership functions of the average error and rate signals decompose their value ranges into twenty adjacent inputcombination (IC) regions as shown in Figure 6.16. This figure is explained as follows:

We put the membership function of the error signal (given by the first picture of Figure 6.15) over the horizontal K_pd(nT)-axis in Figure 6.16 and put the membership function of the rate of change of the error signal (given by the second picture of Figure 6.17) over the vertical K_dr(nT)-axis in Figure 6.16.

These two membership functions then overlap and form the third-dimensional picture (which is not shown in Figure 6.16) over the 2-D regions shown in Figure 6.16 in the region IC. For example, if we look upward to the K_pd(nT)-axis, we see the domain [0,L] and the membership function (in the third dimension) over [0,L] of the error signal; if we look leftward to the K_dr(nT)-axis, we see the domain [–L,0] and the membership function (in the third dimension) over [–L,0] of the rate of change of the error signal.

Next, we consider the locations of the error K_pd(nT) and the rate K_dr(nT) in the region IC1 and IC2 (see Figure 6.16). Let us look at region IC1, for example, where we have ep > 0.5 > vp (see the first two pictures in Figure 6.17). Hence, the logical AND used in ($R^{(1)}$) leads to

$$\{\text{“error = ep AND rate = vp”}\} = \min\{ ep, vp \} = vp,$$

so that Rule 1 ($R^{(1)}$) yields

$$R^{(1)} : \begin{cases} \text{the selected input membership value is vp;} \\ \text{the corresponding output value is oz} \end{cases}$$

Similarly, in region IC1, Rules 2-4, ($R^{(2)}$)-($R^{(4)}$), and the logical AND used in ($R^{(2)}$)-($R^{(4)}$) together yield

$$R^{(2)} : \begin{cases} \text{the selected input membership value is vn;} \\ \text{the corresponding output value is op} \end{cases}$$

$$R^{(3)} : \begin{cases} \text{the selected input membership value is en;} \\ \text{the corresponding output value is on} \end{cases}$$

$$R^{(4)}:\begin{cases}\text{the selected input membership value is en;}\\ \text{the corresponding output value is oz}\end{cases}$$

It can be verified that the above are true for the two regions IC1 and IC2.

Thus, in regions IC1 and IC2 it follows from the defuzzification formula (6.27) that

$$\Delta u(nT) = \frac{vp.oz + vn.op + en.on + en.oz}{vp + vn + en + en}$$

It is very important to note that if one follows the above procedure to work through the two cases, then it is found that both the last two cases give the same result of en (i.e., the two en in the above formula are not the misprint of en and ep!). To this end, by applying op = L, on = –L, oz = 0 (obtained from Figure 6.17), and the following straight line formulas from the geometry of the membership functions associated with Figure 6.17:

$$ep = \frac{K_p d(nT) + L}{2L}\ , en = \frac{-K_p d(nT) + L}{2L},$$

$$vp = \frac{K_d v(nT) + L}{2L} \text{ and } vn = \frac{-K_d v(nT) + L}{2L}\text{, we obtain}$$

$$\Delta u(nT) = \frac{L}{2[2L - K_p d(nT)]}[K_p d(nT) - K_d v(nT)]$$

Here, we note that $d(nT) \geq 0$ in regions IC1 and IC2.

In the same way, one can verify that in regions IC5 and IC6,

$$\Delta u(nT) = \frac{L}{2[2L - K_p d(nT)]}[K_p d(nT) - K_d v(nT)]\ ,$$

where $d(nT) \leq 0$ in regions IC5 and IC6.

Therefore, by combining the above two formulas, we arrive at the following result for the four regions IC1, IC2, IC5, and IC6:

$$\Delta u(nT) = \frac{L}{2[2L - K_p\,|\,d(nT)\,|]}[K_p d(nT) - K_d v(nT)],$$

which is (6.16). Similarly, if $K_p d(nT)$ and $K_d v(nT)$ are located in the regions IC3, IC4, IC7, and IC8, we have

$$K_p\,|\,d(nT)\,| \leq K_d\,|\,v(nT)\,| \leq L,$$

and, in this case,

$$\Delta u(nT) = \frac{L}{2[2L - K_d\,|\,d(nT)\,|]}[K_p d(nT) - K_d v(nT)]$$

which is (6.17). Finally, in the regions IC9-IC20, we have the corresponding formulas shown as in (6.18)-(6.24).

To this end, we have determined all the control rules and formulas for the fuzzy PID controller, with the control law (6.26) and the fuzzy control action $\Delta u(nT)$ calculated by (6.16)-(6.24) according to the different locations in Figure 6.16 of the error signal $K_p d(nT)$ and the rate of the change of the error signal $K_d v(nT)$. The initial conditions for the overall control system are the following natural values:

For the fuzzy control action $\Delta u(0) = 0$, for the system output, $y(0)= 0$, for the original error and rate signals, $e(0) = r$ (the set-point) and $v(0) = 0$ respectively as shown in (6.25).

Finally, we remark that in the steady-state situation $|e(nT)| = 0$, so that $|v(nT)| = |d(nT)| = 0$ in the denominators of the coefficients of $\Delta u(nT)$. Thus, we obtain the steady-state relations between the conventional PD control gains K^c_d and K^c_p and the fuzzy PD control gain K_p and K_d as follows:

$$K_d^c = \frac{K_u K_d}{4} \quad \text{and} \quad K_p^c = \frac{K_u K_p}{4} \tag{6.28}$$

Example 6.3. In order to compare the fuzzy PD controller with the conventional one, we first consider a first-order linear system with transfer function

$$H(s) = \frac{1}{s+1},$$

and the reference (set-point) $r = 10.0$. For this system, the fuzzy controller parameters are: $T = 0.1$, $K_d = 0.5$, $K_p = 0.5$, $K_u = 1.0$, and $L = 361.0$. The control performance is shown in Figure 6.18.

We then consider a second-order linear system with transfer function

$$H(s) = \frac{1}{s^2 + 4s + 3},$$

and the reference (set-point) $r = 10.0$. The system response, controlled by the fuzzy PD controller is shown in Figure 6.19. The controller parameters are $T = 0.1$, $K_p = 0.51$, $K_d = 0.02$, $K_u = 0.232$ and $L = 1000.0$.

In the above two cases, both conventional and fuzzy PD controllers work equally well.

To show one more case of the second-order linear system, consider one with the transfer function

$$H(s) = \frac{1}{s(s+100)},$$

which is only marginally stable. This time, let us make it more difficult by setting the reference as a ramp signal, r(t) = t. The controller parameters used are T = 0.01, K_p = 25.0, K_d = –50.0, K_u = 0.5, L = 3000.0 and the tracking tolerance is 5% of the steady-state error. For this example, it is easy to verify that the steady-state error for the conventional PD controller is given analytically by

$$e_{ss}(t) = \frac{100}{K_p^c}$$

Thus, we have to use a high gain of K^c_p = 2000.0 (along with K^c_d = 10.0) to obtain specified performance (of 5% steady-state tracking error). The fuzzy control result is shown in Figure 6.20, where the solid curve is the set point in which the dashed curve is the system output. This result demonstrates the advantage of the fuzzy PD controller over the conventional one (the former uses very small control gains) even for a second-order linear process.

Next, consider a lower-order linear system with time-delay, with transfer function

$$H(s) = \frac{1}{(100s+1)^2} e^{-3z}.$$

The comparison is shown in Figure 6.21. The conventional PD controller with K^c_p = 66.0, K^c_d = 25.0 and T = 0.1 produces the solid curve. On the contrary, the fuzzy PD controller with T = 0.1, K_p = 49.3, K_d = 5.3, K_u = 0.8, and L = 19.0 yields the dashed curve in the same figure.

Finally, the conventional and fuzzy PD controllers are compared using two nonlinear systems. The first one has the simple nonlinear model

$$\dot{y}(t) = 0.0001\,|y(t)| + u(t).$$

Here, we use two different references, namely, the constant set-point r = 1.0 and the ramp signal r(t) = t. For the constant set-point case, the fuzzy PD controller has the parameters T = 0.1, K_p = 19.5, K_d = 0.5, K_u = 0.1 and L = 20.0. The result is shown in Figure 6.22(a). The conventional PD controller, on the other hand, cannot handle this nonlinear system no matter how one changes its two constant gains. One control performance is shown in Figure 6.22(b) with K^c_p = 3.0, K^c_d = 0.1 and T = 0.1.

For the ramp signal reference case, the fuzzy PD controller produces a good tracking result with very small transient oscillation as shown in Figure 6.22(c), where Kp = 19.0, Kd = 0.5, Ku = 0.1, L = 40.0 and T = 0.1. Although the conventional PD controller also performs well after a long transient period, as shown in Figure 6.22 (d), yet its transient behavior is poorer (see Figure 6.22 (e)) as compared to the fuzzy controller (Figure 6.22 (c)).

The second nonlinear example is shown in Figure 6.23, where the system is described by

$\dot{y}(t) = -y(t) + 0.5y2(t) + u(t)$ with the constant set-point r = 2.0. The fuzzy controller is designed with $K_p = 41.9$, $K_d = 15.4$, $K_u = 0.1$, L = 239.0, and T = 0.1, which produces the tracking response shown in Figure 6.23. However, no matter how one adjusts the two constant gains of the conventional PD controller, it does not show any reasonable tracking results.

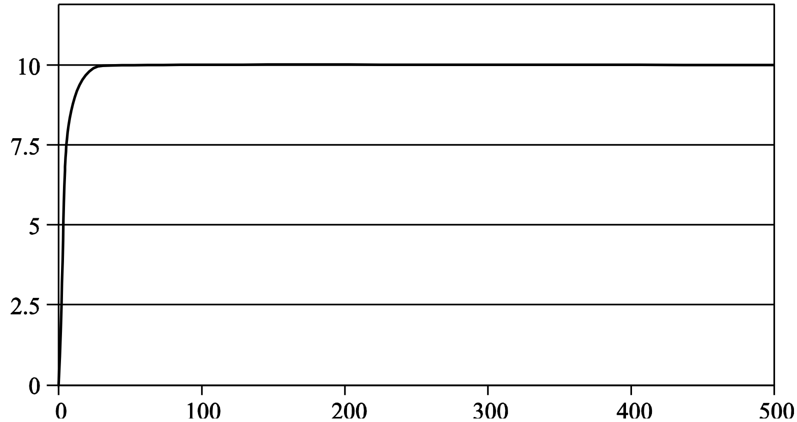


Figure 6.18 Output of a first-order linear fuzzy PD control system.

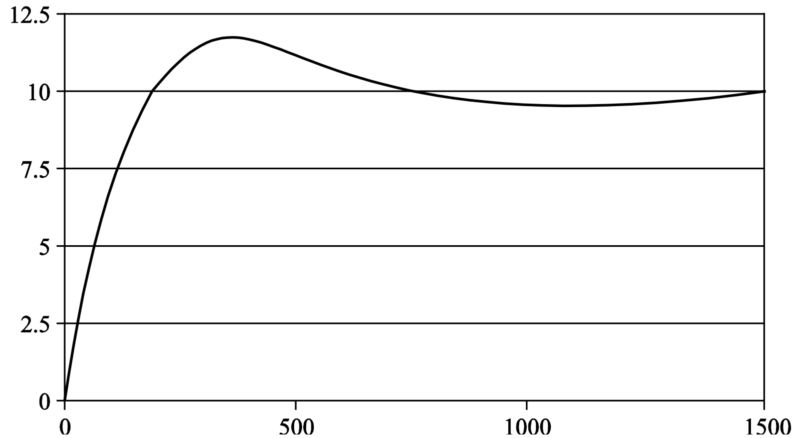


Figure 6.19 Output of a second-order linear fuzzy PD control system.

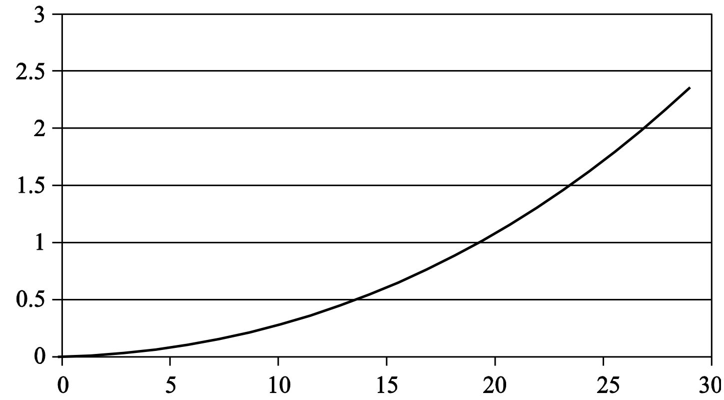


Figure 6.20 Output of a second-order linear fuzzy PD control system.

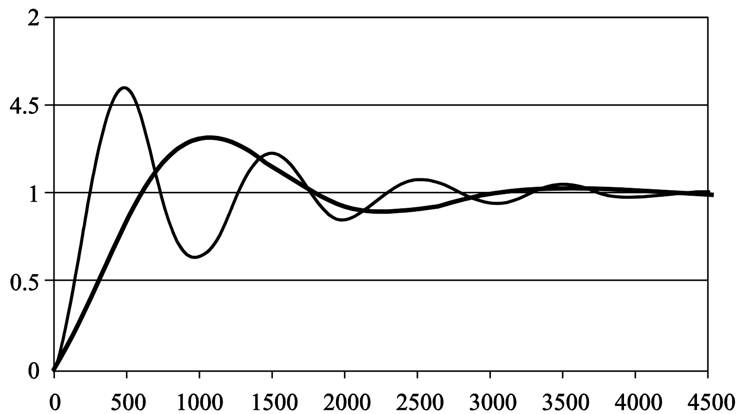


Figure 6.21 Comparison of outputs for a second-order linear fuzzy PD control system with a time delay.

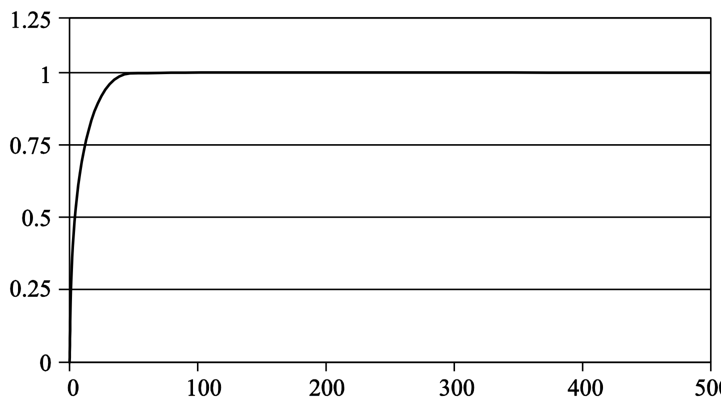


Figure 6.22(a) Output of a nonlinear fuzzy PD control system with a constant set-point.

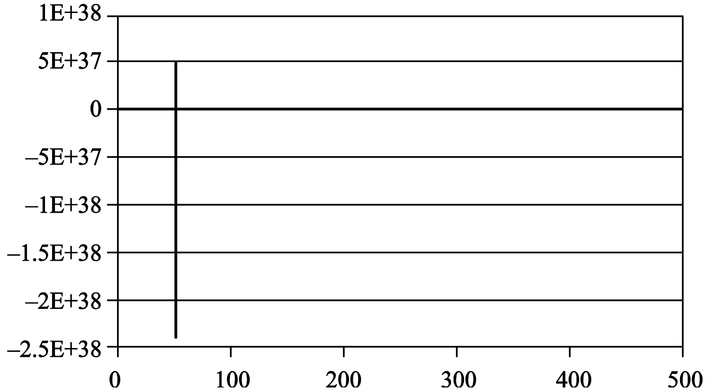


Figure 6.22(b) Output of a nonlinear conventional PD control system (with a constant set-point).

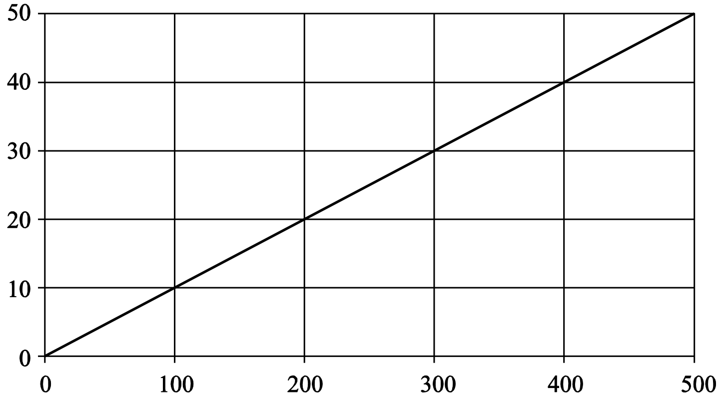


Figure 6.22 (c) Output of a nonlinear fuzzy PD control system (with a ramp set-point).

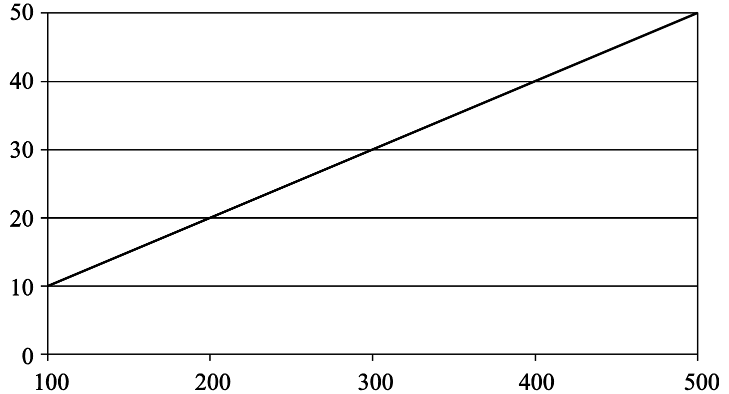


Figure 6.22 (d) Steady-state output of a nonlinear conventional PD control system (with a ramp set-point).

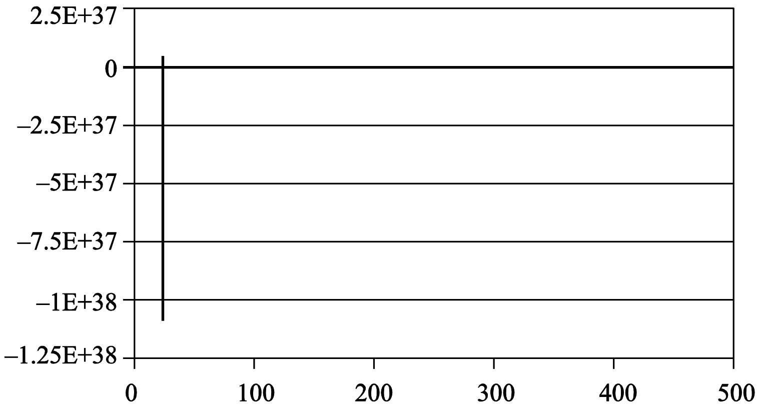


Figure 6.22 (e) Transient output of a nonlinear conventional PD control system (with a ramp set-point).

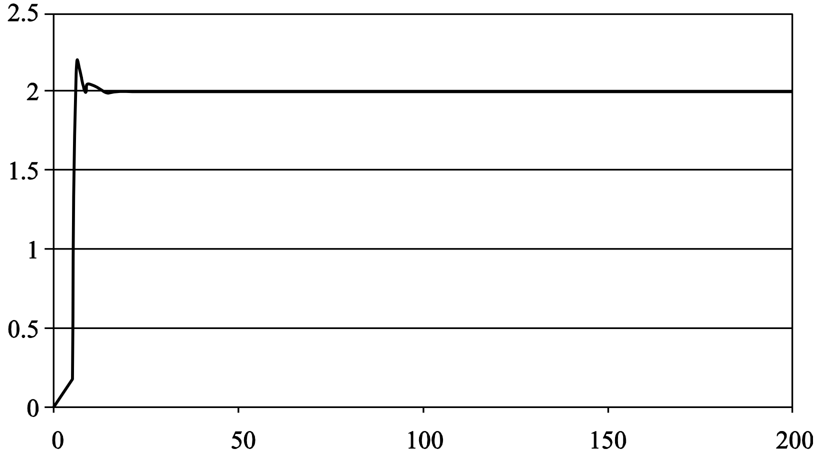


Figure 6.23 Output of a highly nonlinear PD control system.

6.6. FUZZY PI CONTROLLER

Design of the fuzzy PI controller is similar to that for the fuzzy PD controller and will be introduced briefly in this section.

The overall PI controller system is shown in Figure 6.24 (see Figure 6.13 also). In Figure 6.24, we let $K_p = \tilde{K}_P$ and $K_i = \tilde{K}_I$ as mentioned before. The "fuzzy controller" works in a way similar to that of the fuzzy PD controller. We first decompose the plane with a scalar $L > 0$ into twenty input-combination (IC) regions for the inputs $K_i e(nT)$ and $K_p v(nT)$, where $v(nT) = 1/T\,[e(nT) - e(nT-T)]$ as shown in Figure 6.25. Here, $K_i e(nT)$ and $K_p v(nT)$ are called the error signal and the rate of change of error signal respectively for convenience. Then, according to the location of the inputs ($K_i e(nT)$, $K_p v(nT)$) to the "fuzzy controller" block, the corresponding incremental control output are computed by the following formulas:

$$\Delta u(nT) = \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L - K_i |e(nT)|)}, \quad \text{in IC1, IC2, IC5, IC6,} \quad (6.29)$$

$$= \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L - K_p |e(nT)|)}, \quad \text{in IC3, IC4, IC7, IC8,} \quad (6.30)$$

$$= \frac{1}{2}[L + K_p v(nT)], \quad \text{in IC9, IC10,} \quad (6.31)$$

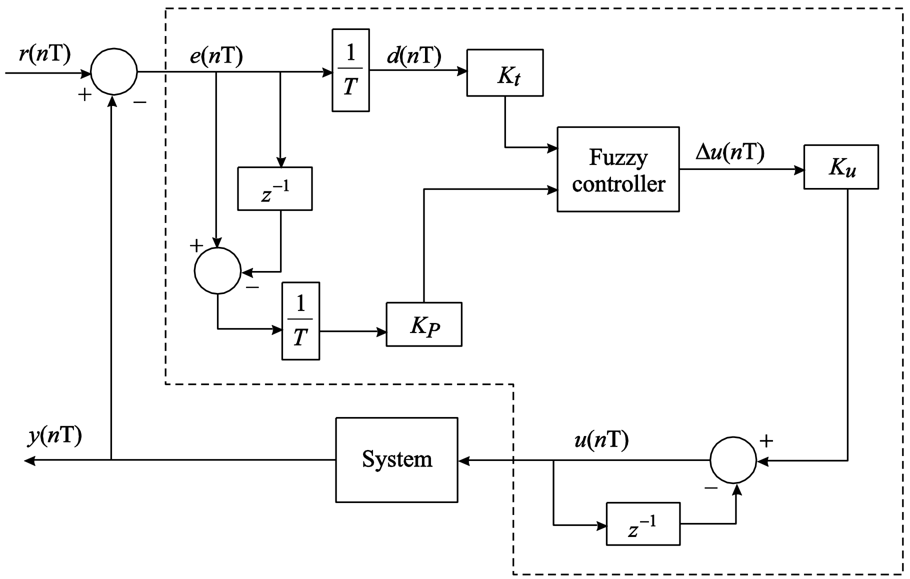


Figure 6.24 The fuzzy PI controller.

$$= \frac{1}{2}[L + K_i e(nT)], \quad \text{in IC11, IC12,} \quad (6.32)$$

$$= \frac{1}{2}[-L + K_p v(nT)], \quad \text{in IC13, IC14,} \quad (6.33)$$

$$= \frac{1}{2}[-L + K_i e(nT)], \quad \text{in IC15, IC16,} \quad (6.34)$$

$$= 0, \quad \text{in IC18, IC20,} \quad (6.35)$$

$$= -L, \quad \text{in IC17,} \quad (6.36)$$

$$= L, \quad \text{in IC19.} \quad (6.37)$$

Then, the control action output of the fuzzy PI controller, which is the control input to the system (plant, process), is given by

$$u(nT) = u(nT-T) + K_u \Delta u(nT), \tag{6.38}$$

where K_u is an adjustable constant control gain that may be fixed to be $K_u = T$ to simplify the design (but one then loses a degree of freedom in the tuning of the controller).

In a comparison of the fuzzy PI and PD control laws (6.38) and (6.26), one can find the main difference: there is a minus sign in front of u(nT–T) in the fuzzy PD controller and this minus sign makes the rule base design quite different. Of course, their incremental controls $\Delta u(nT)$ are given by formulas (6.16)-(6.24) and (6.29)-(6.37) respectively, which are also different.

Similarly, the initial conditions for the fuzzy PI controller are the following natural ones:

$$y(0) = 0,\ \Delta u(0) = 0,\ e(0) = r,\ v(0) = 0. \tag{6.39}$$

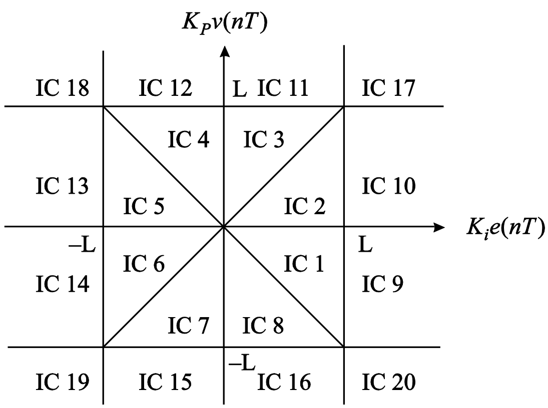


Figure 6.25 Regions of the "fuzzy controller" input-combination values.

A verification of the above fuzzy PI controller formulas can be carried out in a step-by-step procedure by mimicking the fuzzy PD controller's design given in the last section. This will be discussed in more detail in the next subsection within the PI+D controller.

We remark once again that the nine pieces of formulas (6.29)-(6.37) are conventional (crisp) formulas. So the "fuzzy controller" block as well as the entire fuzzy PI controller shown in Figure 6.24 are conventional controllers and the overall control system works in the conventional (crisp) manner despite the name "fuzzy." Therefore, this fuzzy PI controller can be used to replace the conventional digital PI controller anywhere. A control engineer can operate it without knowledge of fuzzy mathematics, fuzzy logic and fuzzy control theory. He only need to tune the control gains and parameters: K_p, K_i, K_u, and L.Finally, the computer simulations for comparison of the fuzzy and conventional PI controllers show similar results.

6.7. FUZZY PI+D CONTROLLER

The conventional analog PI+D controller is shown in Figure 6.2(d). Similar to the fuzzy PD controller design discussed in Section 6.5, we first discretize it by applying the bilinear transform, then design the fuzzy PI and a fuzzy D controller separately, and finally combine them together as a whole in the closed-loop system.

In so doing, the design of the fuzzy PI controller is the same as that mentioned briefly above, but the design of the fuzzy D controller is rather different.

Starting with the conventional analog PI+D control system shown in Figure 6.26, the output of the conventional analog PI controller in the frequency s-domain, as can be verified easily from Figure 6.10, is given by

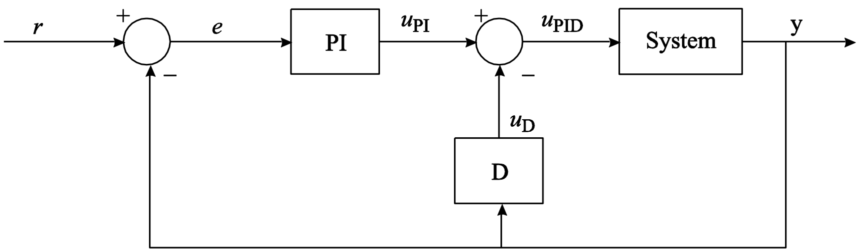


Figure 6.26 The conventional continuous-time PI+D control system.

$$U_{PI}(s) = \left[K_p^c + \frac{K_i^c}{s} \right] E(s),$$

where K^c_p and K^c_i are the proportional and integral gains, respectively, and E(s) is the tracking error signal, after taking the Laplace transform (with zero initial conditions). This equation can be transformed into the discrete version by applying the bilinear transformation (6.8), which results in the following form:

$$U_{PI}(z) = \left[K_p^c - \frac{K_i^c T}{2} + \frac{K_i^c T}{1 - z^{-1}} \right] E(z).$$

Letting

$$K_p = K_p^c - \frac{K_i^c T}{2} \text{ and } K_i = K_i^c T, \tag{6.40}$$

and then taking the inverse z-transform, we obtain

$$u_{PI}(nT) - u_{PI}(nT-T) = K_p[\, e(nT) - e(nT-T) \,] + K_i T e(nT).$$

Dividing this equation by T, we have

$$\Delta u_{PI}(nT) = K_p v(nT) + K_i e(nT), \tag{6.41}$$

where $\Delta u_{PI}(nT) = \dfrac{u_{PI}(nT) - u_{PI}(nT-T)}{T}$ and $v(nT) = \dfrac{e(nT) - e(nT-T)}{T}$

It follows that $u_{PI}(nT) = u_{PI}(nT-T) + T\Delta u_{PI}(nT)$.

In the design of the fuzzy PI controller to be discussed later, we will replace the coefficient T by a fuzzy control gain K_u,PI, so that

$$u_{PI}(nT) = u_{PI}(nT-T) + K_{u,PI}\Delta u_{PI}(nT). \tag{6.42}$$

The D controller in the PI+D control system, as shown in Figure 6.26, has y as its input and u_D as its output. It is clear that

$$U_D(s) = sK^c_d\, Y(s),$$

where K^c_d is the control gain and Y(s) is the output signal. Under the bilinear transformation, the above equation becomes

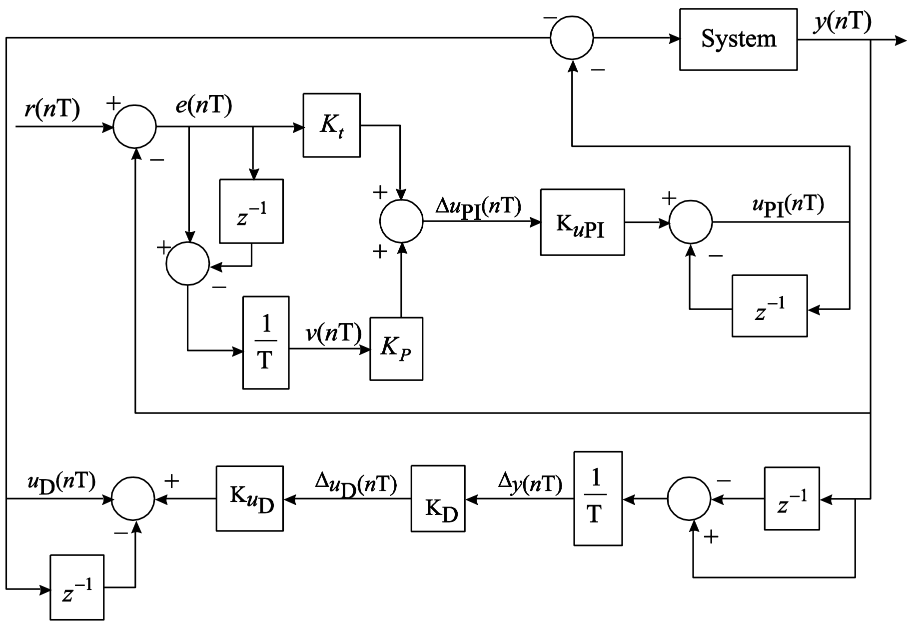


Figure 6.27 The conventional digital PI+D control system.

$$U_D(z) = \frac{2}{T}\frac{z-1}{z+1}K^c_d Y(z) \quad \text{or} \quad U_D(z) = K^c_d \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} Y(z).$$

Consequently,

$$u_D(nT) + u_D(nT-T) = \frac{2K^c_d}{T}[y(nT) - y(nT-T)]$$

Dividing this equation by T yields

$\Delta uD(nT) = K_d \Delta y(nT)$, (6.43)

where $\Delta u_D(nT) = \dfrac{u_D(nT) + u_D(nT - T)}{T}$ is the incremental control output of the fuzzy D controller, $\Delta y(nT) = \dfrac{y(nT) - y(nT - T)}{T}$ is the rate of change of the output y, and

$$K_d = \frac{2K_d^c}{T} \tag{6.44}$$

Next, we modify (6.43) by adding the signal Kyd(nT) to its right-hand side and take

$y_d(nT) = y(nT) - r(nT) = - e(nT)$ in order to obtain correct control values.

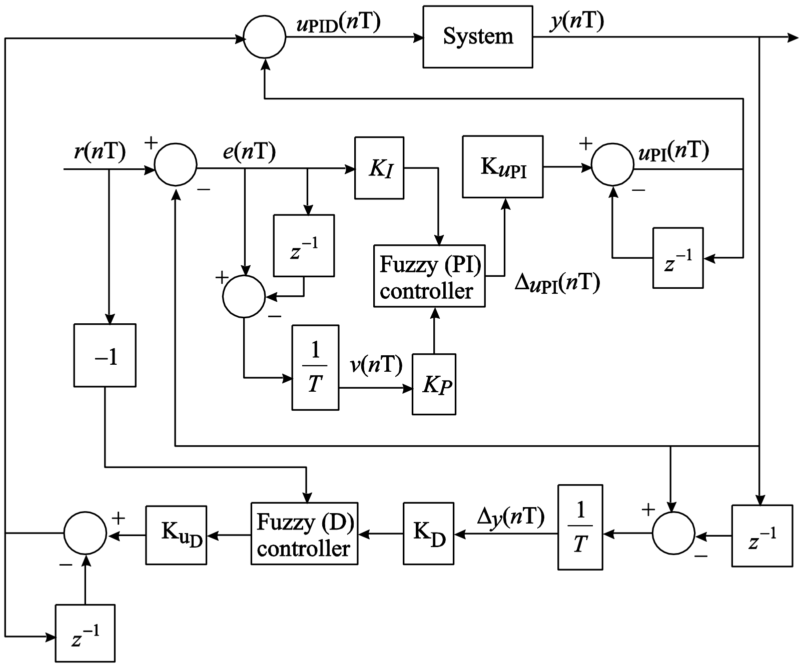


Figure 6.28 The fuzzy PI+D control system.

Thus, (6.43) becomes

$$\Delta u_D(nT) = K_d \Delta y(nT) + K y_d(nT).$$

We always use K = 1 to simplify the discussion in this section, while it is not necessary in a real design. Note that from (6.43) we consider

$$\Delta u_D(nT) = \frac{u_D(nT) + u_D(nT - T)}{T}$$

so that

$$u_D(nT) = -\,u_D(nT-T) + T\Delta u_D(nT).$$

When $\Delta u_D(nT)$ becomes a fuzzy control action later in the design, we use $K_{u,D}$ as this fuzzy control gain (which will be determined later in the design). Thus, we can rewrite the above formula as

$$u_D(nT) = -\,u_D(nT-T) + K_{u,D}\Delta u_D(nT). \tag{6.45}$$

Finally, the overall fuzzy PI+D control law can be obtained by algebraically summing the fuzzy PI control law (6.42) and fuzzy D law (6.45) together. The result is

$$u_{PID}(nT) = u_{PI}(nT) - u_D(nT)$$

or

$$u_{PID}(nT) = u_{PI}(nT-T) + K_{u,PI}\Delta u_{PI}(nT) + u_D(nT-T) - K_{u,D}\Delta u_D(nT). \tag{6.46}$$

This equation is referred to as the fuzzy PI+D control law.

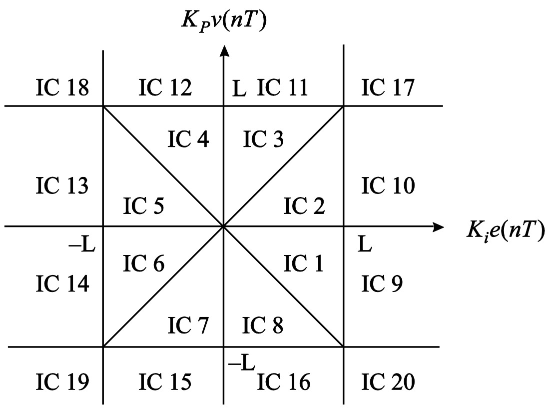


Figure 6.29 Regions of the “fuzzy (PI) controller” input-combination values.

The overall conventional PI+D control system is shown in Figure 6.27. To this end, the fuzzy PI and fuzzy D controllers will be inserted into Figure 6.27 resulting in the configuration as shown in Figure 6.28.

In summary, the fuzzy PI+D control system is implemented by Figure 6.28, in which there are five constant control gains that can be tuned: K_i, K_p, $K_{u,PI}$, K_D, and $K_{u,D}$, where one may set $K_{u,PI} = K_{u,D} = T$ to simplify the design.

In this implementation, the two incremental control actions $\Delta u_{PI}(nT)$ and $\Delta u_D(nT)$ have the analytical formulas as given by (6.47) to (6.55).

Similar to Figure 6.25, we first divide the plane as twenty regions of the “fuzzy (PI) controller” input-combination (IC) for $(K_i e(nT), K_p v(nT))$, as shown in Figure 6.29.

Then, the incremental control of the fuzzy PI controller is calculated by formulas:

$$\Delta u_{PI}(nT) = \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L - K_i\,|e(nT)|)}, \qquad \text{in IC1, IC2, IC5, IC6,} \qquad (6.47)$$

$$= \frac{L[K_i e(nT) + K_p v(nT)]}{2(2L - K_p\,|e(nT)|)}, \qquad \text{in IC3, IC4, IC7, IC8,} \qquad (6.48)$$

$$= \frac{1}{2}[L + K_p v(nT)], \qquad \text{in IC9, IC10,} \qquad (6.49)$$

$$= \frac{1}{2}[L + K_i e(nT)], \qquad \text{in IC11, IC12,} \qquad (6.50)$$

$$= \frac{1}{2}[-L + K_p v(nT)], \qquad \text{in IC13, IC14,} \qquad (6.51)$$

$$= \frac{1}{2}[-L + K_i e(nT)], \qquad \text{in IC15, IC16,} \qquad (6.52)$$

$$= 0, \qquad \text{in IC18, IC20,} \qquad (6.53)$$

$$= -L, \qquad \text{in IC17,} \qquad (6.54)$$

$$= L, \qquad \text{in IC19,} \qquad (6.55)$$

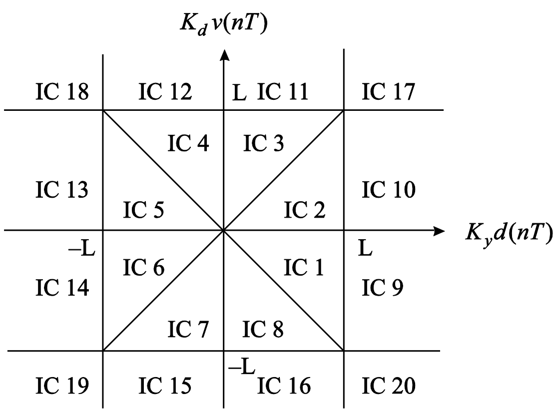


Figure 6.30 Regions of the "fuzzy (D) controller" input-combination values.

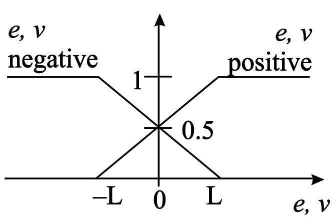


Figure 6.31 Input membership functions for the PI component.

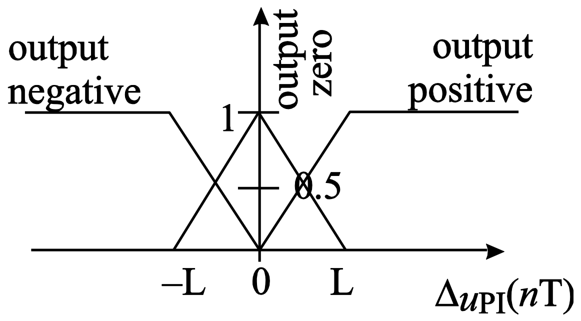


Figure 6.32 Output membership functions for the PI component.

For the fuzzy D controller, the twenty regions of the input-combination are shown in Figure 6.30. The incremental control of the fuzzy D controller is calculated by the following formulas:

$$\Delta u_D(nT) = \frac{L[Ky_d(nT)+K_d\Delta y(nT)]}{2(2L-K\,|\,y_d(nT)\,|)}, \qquad \text{in IC1, IC2, IC5, IC6,} \qquad (6.56)$$

$$= \frac{L[Ky_d(nT)+K_d\Delta y(nT)]}{2(2L-K_d\,|\,y_d(nT)\,|)}, \qquad \text{in IC3, IC4, IC7, IC8,} \qquad (6.57)$$

$$= \frac{1}{2}[L-K_d\Delta y(nT)], \qquad \text{in IC9, IC10,} \qquad (6.58)$$

$$= \frac{1}{2}[-L+Ky_d(nT)] \qquad \text{in IC11, IC12,} \qquad (6.59)$$

$$= \frac{1}{2}[-L+Ky_d(nT)], \qquad \text{in IC13, IC14,} \qquad (6.60)$$

$$= \frac{1}{2}[L+Ky_d(nT)], \qquad \text{in IC15, IC16,} \qquad (6.61)$$

$$= 0, \qquad \text{in IC18, IC20,} \qquad (6.62)$$

$$= -L, \qquad \text{in IC17,} \qquad (6.63)$$

$$= L, \qquad \text{in IC19,} \qquad (6.64)$$

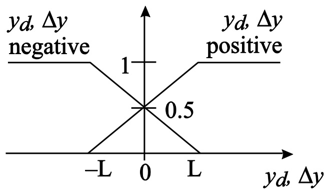


Figure 6.33 Input membership functions for the D component.

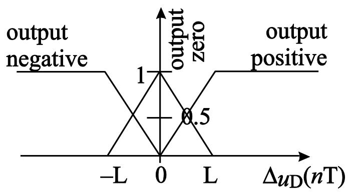


Figure 6.34 Output membership functions for the D component.

Next we describe derivations of the above formulas of the fuzzy PI and D controllers.

We first fuzzify the PI and D components of the PI+D control system individually and then establish the desired fuzzy control rules for each of them taking into consideration the interconnected PI+D fuzzy control law given in equation (6.46). The input and output membership functions of the PI component are shown in Figures 6.31 and 6.32 respectively.

The fuzzy PI controller employs two inputs: the error signal e(nT) and the rate of change of the error signal v(nT). The fuzzy PI controller has a single output, $\Delta u_{PI}(nT)$, as shown in Figure 6.28, where the constant L > 0 is a tunable parameter that can also be fixed after being determined.

Using these membership functions, the following control rules are established for the fuzzy PI controller:

$R^{(1)}$: IFe = en AND v = vn THEN PI-output =on.

$R^{(2)}$: IFe = en AND v = vp THEN PI-output =oz.

$R^{(3)}$: IFe = ep AND v = vn THEN PI-output =oz.

$R^{(4)}$: IFe = ep AND v = vp THEN PI-output =op.

In these rules, e := r – y is the error, $v = \dot{e} = 0 - \dot{y} = -\dot{y}$ is the rate of change of the error, "PI-output" is the fuzzy PI control output $\Delta u_{PI}(nT)$, "ep" means "error positive," "op" means "output positive," etc. Also, AND is the logical AND operator defined as before.

If we look at the fuzzy control law for the D component, equation (6.43), the only information it contains that is relevant to the output performance is $\Delta y(nT)$. Based on this signal alone, it is impossible to come up with a useful fuzzy control law. We therefore look for another control signal that can be used in conjunction with $\Delta y(nT)$ to provide information about the output (above or below the reference signal). For this purpose, a logical and natural choice is the negative error signal

$$y_d(nT) = -e(nT). \qquad (6.65)$$

Here, it is important to observe that yd positive (resp. negative) means that the system output y is above (resp. below) the reference r. This yd control signal is implemented as shown by the path with the –1 block in Figure 6.28 (compared with Figure 6.27).

The input and output membership functions for the fuzzy D controller are shown in Figures 6.33 and 6.34, respectively.

Similarly, from the membership functions of the fuzzy D controller, the following control rules are used for the D component:

$R^{(5)}$: IF $y_d = y_d p$ AND $\Delta y = \Delta yp$ THEN D-output =oz.

$R^{(6)}$: IF $y_d = y_d p$ AND $\Delta y = \Delta yn$ THEN D-output =op.

$R^{(7)}$: IF $y_d = y_d n$ AND $\Delta y = \Delta yp$ THEN D-output =on.

$R^{(8)}$: IF $y_d = y_d n$ AND $\Delta y = \Delta yn$ THEN D-output =oz.

In the above rules, “D-output” is the fuzzy D control output $\Delta u_D(nT)$, and the other terms are defined similar to the PI components.

These eight rules altogether yield the control actions for the fuzzy PI+D control law.

Formulation of these rules is explained as follows:

For Rule 1 (R(1)): if we look at this rule for the PI controller, condition “en” (error is negative) implies that the system output, y, is above the set-point, and “vn” (rate of error is negative) implies $\dot{y} > 0$ (meaning that the controller at the previous step is driving the system output to move upward). Since the $\Delta u_{PI}(nT)$ component of formula (6.46) contains more control terms with gain parameters than D controller, we set this term to be negative and set the $\Delta u_D(nT)$ component to be zero. Thus, the combined control action will drive the system output to move downward by Rules 1 to 5 of both controllers.

We note that one could have set the output of the D controller to be positive in order to drive the output to move downward faster. However, having both controllers’ outputs being nonzero at the same time will complicate the design of the controller. Computer simulations have demonstrated that the simple design described above performs sufficiently well in all numerical examples tested, so more sophisticated design is not discussed here.

Similarly, for Rule 2 ($R^{(2)}$), since the output is above the set-point and is moving downward, we set the “larger” component of formula (6.46), namely, the term $\Delta u_{PI}(nT)$, to be zero, and set the “smaller” component $\Delta u_D(nT)$ to be positive. Thus, the combined controller will tend to drive the system output to move downward faster by the combined action of these two rules. Rules 3 and 4 are similarly determined.

In the defuzzification step, for both fuzzy PI and D controllers, the same weighted average formula is employed to defuzzify the incremental control of the fuzzy control law:

$$\Delta u(nT) = \frac{\Sigma(\text{membership value of input} \times \text{corresponding value of output})}{\Sigma(\text{membership value of input})} \quad (6.66)$$

For the fuzzy PI controller, the value ranges of the two inputs, the error, and the rate of change of the error are actually decomposed into twenty adjacent input-combination (IC) regions, as shown in Figure 6.29. This figure is understood as follows: We put the membership function of the error signal (given by the curves for e in Figure 6.31) over the horizontal $K_i e(nT)$-axis on Figure 6.29, and put the membership function of the rate of change of the error signal (given by the same curves in Figure 6.31 for v) over the vertical $K_p v(nT)$-axis on Figure 6.29. These two membership functions then overlap and form the third-dimensional picture (which is not shown in Figure 6.29) over the two-dimensional regions shown in Figure 6.29. When we look at region IC1, for example, if we look upward to the $K_i e(nT)$-axis, we see the domain [0,L] and the membership function (in the third dimension) over [0,L] of the error signal; if we look leftward to the Kpv(nT)-axis, we see the domain [–L,0] and the membership function (in the third dimension) over [–L,0] of the rate of change of the error signal. This situation is completely analogous to the analysis of the fuzzy PD controller studied in Section II.A.

The control rules for the fuzzy PI controller ($R^{(1)}$-$R^{(4)}$), with membership functions and IC regions together, are used to evaluate appropriate fuzzy control formulas for each region.

In doing so, we consider the locations of the error $K_i e(nT)$ and the rate $K_p v(nT)$ in the regions IC1 and IC2 (see Figure 6.29). Let us look at region IC1, for example, where we have e > 0.5 > v(nT) (see Figure 6.31). Hence, the logical AND used in ($R^{(1)}$) leads to

{"error = e AND rate = v"} = min{ e, v } = e, so that Rule 1 ($R^{(1)}$) yields

$$R^{(1)}:\begin{cases}\text{the selected input membership value is en;}\\ \text{the corresponding output value is on.}\end{cases}$$

Similarly, in region IC1, Rules 2-4, ($R^{(2)}$)-($R^{(4)}$) are obtained as

$$R^{(2)}:\begin{cases}\text{the selected input membership value is en;}\\ \text{the corresponding output value is oz.}\end{cases}$$

$$R^{(3)}:\begin{cases}\text{the selected input membership value is vn;}\\ \text{the corresponding output value is oz.}\end{cases}$$

$$R^{(4)}:\begin{cases}\text{the selected input membership value is vp;}\\ \text{the corresponding output value is op.}\end{cases}$$

It can be verified that the above are true for the two regions IC1 and IC2.

Thus, in regions IC1 and IC2, it follows from the defuzzification formula (6.66) that

$$\Delta u(nT) = \frac{en \times on + en \times oz + vn \times oz + vp \times op}{en + en + vn + vp}$$

It is very important to note that if one follows the above procedure to work through the two cases, then it is found that both the last two cases give the same result of en (i.e., the two en in the above formula are not the misprint of en and ep!). To this end, by applying op = L, on = –L, oz = 0 (obtained from Figure 6.32), and the following straight line formulas from the geometry of the membership functions associated with Figure 6.29:

$$ep = \frac{K_i e(nT) + L}{2L}, \qquad en = \frac{-K_i e(nT) + L}{2L},$$

$$vp = \frac{K_p v(nT) + L}{2L}, \quad \text{and} \quad vn = \frac{-K_p v(nT) + L}{2L},$$

we obtain

$$\Delta u_{PI}(nT) = \frac{L[K_i e(nT) + K_p v(nT)]}{2[2L - K_i e(nT)]}$$

Here, we note that $e(nT) \geq 0$ in regions IC1 and IC2. In the same way, one can verify that in regions IC5 and IC6:

$$\Delta u_{PI}(nT) = \frac{L[K_i e(nT) + K_p v(nT)]}{2[2L + K_i e(nT)]},$$

where it should be noted that $e(nT) \leq 0$ in regions IC5 and IC6. Thus, by combining the above two formulas, we arrive at the following control formula for the four regions IC1, IC2, IC5, and IC6:

$$\Delta u_{PI}(nT) = \frac{L[K_i e(nT) + K_p v(nT)]}{2[2L - K_i \,|\, e(nT) \,|]}.$$

Working through all regions in the same way, we obtain the PI control formulas (6.47)-(6.55) for the twenty IC regions.

Similarly, defuzzification of the fuzzy D controller follows the same procedure as described above for the PI component, except that the input signals in this case are different. The IC combinations of these two inputs are decomposed into twenty similar regions, as shown in Figure 6.30.

Similarly, by applying the value op = L, on = –L, oz = 0, and the following straight line formulas obtained from the geometry of Figure 6.34:

$$y_d p = \frac{Ky_d(nT) + L}{2L}, \qquad y_d n = \frac{-Ky_d(nT) + L}{2L},$$

$$\Delta yp = \frac{K_d \Delta y(nT) + L}{2L} \quad \text{and} \quad \Delta yn = \frac{-K_d \Delta y(nT) + L}{2L},$$

we obtain the D control formulas (6.56)-(6.64) for the twenty IC regions.

It may be noted that the constant K that multiplies the signal $y_d(nT)$ is used as a parameter for generality here, and in the derivation of $\Delta u_D(nT)$. Although it could be used as a control gain, its value is permanently set to one throughout the computer simulations shown in the next example.

Example6.4. We first apply the fuzzy PI+D controller to a lower-order linear system, to see how well it performs for such simple cases. Recall that the conventional PI+D controller is designed for linear systems, for which it works very well. We show that the fuzzy PI+D controller is as good as the conventional one for such lower-order linear systems.

Firstly, a first-order linear system is considered with transfer function

$$H(s) = \frac{1}{s+1}$$

and with controller parameters T = 0.1, K = 1.0, K_p = 1.2, K_i = 0.1, $K_{u,PI}$ = 0.2, $K_{u,D}$ = 0.01, L = 360.0. The set-point is r = 5.0. The response of the fuzzy PI+D controller for a step input is shown in Figure 6.35.

The second example is a second-order linear system with transfer function

$$H(s) = \frac{1}{s^2 + 4s + 3},$$

where the controller parameters are T = 0.01, K = 1.0, K_p = 8.0, K_d = 0.01, K_i = 1.0, $K_{u,PI}$ = 0.2, $K_{u,D}$ = 0.01, L = 1000.0, and the set-point is r = 5.0. The response of the fuzzy PI+D controller is shown in Figure 6.36.

Finally, we compare the performance of both the conventional and the fuzzy PI+D controllers, using two nonlinear systems. Although the fuzzy PI+D controller has the same linear structure as the conventional PI+D controller, the fuzzy PI+D gains are nonlinear with self-tuning capability and, therefore, have better performance in this simulation and in general.

The first nonlinear system has the following simple model:

$$\dot{y}(t) = 0.0001\,|y(t)| + u_{PID}(t),$$

with fuzzy PI+D parameters T = 0.1, K = 1.0, K_p = 1.5, K_d = 0.1, K_i = 2.0, $K_{u,PI}$ = 0.11, $K_{u,D}$ = 1.0, L = 45.0, and the set-point is r = 5.0. The result is shown in Figure 6.37. On the contrary, the conventional PI+D controller is unable to track the set-point, no matter how one changes its parameters. A typical response of the conventional PI+D controller is shown in Figure 6.38 with parameters T = 0.1, K^c_p = 19.5, K^c_i = 1.0, for r = 5.0.

The second nonlinear system used in the simulation is

$$\dot{y}(t) = y(t) + \sqrt{y(t)} = u_{PID}(t)$$

In this case, the fuzzy PI+D parameters are T = 0.1, K = 1.0, K_p = 2.0, K_d = 1.942, K_i = 1.0, $K_{u,PI}$ = 0.1, $K_{u,D}$ = 0.27, L = 350.0, and the set-point r = 5.0.

The response of this fuzzy PI+D control system to a step input is shown in Figure 6.39. The conventional PI+D controller, however, cannot yield any reasonable response, no matter how one adjusts its gains. A typical response of the conventional PI+D controller (with parameters $T = 0.1$, $K^c_p = 2.0$, $K^c_i = 1.0$, and $r = 5.0$) is shown in Figure 6.40. We remark that this result is due to the fact that the conventional PI+D controller usually has difficulties in controlling higher-order and time-delayed linear systems as well as nonlinear systems, because they are designed only for lower-order linear systems, for which they can work very well as has been widely experienced.

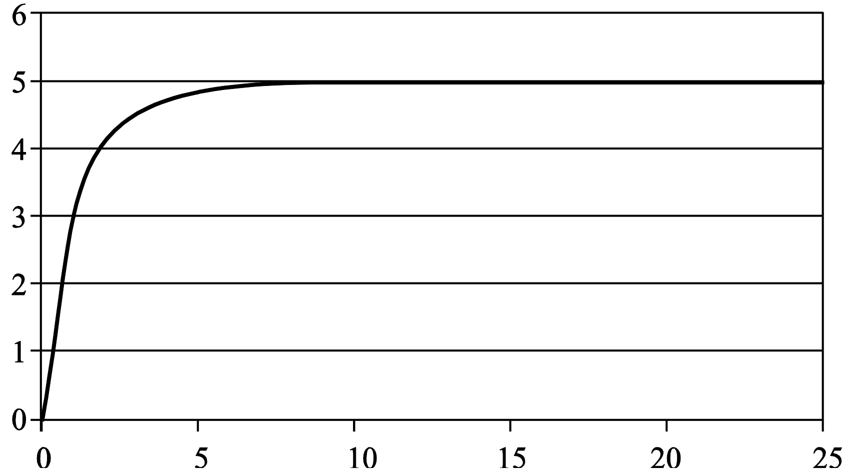


Figure 6.35 Output of a first-order linear fuzzy PI+D control system.

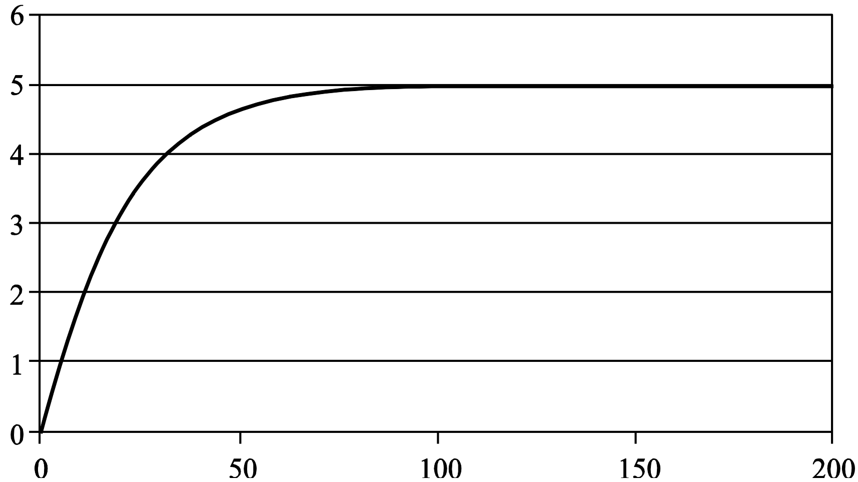


Figure 6.36 Output of a second-order linear fuzzy PI+D control system.

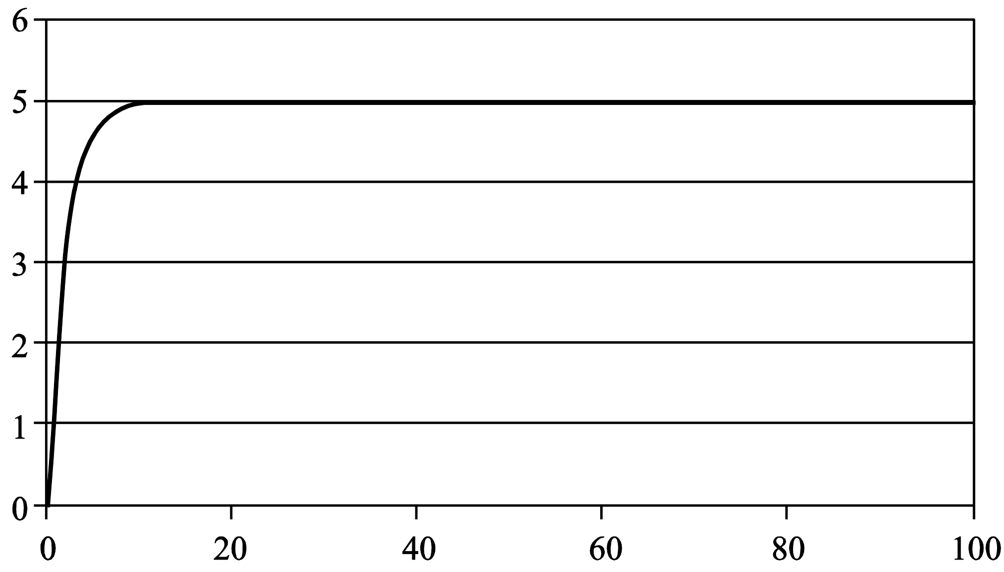


Figure 6.37 Output of a nonlinear fuzzy PI+D control system.

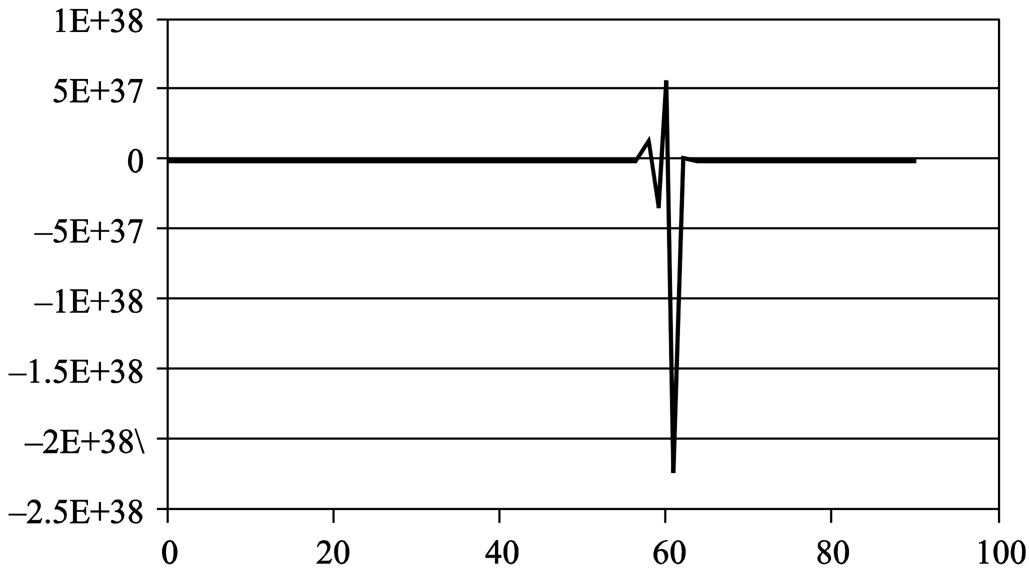


Figure 6.38 Output of a nonlinear conventional PI+D control system.

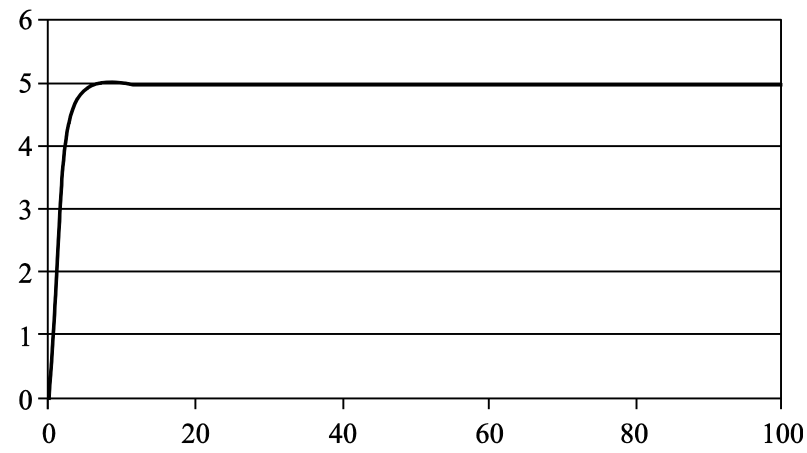


Figure 6.39 Output of a nonlinear fuzzy PI+D control system.

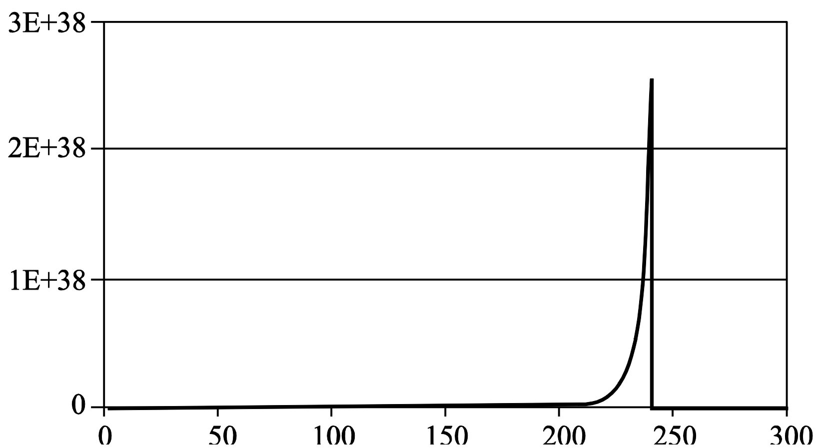


Figure 6.40 Output of a nonlinear conventional PI+D control system.

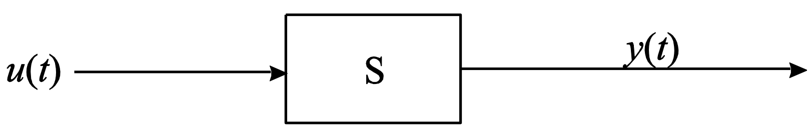


Figure 6.41 Input-output relation of a system.

6.8. STABILITY ANALYSIS OF FUZZY PID CONTROLLER

In this section, we analyze the stability of the fuzzy PI, PD, and PI+D control systems designed. Recall the Lyapunov asymptotic stability for fuzzy modeling discussed in Chapter 3, for both discrete-time and continuous-time fuzzy dynamic systems.

In order to show the bounded input bounded-output (BIBO) stability of a control system, we introduce the Small Gain Theorem in this section and discuss the BIBO stability of the fuzzy PI, PD, and PI+D control systems. We first note that differing from the Lyapunov asymptotic stability, which is usually local (in a neighborhood of an equilibrium point), the BIBO stability is global and is particularly suitable for nonlinear systems described by input-output maps. We should also note that both the Lyapunov asymptotic stability and the BIBO stability analyses provide conservative sufficient conditions, especially for nonlinear systems.

From a theoretical point of view, the larger the region of stability can be found, the better the result is. However, from the design point of view, relatively conservative stability region is actually safer and more reliable in applications. BIBO stability theory turns out to be appropriate for this purpose.

6.8.1. BIBO Stability and the Small Gain Theorem

Definition 6.1. A (linear or nonlinear) control system is said to be bounded-input bounded-output (BIBO) stable if a bounded control input to the system always produces a bounded output through the system.

Here, the bound is defined in the norm (l_2, l_∞, etc.) of the function space which we consider in the design.

Let S denote a (linear or nonlinear) system. S may be considered as a mapping which maps a control input, u(t), to the corresponding system output y(t), as shown in Figure 6.41, where S: u(t) → y(t) or y(t) = S{u(t)}.

Recall the standard L_p-spaces of signals:

$$1 \le p < \infty: \qquad L_p = \{f(t) \,|\, \int_0^\infty |f(t)|^p \, dt < \infty\},$$

$$p = \infty: \quad L_p = \{f(t) \,|\, \operatorname*{ess\,sup}_{0 \le t < \infty} |f(t)| < \infty\}$$

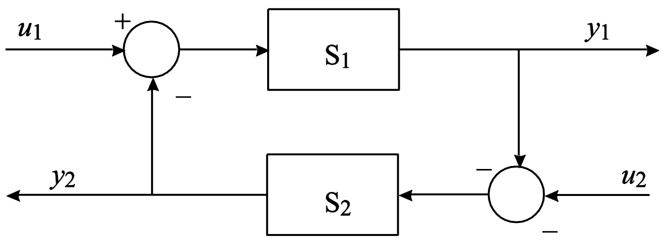


Figure 6.42 A nonlinear feedback system.

where "ess" means "essential," namely, the supirum holds except over a set of measure zero. For piecewise continuous signals, essential supirum and supirum are the same, so "ess" can be dropped from the above.

Consider a nonlinear (including linear) feedback system shown in Figure 6.42, where for simplicity it is assumed that all signals u, e, y_1, u_2, e_2, $y_2 \in R^n$.

It is clear from Figure 6.42 that

$$\begin{cases} e_1 = u_1 - S_2(e_2), \\ e_2 = u_2 + S_1(e_1) \end{cases} \tag{6.67}$$

or, equivalently,

$$\begin{cases} u_1 = e_1 + S_2(e_2), \\ u_2 = e_2 - S_1(e_1) \end{cases} \tag{6.68}$$

For this system, we have the following so-called Small Gain Theorem, which gives sufficient conditions under which a "bounded input" yields a "bounded output," where the norm || . || is the standard Euclidean norm ("length" of a vector).

Theorem6.1. Consider the nonlinear feedback system shown in Figure 6.42, which is described by the relationship (6.67)-(6.68). Suppose that there exist constants L_1, L_2, M_1, M_2, with $L_1L_2 < 1$ such that

$$\begin{cases} \| S_1(e_1) \| \le M_1 + L_1 \| e_1 \|, \\ \| S_2(e_2) \| \le M_2 + L_2 \| e_2 \| \end{cases} \tag{6.69}$$

Then, we have

$$\begin{cases} \| e_1 \| \le (1 - L_1L_2)^{-1}(\| u_1 \| + L_2 \| u_2 \| + M_2 + L_2M_1), \\ \| e_2 \| \le (1 - L_1L_2)^{-1}(\| u_2 \| + L_1 \| u_1 \| + M_1 + L_1M_2) \end{cases} \tag{6.70}$$

Proof. It follows from

$e_1 = u_1 - S_2(e_2)$, which implies that

$\| e_1 \| \le \| u_1 \| + \| S_2(e_2) \| \le \| u_1 \| + M_2 + L_2 \| e_2 \|.$

Similarly, we have

$\| e_2 \| \le \| u_2 \| + M_1 + L_1 \| e_1 \|.$

Combining these two inequalities, we obtain

$\| e_1 \| \le L_1L_2 \| e_1 \| + \| u_1 \| + L_2 \| u_2 \| + M_2 + L_2M_1,$

On using the fact $L_1L_2 < 1$, the theorem follows immediately.

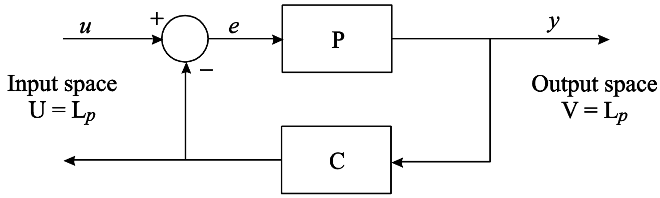


Figure 6.43 A feedback control system.

$$\| e_1 \| \leq (1-L_1L_2)^{-1}(\| u_1 \| + L_2 \| u_2 \| + M_2 + L_2M_1).$$

It is clear that the Small Gain Theorem is applicable to both continuous time and discrete-time systems, and to both SISO and MIMO systems. Hence, although its statement and proof are quite simple, it is very useful.

We next point out an interesting relation between the BIBO stability and the Lyapunov asymptotic stability. It is clear that the asymptotic stability generally implies the BIBO stability, but the reverse can also be true under some conditions.

Consider a nonlinear system described by the following first-order vectorvalued ordinary differential equation:

$$\begin{cases} \dot{x}(t) = Ax(t) - f(x(t), t), \\ x(0) = x_0 \end{cases} \tag{6.71}$$

with an equilibrium solution $\overline{x}(t) = 0$, where A is an n×n constant matrix whose eigenvalues are assumed to have negative real parts, and $f : R^n \times R^1 \rightarrow R^n$ is a real vector-valued integrable nonlinear function of $t \in [0,\infty)$. By adding and then subtracting the term Ax(t), a general nonlinear system can always be written in this form. Let

$$\begin{cases} x(t) = u(t) - \int_0^t e^{(t-\tau)} y(\tau) d\tau \\ y(x) = f(x(t), t), \end{cases} \tag{6.72}$$

with $u(t) = e^{At}x_0$. Then we can implement system (6.72) by a feedback configuration as depicted in Figure 6.43, where the error signal e(t) = x(t), the plant P(.)(t) = f(. , t), and the compensator C(.)(t) = $\int_0^t e^{(t-\tau)A}(.)(\tau)d\tau$

Theorem6.2. Consider the nonlinear system (6.72) and its associate feedback configuration as shown in Figure 6.43. Suppose that $U = V = L_p([0,\infty),R^n)$, where $1 \leq p \leq \infty$. Then, if the feedback system shown in Figure 6.43 is BIBO stable, then it is also asymptotically stable.

Proof. Since all eigenvalues of the constant matrix A have negative real parts, we have

$$| e^{tA} x_0 | \leq Me^{-\alpha t}$$

for some constants $0 < \alpha$, $M < \infty$ for all $t \in [0,\infty)$, so that $|u(t)| = | e^{tA} x_0 | \rightarrow 0$ as $t \rightarrow \infty$.

Hence, in view of the first equation defined above, i.e., x(t) = u(t) – $\int_0^t e^{(t-\tau)A} y(\tau)d\tau$, if we succeed to prove that

$$v(t) := \int_0^t e^{(t-\tau)A} y(\tau)d\tau \to 0 \qquad \text{as } t \to \infty,$$

then it will follow that

$$| x(t) | = | u(t) - v(t) | \to 0 \text{ as } t\to\infty.$$

To do so, we write

$$v(t) = \int_0^{t/2} e^{(t-\tau)A} y(\tau)d\tau + \int_{t/2}^{t} e^{(t-\tau)A} y(\tau)d\tau$$

$$= \int_{t/2}^{t} e^{\tau A} y(t-\tau)d\tau + \int_{t/2}^{t} e^{(t-\tau)A} y(\tau)d\tau$$

Then, by the Holder inequality we have

$$| v(t) | \le \left| \int_{t/2}^{t} e^{\tau A} y(t-\tau)d\tau \right| + \left| \int_{t/2}^{t} e^{(t-\tau)A} y(\tau)d\tau \right|$$

$$\le \left[\int_{t/2}^{t} | e^{\tau A} |^q \, d\tau \right]^{1/q} \left[\int_{t/2}^{t} | y(t-\tau) |^p \, d\tau \right]^{1/p} + \left[\int_{t/2}^{t} | e^{(t-\tau)A} |^q \, d\tau \right]^{1/q} \left[\int_{t/2}^{t} | y(\tau) |^p \, d\tau \right]^{1/p}$$

$$\le \left[\int_{t/2}^{\infty} | e^{\tau A} |^q \, d\tau \right]^{1/q} \left[\int_{0}^{\infty} | y(t-\tau) |^p \, d\tau \right]^{1/p}$$

$$+ \left[\int_{t/2}^{\infty} | e^{(t-\tau)A} |^q \, d\tau \right]^{1/q} \left[\int_{0}^{\infty} | y(\tau) |^p \, d\tau \right]^{1/p}$$

As all eigenvalues of A have negative real parts and the feedback system is BIBO stable from U to V, so that $y \in V = L_p([0,\infty),R^n)$. Thus, we have

$$\lim_{t\to\infty} = \int_{t/2}^{\infty} | y(\tau) |^p \, d\tau = 0 \text{ and}$$

$$\lim_{t\to\infty} = \int_{t/2}^{\infty} | y(\tau) |^p \, d\tau = 0$$

Therefore, it follows that | v(t) | → ∞ as t → ∞, completing the proof of the theorem.

6.8.2. BIBO Stability of Fuzzy PD Control Systems

We are now in a position to study the BIBO stability of the fuzzy PI, PD, and PI+D control systems.

We first discuss in detail the fuzzy PD system. Return to the fuzzy PD control system described in Figure 6.15.

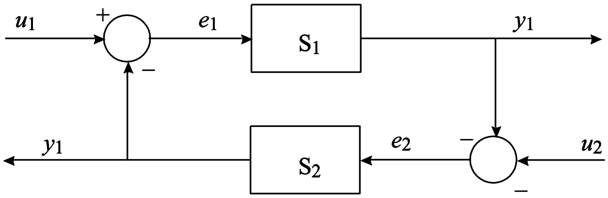


Figure 6.44 An equivalent closed-loop control system.

We consider the general case where the system under control is nonlinear, which will be denoted by N. Suppose that the fuzzy control law (6.26) together with the incremental control formula (6.16) are used, and let the reference signal be r = r(nT) for generality. By defining

$$\begin{cases} e_1(nT) = e(nT), \\ e_2(nT) = u(nT), \\ u_1(nT) = r(nT), \\ u_2(nT) = -u(nT-T), \\ S_1(e_1(nT)) = K_u \Delta u(nT), \\ S_2(e_2(nT)) = N(e_2(nT)), \end{cases} \tag{6.73}$$

it is easy to see that an equivalent closed-loop control system as shown in Figure 6.44 is

$$\begin{cases} u_1(nT) & = r(nT) = e(nT) + N(u(nT)) \\ & = e_1(nT) + S_2(e_2(nT)), \\ u_2(nT) & = -u(nT-T) = u(nT) - K_u \Delta u(nT) \\ & = e_2(nT) + S_1(e_1(nT)) \end{cases} \tag{6.74}$$

Observe that when e(nT) and r(nT) are in the regions IC1, IC2, IC5, IC6, we have formula (6.16), so that

$$\begin{aligned} & \| S_1(e_1(nT)) \| \\ & \le \frac{K_u L}{2(2L - K_p M_e)} \left[\frac{|K_p - K_d|}{T} |e_1(nT)| + |e(nT-T)| \right] \\ & = \frac{K_u L |K_p - K_d|}{2(2L - K_p M_e)} M_e + \frac{K_u L |K_p - K_d|}{2(2L - K_p M_e)} |e_1(nT)|, \end{aligned} \tag{6.75}$$

and

$$\| S_2(e_2(nT)) \| \leq \| N \| \, . \, | e_2(nT) |, \tag{6.76}$$

where $\| N \|$ is the operator norm of the given N(.) or the gain of the given nonlinear system defined as usual by

$$\| N \| := \sup_{v_1 \neq v_2, n \geq 0} \frac{| N(v_1(nT)) - N(v_2(nT)) |}{| v_1(nT) - v_2(nT) |} \tag{6.77}$$

over a set of admissible control signals that have any meaningful function norms, and M_e is defined by

$$M_e := \sup_{n \geq 1} | d(nT) | = \sup_{n \geq 1} \frac{2}{T} | e(nT) | \tag{6.78}$$

To this end, an application of the Small Gain Theorem (Theorem 6.1) yields the following sufficient condition for the BIBO stability of the nonlinear fuzzy PD control systems:

$$\frac{K_u L | K_p - K_d |}{2T(2L - K_p M_e)} \| N \| < 1 \tag{6.79}$$

When e(nT) and r(nT) are in the regions IC3, IC4, IC7, IC8, we can similarly obtain a sufficient stability condition as follows:

$$\frac{K_u L | K_p - K_d |}{2T(2L - K_p M_r)} \| N \| < 1, \tag{6.80}$$

where

$$M_r := \sup_{n \geq 1} | r(nT) | = \sup_{n \geq 1} \frac{1}{2} | e(nT) - e(nT - T) | \leq M_e \tag{6.81}$$

When e(nT) and r(nT) are in the rest of the regions, from IC9-IC20, the other incremental control formulas (6.18)-(6.24) are used. In these cases, the stability conditions are found to be

$$\begin{cases} \dfrac{K_u K_p}{2T} \| N \| < 1, \\ \dfrac{K_u K_d}{2T} \| N \| < 1 \\ \| N \| \text{ is bounded} \end{cases} \tag{6.82}$$

By combining all the above conditions together, and noting that in IC1-IC8, $K_p M_e \leq L$, and that $K_p > 0$ and $K_d > 0$, we arrive at the following result for the stability of the nonlinear fuzzy PD control systems.

Theorem6.3. A sufficient condition for the nonlinear fuzzy PD control systems to be BIBO stable is that the given nonlinear system has a bounded norm (gain) $\| N \| < \infty$ and the parameters of the fuzzy PD controller, K_p, K_d, and K_u, satisfy

$$\frac{\gamma K_m K_u}{2TL} \| N \| < 1, \tag{6.83}$$

where $\gamma = \max\{ 1, L \}$ and $K_m = \max\{ K_p, K_d \}$.

Remarks: This theorem provides a useful criterion for the design of the nonlinear fuzzy PD controller when a nonlinear process N is given. We may first choose $K_u = T$ and then find a value of K_d (or K_p) for the tracking purpose

$$e(nT) = y(nT) - r(nT) \to 0 \text{ as } n \to \infty. \tag{6.84}$$

Finally, we may determine K_d (or K_p) among all possible choices such that the inequality (6.83) is satisfied.

6.8.3. BIBO Stability of Fuzzy PI Control Systems

The analysis of the BIBO stability condition for the fuzzy PI control system is similar to that for the fuzzy PD control system discussed in the last subsection.

We again consider the general case where the system under control, N, is nonlinear. We start with the configuration shown in Figure 5.17 with the control law (6.38) and (6.39). By defining

$$\begin{cases} e_1(nT) = e(nT), \\ e_2(nT) = u(nT), \\ u_1(nT) = r(nT), \\ u_2(nT) = -u(nT - T), \\ S_1(e_1(nT)) = K_u \Delta u(nT), \\ S_2(e_2(nT)) = N(e_2(nT)), \end{cases} \tag{6.85}$$

it is easy to see that we obtain an equivalent closed-loop control system as shown in Figure 6.44 where, differing from the fuzzy PD control system, we have

$$\begin{cases} u_1(nT) & = r(nT) = e(nT) + N(u(nT)) \\ & = e_1(nT) + S_2(e_2(nT)), \\ u_2(nT) & = -u(nT - T) = u(nT) - K_u \Delta u(nT) \\ & = e_2(nT) + S_1(e_1(nT)) \end{cases} \tag{6.86}$$

Observe, moreover, that when e(nT) and r(nT) are in the regions IC1, IC2, IC5, or IC6, we have

$$\| S_1(e_1(nT)) \| = \left\| -\frac{K_u L}{2(2L - K_i \,| e_1(nT) |)} \times \left[\left(K_i + \frac{K_p}{T} \right) e_1(nT) - \frac{K_p}{T} e_1(nT - T) \right] \right\|$$

$$\leq \frac{K_u L}{2(2L - K_i M_e)} \left[\left(K_i + \frac{K_p}{T} \right) | e_1(nT) | - \frac{K_p}{T} M_e \right]$$

$$= \frac{K_u K_p M_e L}{2(2L - K_i M_e)} + \frac{K_u K_i + K_p L}{2(2L - K_i M_e)} \,| e(nT) | \qquad (6.87)$$

and

$$\| S_2(e_2(nT)) \| \leq \| N \| \,.\, | e_2(nT) |, \qquad (6.88)$$

where $\| N \|$ is the operator norm of the given N(.), or the gain of the given nonlinear system, defined as in formula (6.77), and M_e is the maximum magnitude of the error signal

$$M_e := \max\{ \sup_{n \geq 1} | e(nT) |, \sup_{n \geq 1} | e(nT - T) \} \qquad (6.89)$$

In regions IC1-IC8, $K_i M_e \leq L$.

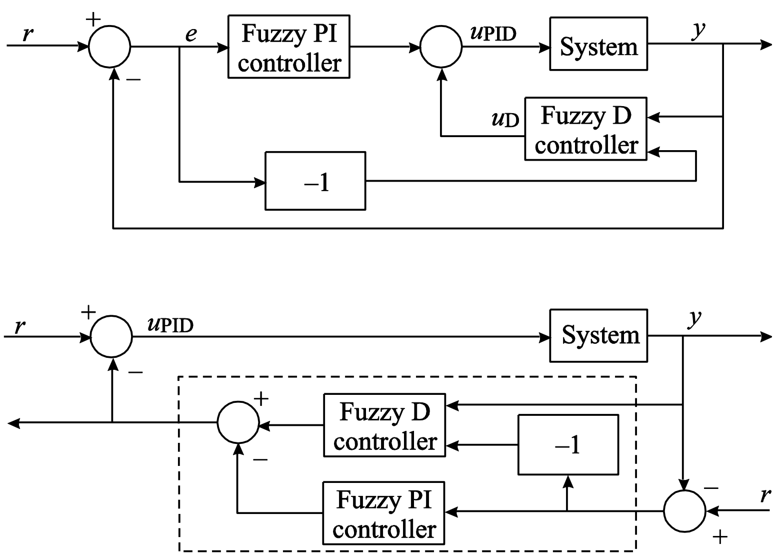


Figure 6.45 Equivalent closed-loop control systems.

To this end, an application of the Small Gain Theorem (Theorem 6.1) produces the following sufficient condition for the BIBO stability of the closed-loop nonlinear fuzzy PI control system:

$$\frac{K_u(TK_i + K_p)}{2T} \| N \| < 1 \tag{6.90}$$

When e(nT) and r(nT) are in the regions IC3, IC4, IC7, or IC8, we can similarly obtain a sufficient stability condition as follows:

$$\frac{K_u(TK_i + K_p)}{2T} \| N \| < 1 \tag{6.91}$$

When e(nT) and r(nT) are in the rest of the regions, from IC9 to IC20, the other control laws are used. In this case, the stability conditions are found to be

$$\begin{cases} \dfrac{K_u K_i}{2T} \| N \| < 1, \\ \dfrac{K_u K_p}{2T} \| N \| < 1 \\ \| N \| \text{ is bounded} \end{cases} \tag{6.92}$$

By combining all the above conditions together, we arrive at the following result:

Theorem6.4. A sufficient condition for the nonlinear fuzzy PI control system shown in Figure 6.24 to be globally BIBO stable is: (1) the given nonlinear system has a bounded norm (gain) $\| N \| < \infty$, and (2) the parameters of the fuzzy PI controller, K_p, K_d, and K_u, satisfy

$$\frac{K_u(\gamma K_i + K_p)}{2T} \| N \| < 1, \tag{6.93}$$

where $\gamma = \max\{L,T\}$.

6.8.4. BIBO Stability of Fuzzy PI+D Control Systems

The stability analysis of the fuzzy PI+D control system is in a sense to combine the results obtained individually for the fuzzy PI and PD control systems.

Consider the fuzzy PI+D control system shown in Figure 6.21. First, observe that if we disconnect the fuzzy D controller from Figure 6.21, we have the fuzzy PI control system, exactly the same as Figure 6.17. Hence, all the results obtained in the last subsection for the fuzzy PI control system apply to this case also.

For the fuzzy PI+D control system shown in Figure 6.21, one can easily verify that it is equivalent to either one of the two configurations shown in Figure 6.45. Recall from the

Small Gain Theorem (Theorem 6.1) that if we let the system denoted by S_1 and the fuzzy PI+D controller together be denoted by S_2, which is the dashed box in the second picture of Figure 6.45, then we can obtain a sufficient condition for the BIBO stability of the overall closed loop control system from the bounds

$$\begin{cases} \| S_1(u_{PID}) \leq M_1 + L_1 \| u_{PID} \|, \\ \| S_2\left(\begin{bmatrix} y \\ e \end{bmatrix}\right) \leq M_2 + L_2 \| \begin{bmatrix} y \\ e \end{bmatrix} \| \end{cases}, \tag{6.94}$$

where M_1, M_2, L_1, L_2 are constants with $L_1L_2 < 1$and has been discussed in detail in Theorem 6.1.

Here, we observe that due to the special structure of the D controller, denoted S_D, we actually have

$$\left\| S_2\left(\begin{bmatrix} y \\ e \end{bmatrix}\right) \right\| = \left\| \begin{bmatrix} S_D & -1 \\ 0 & S_{PI} \end{bmatrix} \begin{bmatrix} y \\ e \end{bmatrix} \right\| \leq \left\| \begin{matrix} S_D & -1 \\ 0 & S_{PI} \end{matrix} \right\| . \left\| \begin{bmatrix} y \\ e \end{bmatrix} \right\|$$

$$\leq \max\{ \| S_D \|, \| S_{PI} \|, 1 \} . \max\{ \| y \|, \| e \| \}.$$

Hence, a sufficient condition for the overall fuzzy PI+D control system to be BIBO stable is the worst one between the fuzzy PI and D control systems.

Namely, we may use the larger norms from the right-hand side of the above.

Thus, we will have the second equation of inequalities (6.94) in which SD and SPI are separated in M_2 and/or L_2, so that the analysis performed in the last two subsections can be repeated here for inequalities (6.94).

Note that in this case we may assume that $\max\{ \|S_D\|, \|S_{PI}\| \} \geq 1$; otherwise, the system will be stable without additional conditions by the contraction mapping principle. Under this inequality, the second condition of (6.94) can be guaranteed.

6.8.5. Graphical Stability Analysis of Fuzzy PID Control Systems

We have shown how to analytically analyze the BIBO stability of a fuzzy PID type of control system in the last few sections. Sufficient conditions so obtained are useful for controller design. Although such sufficient conditions are generally conservative, just like many other sufficient conditions derived via different methods for nonlinear systems, they provide useful guidelines for designing “safe” controllers (namely, stabilizing controllers with desirable robustness against system parameter variation and/or external disturbances).

In this section, we introduce a graphical approach to the BIBO stability analysis for PID type of fuzzy control systems. We only discuss the fuzzy PI+D control systems shown in Figure 6.28, but the methodology clearly is applicable to different types of fuzzy PID control systems.

In Figure 6.28, we observe that the control signal $u_{PID}(nT)$ to the error signal $e(nT)$ can be related implicitly through the closed-loop configuration. In the z-domain, let

$$f(z) = \frac{e(z)}{u_{PID}(z)} \tag{6.95}$$

and

$$g(z) = \frac{v(z)}{u_{PID}(z)} \tag{6.96}$$

where

$$f(z) = f_0 + f_1 z^{-1} + f_2 z^{-2} + \dots$$

$$g(z) = g_0 + g_1 z^{-1} + g_2 z^{-2} + \dots$$

are unknown (not explicitly known) but well-defined, and similarly,

$$e(z) = e_0 + e_1 z^{-1} + e_2 z^{-2} + \dots$$

$$v(z) = v_0 + v_1 z^{-1} + v_2 z^{-2} + \dots$$

$$u_{PID}(z) = u_0 + u_1 z^{-1} + u_2 z^{-2} + \dots .$$

Let $\quad \| F \| = \sum_{i=0}^{\infty} f_i , \qquad \| G \| = \sum_{i=0}^{\infty} g_i , \qquad \| E \| = \sum_{i=0}^{\infty} e_i ,$

$\| V \| = \sum_{i=0}^{\infty} v_i , \qquad \| U \| = \sum_{i=0}^{\infty} u_i$

For the BIBO stability of the closed-loop system, we must require that both

$\| F \|$ and $\| G \|$ be finite. So let $\| F \| = \alpha_1$ and $\| G \| = \alpha_2$,where α_1 and α_2 are constants to be determined.

To this end, we have

$$\| E \| \le \alpha_1 \| U \|, \tag{6.97}$$

$$\| V \| \le \alpha_2 \| U \|. \tag{6.98}$$

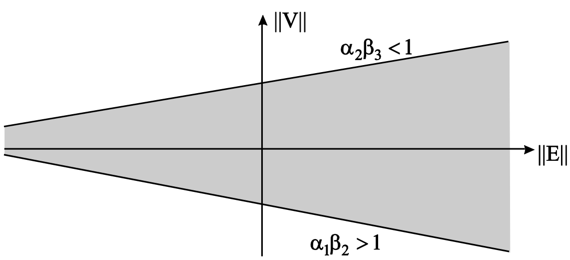


Figure 6.46 An open common region in the error space.

Suppose that

$$y(nT+T) = N(\, y(nT)\,) + H_1(\, u_{PID}(nT)\,) + H_2(\, w(nT)\,), \tag{6.99}$$

where $\{w(nT)\}$ are disturbances ($H_2 = 0$ if it does not exist), N is the closedloop system operator, and N, H_1, H_2 are bounded nonlinear functions (in operator norm):

$$\| N \| < \infty,\ \| H_1 \| < \infty,\ \| H_2 \| < \infty.$$

Then, it follows that

$$\| H_1(u_{PID}) \| = \| y(nT + T) - N\,(y(nT)) - H_2(w(nT)) \|$$

$$\le \| y(nT+T) \| + \| N \| \,.\, \| y(nT) \| + \| H_2 \| \,.\, \| w \|.$$

Since $\| H_1(u_{PID}) \| \le \| H_1 \| \,.\, \| u_{PID} \|$, to ensure the left-hand side be bounded,

we can require

$$\| H_1 \| \,.\, \| u_{PID} \| \le \| y(nT+T) \| + \| N \| \,.\, \| y(nT) \| + \| H_2 \| \,.\, \| w \|$$

or, even more conservatively,

$$\| u_{PID} \| \le \beta_1 + \beta_2 \| E \| + \beta_3 \| V \|, \tag{6.100}$$

where

$$\beta_1 \le \frac{1}{\| H_1 \|} (\| y(nT+T) \| + \| N \| \,.\, \| y(nT) \| + \| H_2 \| \,.\, \| w \|),$$

$\beta_2 \le \max\{ \| K_P \| , 1 \}$and $\beta_3 \le \max\{ \| K_I \| , \| K_D \| \}$.

Substituting (6.100) into (6.97) gives

$$\| E \| \le \alpha_1(\beta_1 + \beta_2 \| E \| + \beta_3 \| V \|),$$

which implies

$$\| V \| \ge \frac{1}{\alpha_1\beta_3} [\ (1- \alpha_1\beta_2) \| E \| - \alpha_1\beta_1\ ] \text{ if } \alpha_1\beta_2 < 1. \tag{6.101}$$

Similarly, substituting (6.100) into (6.98) leads to

$$\| V \| \le \frac{1}{1-\alpha_2\beta_3} [\alpha_2\beta_2 \| E \| + \alpha_2\beta_1\ ], \text{ if } \alpha_2\beta_3 < 1 \tag{6.102}$$

It is clear that the two straight lines on the right-hand side of (6.101) and (6.102) will create a common region for $\| V \|$ in the $\| E \|$ - $\| V \|$ plane. If $\alpha_1\beta_2 > 1$ (the first straight line has a negative slope) and $\alpha_2\beta_3 < 1$ (the second straight line has a positive slope), then an open region is created as shown in Figure 6.46. In this case, there is no common region in which the errors e and v are guaranteed to be bounded. Therefore, this case should be avoided.

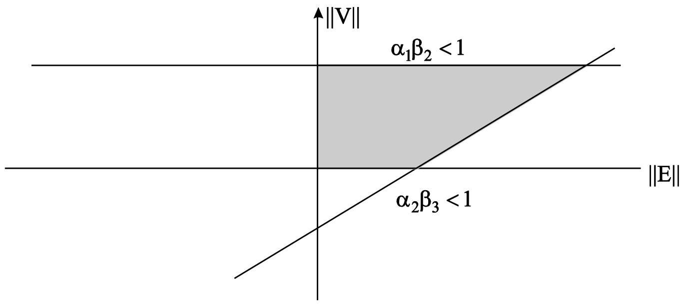


Figure 6.47 A closed common region in the error space.

Observe that we are dealing with norms, so only the first quadrant of the || E || - || V || plane is intersecting. To obtain a closed region on the plane, we can equate (6.101) and (6.102), which gives

$$\frac{1}{\alpha_1\beta_3}\left[(1-\alpha_1\beta_2)\,\|E_r\| - \alpha_1\beta_1\right] = \frac{1}{1-\alpha_2\beta_3}\left[\alpha_2\beta_2\,\|E_r\| + \alpha_2\beta_1\right], \qquad (6.103)$$

where $\|E_r\|$ is the value of || E || at a point denoted by r on the plane. Solving (6.103) for $\|E_r\|$, we have

$$\|E_r\| = \frac{\alpha_1\beta_1}{1-(\alpha_1\beta_2+\alpha_2\beta_3)} \qquad (6.104)$$

Substituting (6.104) into (6.102) yields the corresponding

$$\|V_r\| = \frac{\alpha_2\beta_1}{1-(\alpha_1\beta_2+\alpha_2\beta_3)} \qquad (6.105)$$

Therefore, the intersection point r of the two straight lines has the coordinates

$$(\|E_r\|,\|V_r\|) = \left(\frac{\alpha_1\beta_1}{1-(\alpha_1\beta_2+\alpha_2\beta_3)}.\frac{\alpha_2\beta_1}{1-(\alpha_1\beta_2+\alpha_2\beta_3)}\right), \qquad (6.106)$$

which are finite if $\alpha_1\beta_2 + \alpha_2\beta_3 < 1$. This situation is shown in Figure 6.47. We summarize the sufficient conditions in the following theorem:

Theorem6.5. A sufficient condition for the BIBO stability of the fuzzy PI+D control system is that the fuzzy control gains K_P, K_I, K_D are chosen such that there are constants α_1, α_2, β_1, β_2, β_3 that together satisfy

(a) $\|E\| \le \alpha_1 \|U\|$, $\|V\| \le \alpha_2 \|U\|$,

(b) $\beta 1 \leq \dfrac{1}{\| H_1 \|} [(1 + \| N \|) . \| y \| + \| H_2 \| . \| w \|]$,

(c) $\beta_2 \leq \max\{ \| K_P \| , 1\}$,

(d) $\beta_3 \leq \max\{ \| K_I \| , \| K_D \| \}$,

(e) $\alpha_1\beta_2 < 1$, $\alpha_2\beta_3 < 1$,

(f) $\alpha_1\beta_2 + \alpha_2\beta_3 < 1$,

and the closed region shown in Figure 6.47 is not empty. Here,

$$\| y \| = \sup_{n\geq 0} \| y\ nT \|$$

Remark: The norm $\| y \|$ in condition (b) of Theorem 6.5 need not be finite when verifying the constant β_1; namely, if $\| y \| = \infty$ then condition (b) is trivially satisfied. However, if all the other conditions are satisfied simultaneously, then $\| y \|$ would be finite as a result of the BIBO stability of the overall control system.

EXERCISE-6

1. Following the derivation of the digital PD controller shown in Figure 6.14, verify the digital PI controller shown in Figure 6.24.
2. Verify formulas (6.18) – (6.24) for the fuzzy PD controller by showing all the detailed derivation steps.
3. Verify formulas (6.29) – (6.37) for the fuzzy PI controller by showing all the detailed derivation steps.
4. Repeat all numerical simulations shown in Example 6.3, so as to experience how and how well the fuzzy PD controller works.
5. Perform some numerical simulations for the fuzzy PI controller and compare them with the conventional PI controller. You may use the same models as those simulated in Example 6.3.
6. Perform the graphical stability analysis studied in Section 6.8.5 on the fuzzy PD and fuzzy PI controller individually.

BIBLIOGRAPHY

1. C. R. Asfahl. Robots and Manufacturing Automation. New York, NY: Wiley, 1992.
2. G. Chen (1996). "Conventional and fuzzy PID controllers: an overview," Int. J. of Intelligent Control Systems, Vol. 1, pp 235-246.
3. G. Chen and H. Ying (1997)"BIBO stability of nonlinear fuzzy PI control systems," J. of Intelligent and Fuzzy Systems, Vol. 5, pp 245-256.
4. D. Drianker, H. Hellendoorn, and M. Reinfrank. An Introduction to Fuzzy Control. Berlin, Germany: Springer-Verlag,1993.
5. H. Malki, H. Li, and G. Chen(1994) "New design and stability analysis of fuzzy proportional-derivative control systems," IEEE Trans. on Fuzzy Systems, Vol. 2, pp 345-354.
6. H. Malki, D. Feigenspan, D. Misir, and G. Chen(1997). "Fuzzy PID control of a flexible-joint robot arm with uncertainties from time-varying loads," IEEE Trans. on Control Systems Technology, Vol. 5, pp 371-378.
7. D. Misir, H. Malki, and G. Chen. "Design and analysis of a fuzzy proportional-integral-derivative controller," Int. J. of Fuzzy Sets and Systems, Vol. 79, pp 297-314, 1996.
8. P. Sooraksa and G. Chen (1998). "Mathematical modeling and fuzzy control of a flexible-link robot arm," Mathematical and Computer Modeling, Vol. 27, pp 73-93.
9. K. Tanaka. "Design of model-based fuzzy control using Lyapunov's stability approach and its applications to trajectory stabilization of a model car," Theoretical Aspects of Fuzzy Control (ed. H. T. Nguyen, M. Sugeno, R. Tong, and R. R. Yager). pp 31-63, New York, NY: Wiley ,1995.
10. L. X. Wang. Adaptive Fuzzy Systems and Control. Englewood Cliffs, NJ: Prentice-Hall, 1994.
11. J. W. Webb and R. A. Reis. Programmable Logic Controllers: Principles and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1995.
12. L. A. Zadeh (1994). "The role of fuzzy logic in modeling, identification, and control," Modeling, Identification, and Control, Vol. 15, No. 3, pp 191-203
13. Guanrong Chen, Trung Tat Pham,"Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems, CRC Press, Boca Raton London, New York Washington, D.C. pp 183-237.

Chapter-7

Non-linear Fuzzy Control

7.1 INTRODUCTION

The analytic functions employed in models of linear and nonlinear system operate on the domain of crisp reals. In addition we have the class of fuzzy systems whose models are generally algebraic mappings from the domain of crisp reals into a prespecified domain of fuzzy reals. Similarly the class of controllers can be divided into linear, nonlinear and fuzzy knowledge base controllers (FKBC). Figure 7.1 shows an open scheme for different types of systems and controllers.

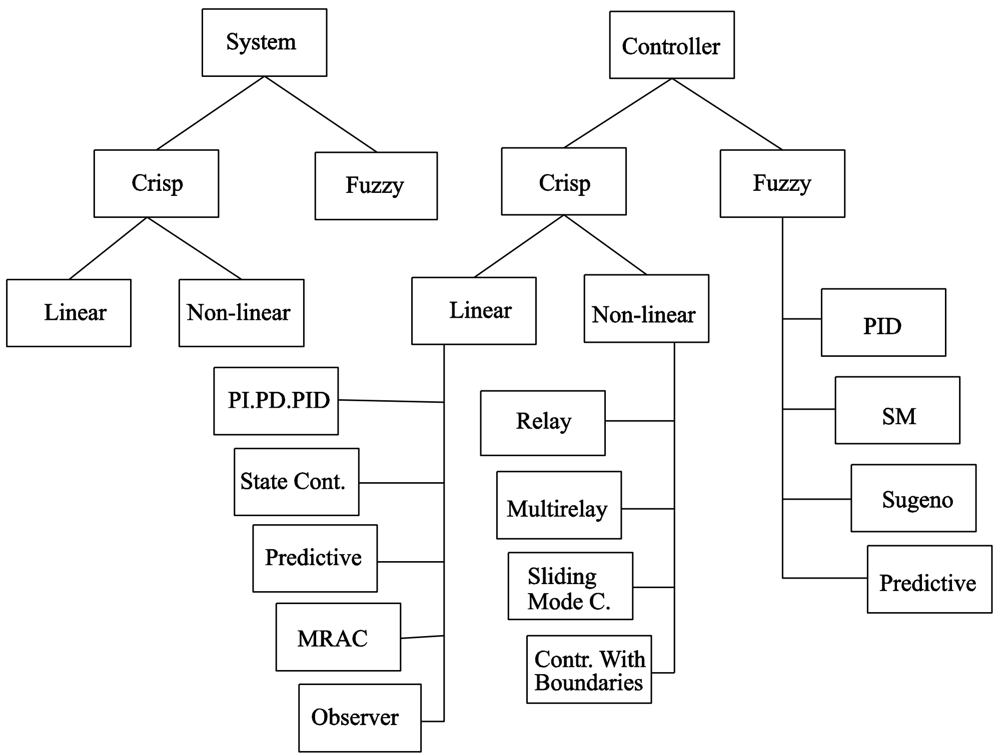


Fig. 7.1 open scheme for different types of systems and controllers.

7.2 | Fuzzy Logic Models and Fuzzy Control: An Introduction

Control systems theory, or what is called modern control systems theory today, can be traced back to the age of World War II, or even earlier, when the design, analysis, and synthesis of servomechanisms were essential in the manufacturing of electromechanical systems. The development of control systems theory has since gone through an evolutionary process, starting from some basic, simplistic, frequency-domain analysis for single-input single-output (SISO) linear control systems, and generalized to a mathematically sophisticated modern theory of multi-input multi-output (MIMO) linear or nonlinear systems described by differential and/or difference equations.

It is believed that the advances of space technology in the 1950s completely changed the spirit and orientation of the classical control systems theory: the challenges posed by the high accuracy and extreme complexity of the space systems, such as space vehicles and structures, stimulated and promoted the existing control theory very strongly, developing it to such a high mathematical level that can use many new concepts like state-space and optimal controls. The theory is still rapidly growing today; it employs many advanced mathematics such as differential geometry, operation theory, and functional analysis, and connects to many theoretical and applied sciences like artificial intelligence, computer science, and various types of engineering.

The modern control systems theory, referred to as conventional or classical control systems theory, has been extensively developed. The theory is now relatively complete for linear control systems, and has taken the lead in modern technology and industrial applications where control and automation are fundamental. The theory has its solid foundation built on contemporary mathematical sciences and electrical engineering, as was just mentioned. As a result, it can provide rigorous analysis and often perfect solutions when a system is defined in precise mathematical terms. In addition to these advances, adaptive and robust as well as nonlinear systems control theories have also seen very rapid development in the last two decades, which have significantly extended the potential power and applicable range of the linear control systems theory in practice.

Conventional mathematics and control theory exclude vagueness and contradictory conditions. As a consequence, conventional control systems theory does not attempt to study any formulation, analysis, and control of what has been called fuzzy systems, which may be vague, incomplete, linguistically described, or even inconsistent. Fuzzy set theory and fuzzy logic, studied in some detail in Chapters 1 and 2, play a central role in the investigation of controlling such systems. The main contribution of fuzzy control theory, a new alternative and branch of control systems theory that uses fuzzy logic, is its ability to handle many practical problems that cannot be adequately managed by conventional control techniques. At the same time, the results of fuzzy control theory are consistent with the existing classical ones when the system under control reduces from fuzzy to non-fuzzy. In other words, many well-known classical results can be extended in some natural way to the fuzzy setting. We have many examples in which the interval arithmetic is consistent with the classical arithmetic when an interval becomes a point.

The fuzzy logic is consistent with the classical logic when the multi-valued inference becomes two-valued andThe fuzzy stability and fuzzy controllability and observability become the classical ones when the fuzzy control systems become non-fuzzy. On the other hand, if there exists a good enough non-fuzzy (crisp) models of the process; we can adopt methods from linear and nonlinear control theory for the FKBC design.

Basically, the aim of fuzzy control systems theory is to extend the existing successful conventional control systems techniques and methods as much as possible, and to develop many new and special-purposed ones for a much larger class of complex, complicated and ill-modeled systems of fuzzy systems. This theory is developed for solving real-world problems. The fuzzy modeling techniques, fuzzy logic inference and decision-making, and fuzzy control methods are studied in this chapter. The real-world problems exist in the first place. Fuzzy logic, fuzzy set theory, fuzzy modeling, fuzzy control methods, etc. are all man-made and subjectively introduced to the scene. If this fuzzy interpretation is correct and if the fuzzy theory works, then we are able to solve the real-world problems. We also study this routine in detail for various control systems design and applications in this chapter.

7.2 THE CONTROL PROBLEM (CRISP LOGIC)

A programmable logic controller (PLC) is a simple microprocessor-based, specialized computer that carries out logical control functions of many types, with the general structure shown in Figure 7.2. A PLC usually operates on ordinary house current but may control circuits of large amperage and voltage of 440V and higher.

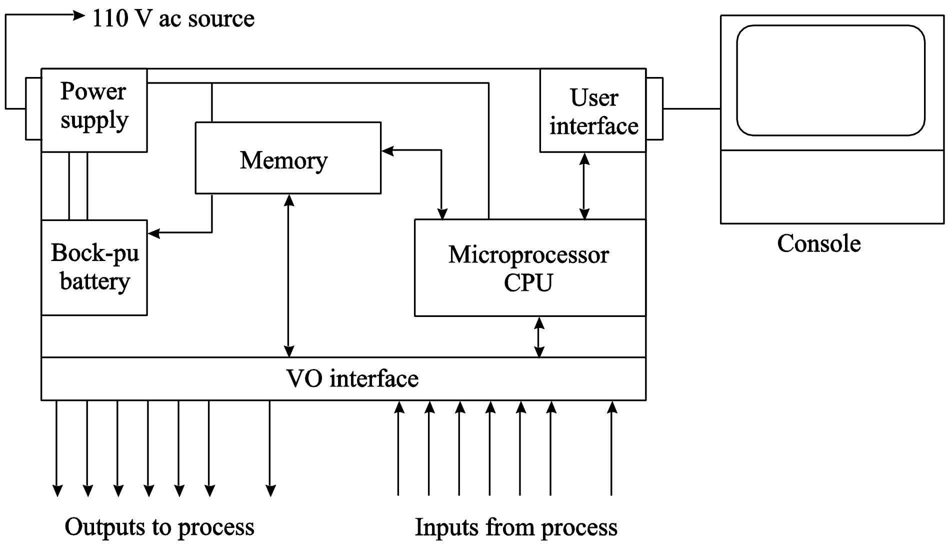


Fig. 7.2 A programmable logic controller (PLC)

A PLC typically has a detachable programmer module used to construct a new program or modify an existing one, which can be taken easily from one PLC to another. The program instructions are typed into the memory by the user via a keyboard. The central processing unit (CPU) is the heart of the PLC, which has three parts:

- (i) The processor,
- (ii) Memory and
- (iii) Power supply.

In the input-output (I/O) modules, the input module has terminals into which the user enters outside process electrical signals and the output module has another set of terminals that send action signals to the processor. A remote electronic system for connecting the I/O modules to long-distance locations can be added, so that the actual operating process under the PLC control can be some miles away from the CPU and its I/O modules. Optional units include racks, chassis, printer, program recorder/player, etc.

To see how a PLC works, let us consider an example of a simple pick-and place robot arm working in an automatic cycle under the control of a PLC. This simple robot arms is shown in Figure 7.3, which has three joints: one rotational type and two sliding types, as indicated by the arrows; and has a gripper that can close (to grasp an object) and open (to release the object).

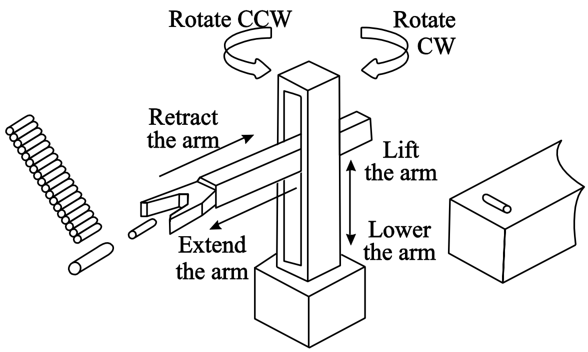


Fig. 7.3 A pick-and-place robot arm controlled by a PLC.

To control this robot arm, according to the structure shown in Figure 7.3, we need eight actions created by the PLC, called outputs of the PLC, as follows:

A_0: rotate the base counterclockwise (CCW),

A_1: rotate the base clockwise (CW),

A_2: lift the arm,

A_3: lower the arm,

A_4: extend the arm,

A_5: retract the arm,

A_6: close the gripper,

A_7: open the gripper.

Suppose that in each cycle of the control process, the entire job performance is completed in 12 steps sequentially within 40 seconds, as shown in Figure 7.4

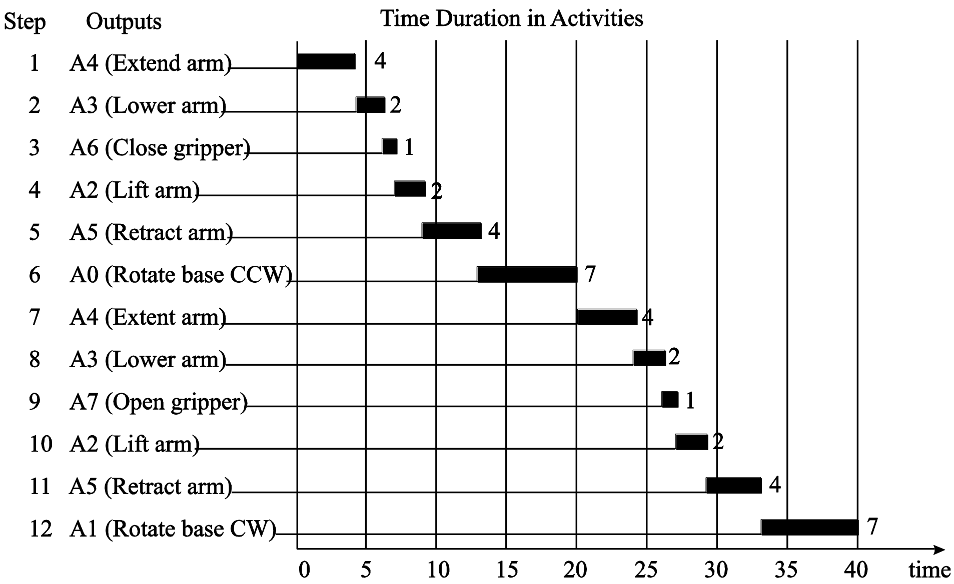


Fig.7.4 Time duration of activities of the robot arm.

In Figure 7.4, it is shown that the PLC is needed to first operate the "extend arm" action for 4 seconds (from the beginning), and then stop this action. To this end, the PLC operates the "lower arm" action for 2 seconds and then stops, and so on. For simplicity, we assume that the transitions between any two consecutive steps are continuous and smooth, so that no time is needed for any of these transitions.

Table 7.1 PLC Drum-Timer Array for the Robot Arm

Step	Counts (Count/sec	Outputs							
		A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7
1	4	0	0	0	0	1	0	0	0
2	2	0	0	0	1	0	0	0	0
3	1	0	0	0	0	0	0	1	0
4	2	0	0	0	0	0	0	0	0
5	4	0	0	0	0	0	1	0	0
6	7	1	0	0	0	0	0	0	0
7	4	0	0	0	0	1	0	0	0
8	2	0	0	0	1	0	0	0	0
9	1	0	0	0	0	0	0	0	1
10	2	0	0	1	0	0	0	0	0
11	4	0	0	0	0	0	1	0	0
12	7	0	1	0	0	0	0	0	0

To set up the PLC timer, called the drum-timer, so as to accomplish these sequential activities of the robot arm, we need to select the time frequency for the PLC. Since the smallest step time in Figure 7.4 is the 1-second gripper operation, we select a count frequency of 1 count per second. The entire PLC drum-timer array is shown in Table 7.1, where 1 means ON and 0 means OFF.

The above-described sequential control process is by nature an open-loop control procedure. Now, if we use sensors to provide feedback information at each step of the arm's motion, then we can also perform a closed-loop logic control.

For this purpose, mechanical limit switches (or proximity switches) can be used to sense the point of completion of each arm motion and, hence, provide inputs to the PLC. The PLC can then be programmed not to begin the next step until the previous step has been completed, regardless of how much time this might take.

We use, by industrial convention, the number symbol 00 to represent the input from the limit switch that signals completion of the previous output A0 ("rotating the base counterclockwise"), 01 to represent the input from the limit switch that signals completion of the previous output A_1 ("rotating the base clockwise") and 07 for the input that signals completion of A_7. The entire scheme, together with necessary prior inputs for each arm motion, is shown in Table 7.2. It may be noted that in this diagram usually two previous steps are sufficient as prerequisite, but one more is needed in case two are not enough for distinction of different actions.

It is also important to remark that if we do not consider the necessary prior conditions, then when the PLC applies the logic control to the robot arm, it may make mistakes. For example, the second step in Table 7.2 is "lower arm," for which the completion is represented by 03. If 03 were to be used as the logic input to trigger the next step in the sequential operations, the "close gripper" action, the robot arm would always close its gripper after it executes the action "lower arm." However, the eighth step of the program is also "lower arm," but it is followed by "open gripper" instead. Hence, it is necessary to provide other prior conditions to the eighth step to ensure that it would not repeat the second step here.

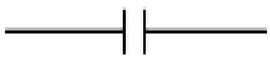
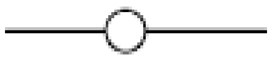
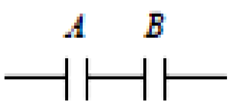
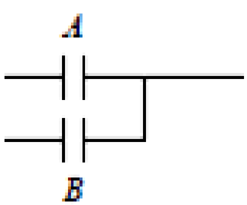
Table 7.2 Robot Arm Control Scheme

Arm Motion	PLC Outputs	PLC Inputs	Necessary Prior Conditions
extend arm	A_4	04	01,07
lower arm	A_3	03	01,04,07
close gripper	A_6	06	01,03
lift arm	A_2	02	01,06
retract arm	A_5	05	01,02,06
rotate base CCW	A_0	00	05,06
extend arm	A_4	04	00,06
lower arm	A_3	03	00,04,06
open gripper	A_7	07	00,03
lift arm	A_2	02	00,07
retract arm	A_5	05	00,02,07
rotate base CW	A_1	01	05,07

Figure 7.5 shows the ladder logic diagram corresponding to Table 7.2 in which the standard symbols listed in Table 7.3 are used and the input contact P0 is used to represent a power-ON switch. Here, the input contacts include switches, relays, photoelectric sensors, limit switches, and other ON-OFF signals from the logical system; the output loads include motors, valves, alarms, bells, lights, actuators, or other electrical loads to be driven by the logical system. We should not confuse the input contact symbol with the familiar circuit symbol for capacitors. Also, we should note that output loads can become input contacts in the following steps, depending on the actual system structure.

In a routine yet somewhat tedious procedure, we can verify that the PLC ladder logic diagram shown in Figure 7.5 performs the automatic control process summarized in Table 7.2. Under the indicated necessary prior conditions for each step during the entire process, this PLC works for the designed “pick-and-place” motion-control of the robot arm described by Figure 7.4.

Table 7.3 Lader diagram symbol

	Input contacts	
	Output loads	
	Logical NOT	$\overline{A}$
	Logical AND	$A \wedge B$
	Logical OR	$A \vee B$

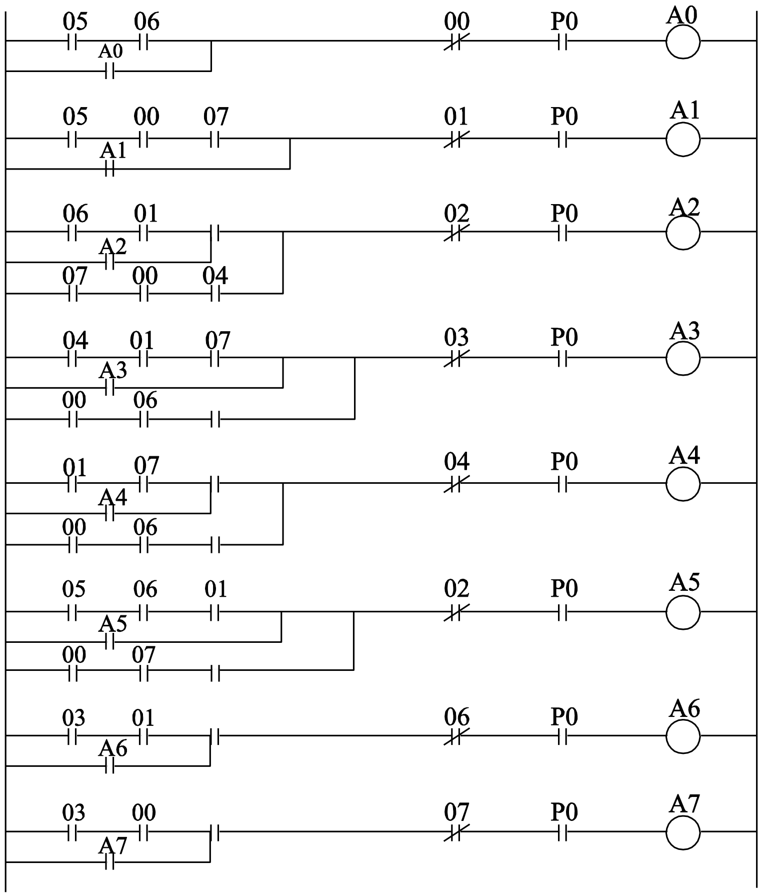


Fig.7.5. PLC ladder logic diagram for the robot.

The design procedure is time-consuming. However, once a design has been completed, the PLC can be automatically programmed over and over again, to control the robot arm to repeat the same "pick-and-place" motion for the production line. When the job assignment is changed, the PLC has the flexibility to be reprogrammed to perform a new logic control process.

7.3 THE CONTROL PROBLEM (FUZZY LOGIC) MODEL-FREE APPROACH

The programmable logic controllers (PLCs) discussed in the section7.2 have been widely used in industries and can only perform classical two valued logic in programming. Some are relatively simple but very precise automatic control processes. To carry out fuzzy logic-based control programming, a new type of programmable fuzzy logic controllers is needed.

The majority of fuzzy logic control systems are knowledge-based systems in that either their fuzzy models or their fuzzy logic controllers are described by fuzzy IF-THEN rules, which have to be established based on experts' knowledge about the systems, controllers, performance, etc. Moreover, the introduction of input-output intervals and membership functions is more or less subjective, depending on the designer's experience and the available information. However, we emphasize once again that after the determination of the fuzzy sets, all mathematics to follow are rigorous. Also, the purpose of designing and applying fuzzy logic control systems is, above all, to tackle those vague, ill-described, and complex plants and processes that can hardly be handled by classical systems theory, classical control techniques, and classical two-valued logic. There are two types of fuzzy logic control system:

(i) The fuzzy logic controller directly performs the control actions and thus completely replaces a conventional control algorithm.

(ii) The fuzzy logic controller is involved in a conventional control system and thus becomes part of the mixed control algorithm, so far as to enhance or improve the performance of the overall control system.

In this section, we first discuss a general approach of fuzzy logic control for a conventional (crisp) system (plant or process) of the feedback (closed loop) type.

7.3.1 .A Closed-Loop Set-Point Tracking System

To facilitate our presentation and discussion, we consider the typical continuous-time, closed-loop, set-point tracking system shown in Figure 7.6. In this figure, we assume that the plant is a conventional (crisp) one, which is given but its mathematical model may not be known, and that all the signals (r, e, and y) are crisp. The closed-loop set-point tracking control problem is to design the controller such that the output signal of the controlled plant, y, can track the given reference signal r (need not be a constant):

$$e(t) := r(t) - y(t) \to 0 \quad (t \to \infty).$$

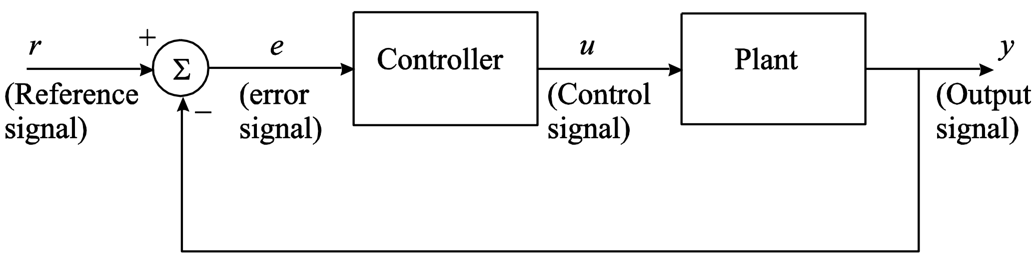


Fig. 7.6 General structure of a fuzzy logic controller

Instead of designing a conventional controller, we study how to design a fuzzy logic controller for the same purpose. Recall that in designing a conventional controller, a precise mathematical model (formulation) of the plant is usually necessary. A typical example is the conventional proportional-integral-derivative (PID) controller design, where the first- or second-order linear plant transfer function has to be first given. We review and discuss these conventional PID controllers in Chapter 6. Here, to design a fuzzy logic controller, not necessarily of PID-type, for set point tracking, we suppose that the mathematical formulation of the plant is completely unknown.

If the mathematical formula of the plant is unknown, for instance we don't even know if it is linear or nonlinear (and, if it is linear, we don't know what order it has; if it is nonlinear, we don't know what kind of nonlinearity it has), how can we design a controller to perform the required set-point tracking? One may think of designing a conventional controller and try to answer this question at this point, to appreciate the ease of fuzzy logic controller design to be studied below.

Before we discuss how fuzzy logic can help in completing such a design for a "black box" system, we need to clarify some basic concepts and to introduce some new terminology.

First, the general structure of a fuzzy logic controller (FLC), or fuzzy controller (FC) consists of three basic portions:The fuzzification unit at the input terminal,

the inference engine built on the fuzzy logic control rule base in the core, and

The defuzzification unit at the output terminal (refer to Figure 7.7).

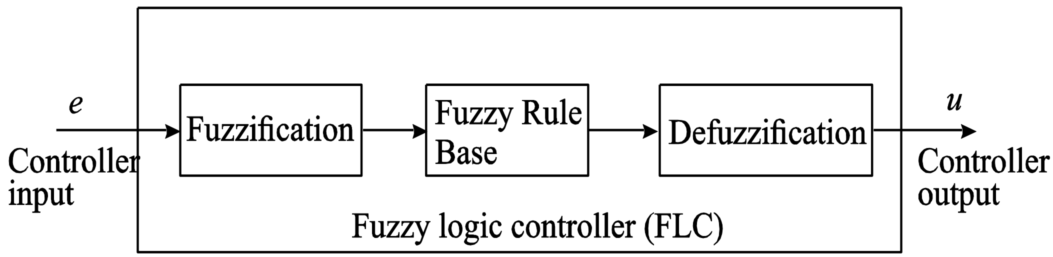


Fig. 7.7 General structure of a fuzzy logic controller

The fuzzification module transforms the physical values of the current process signal, the error signal in Figure 7.6 which is input to the fuzzy logic controller, into a normalized fuzzy subset consisting of a subset (interval) for the range of the input values and an associate membership function describing the degrees of the confidence of the input belonging to this range. The purpose of this fuzzification step is to make the input physical signal compatible with the fuzzy control rule base in the core of the controller. Here, between the physical input signal and the fuzzy subset within the fuzzification unit, a pre-processing unit mapping the physical signal to some pointwise and crisp real values (that the fuzzy subset can accept) depending on the nature of the underlying process may be needed. Generally, a universal fuzzy logic controller for the closed-loop set-point tracking system shown in Figure 7.6 is unlikely possible.

Hence, the fuzzy subset, both the subset and the membership function, has to be selected by the designer according to the particular application at hand. In other words, depending on the nature and characteristics of the given plant and reference signal the FLC has to be designed to fit to the need to make the closed-loop fuzzy control system work for that particular application. This situation is just like the design of a conventional controller for a specifically given system, where there is no universal controller in practical design.

The role of the inference engine in the FLC is key to make the controller work and work effectively. The job of the "engine" is to create the control actions, in fuzzy terms, according to the information provided by the fuzzification module and the set-point tracking requirement (and perhaps other control performance requirements as well). A typical fuzzy logic IF-THEN rule base performing the inference is in the general form that was studied in detail in Sections 3 of Chapter 3. More specifically, the rule base is a set of IF-THEN rules of the form

R^1: IF controller input e_1 is E_{11} AND ... AND controller input e_n is E_{1n}

THEN controller output u_1 is U_1.

R^m: IF controller input e_1 is E_{m1} AND ... AND controller input e_n is E_{mn}

THEN controller output u_m is U_m.

Here, as discussed in Sections 3.2 – 3.4 of Chapter 3, the fuzzy subsets E_{11}, ...,E_{m1} share the same subset E_1 and the same membership function μ_{E_1} defined on E_1, and fuzzy subsets E_{1n}, …., E_{mn} share the same subset E_n and the same membership function μ_{E_n} defined on E_n. In general, m rules produce m controller outputs, u_1, ..., u_m, belonging to m fuzzy subsets, U_1, ..., U_m, in which, of course, some of them may overlap. The establishment of this rule base depends heavily on the designer's work experience, knowledge about the physical plant, analysis and design skills, etc., and is, hence, more or less subjective. Thus, a good design can make the controller work; a better design can make it work more effectively. This situation is just like conventional design: any specific design is not unique in general. Yet, there are some general criteria and some routine steps for the designer to follow in a real design, which will be discussed in more detail later. Here, basically, what have to be determined are the choices of the controller's input and output variables and the IF-THEN rules.

The defuzzification module is the connection between the control rule base and the physical plant to be controlled that plays the role of a transformer mapping the controller outputs (generated by the control rule base in fuzzy terms) back to the crisp values that the plant can accept. Hence, in a sense the defuzzification module is the inverse of the fuzzification module. The controller outputs u_1, ..., u_m generated by the rule base above are fuzzy signals belonging to the fuzzy subsets U_1, ..., U_m respectively. The job of the defuzzification module is to convert these fuzzy controller outputs to a pointwise and crisp real signal, u, and then send it to the physical plant as a control action for tracking. Between the defuzzification step and the physical plant, a post-processing unit mapping the pointwise signal u to a physical signal (that the plant can accept) may be needed, depending again on the nature of the underlying process. What has to be determined in this stage is essentially a defuzzification formula. There are several commonly used, logically meaningful, and practically effective defuzzification formulas available, which are by nature weighted average formulas in various forms.

This tree-step design routine, the "fuzzification – rule base establishment – defuzzification" procedure, is further discussed in next section.

7.3.2. Design Principle of Fuzzy Logic Controllers

Now, we discuss some design principles of a fuzzy logic controller using the typical structure shown in Figure 7.7 for the setpoint tracking system depicted in Figure 7.6.

We only consider a continuous-time single-input/single-output (SISO) system in this study for simplicity of notation. Discrete-time and multi-input/ multi-output (MIMO) cases can be similarly discussed.

Suppose that a scalar-valued reference signal r, the set-point, is given, which needs not be a constant in general. For ease of explanation, we fix it to be constant in the following discussion. A plant is also assumed to be given, whose mathematical formulation is assumed to be unavailable. The aim is to design a fuzzy logic controller, to be put into the controller block of Figure 7.6, to derive the plant output y(t) to track the constant set-point r as $t \to \infty$ or in other words, to force the error signal

$$e(t) = r - y(t) \to 0 \qquad (t \to \infty)$$

This is a general approach for the design of the fuzzy logic controller.

7.3.3 The Fuzzification Module.

The fuzzification module performs the following functions:

(a) It transforms the physical values (position, voltage, degree, etc.) of the process signal, the error signal shown in Figure 7.7 which is an input to the fuzzy logic controller, into a normalized fuzzy subset consisting of a subset (interval) for the range of the input values and a normalized membership function describing the degree of confidence of the input belonging to this range.

For example, suppose that the physical error signal, the input to the fuzzy logic controller shown in Figure 7.7, is the degree of temperature. Suppose also that the distribution of this temperature error signal is within the range [–25°,45°] in an application in which the fuzzy logic controller needs to be designed to reduce this error to 0° ± 0.01°. In this case, the scale of 1° is too large to use in the measurement of the control effect within ±0.01°. Hence, we may first try to rescale the range to be [−2500, 4500] in the unit of 0.01°. But this is not convenient in the calculation of fuzzy membership functions either. To have a compromise, a better choice can be a combination of two different scales (see Figure 7.8): Use [−25°, 45°] when |error| > 1 and

Use [−100*, 100*] when |error| ≤ 1, with 1° = 100*.

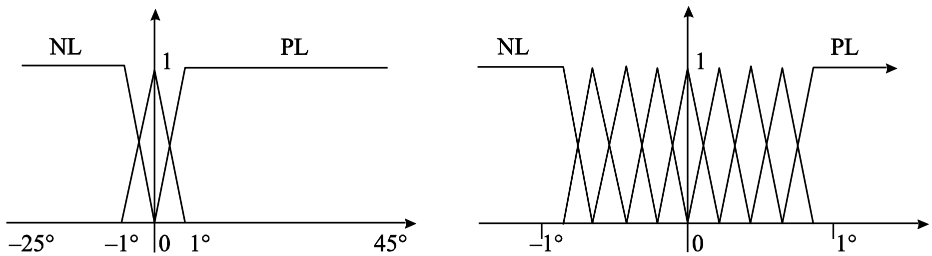


Fig.7.8 Selection example of membership function

(b) It selects reasonable and good, ideally optimal, membership functions under certain convenient criteria meaningful to the application

In the above-described temperature control example, one may use the membership functions shown in Figure 7.8 to describe "the error is positive large (PL)" or "the error is negative large (NL)", etc., in which one figure is in the 1° scale and the other is in the 1* = 0.01° scale. A designer may have other choices, of course, but this example should provide some idea about how the selection can be done in general.

In the fuzzification module, the input is a crisp physical signal (e.g., temperature) of the real process and the output is a fuzzy subset consisting of intervals and membership functions. This output will be the input to the next module, the fuzzy logic IF-THEN rule base for the control, which requires fuzzy-subset inputs in order to be compatible with the fuzzy logic rules.

7.3.4 The Fuzzy Logic Rule Base.

Designing a good fuzzy logic rule base is key to obtaining a satisfactory controller for a particular application. Classical analysis and control strategies should be incorporated in the establishment of a rule base. A general procedure in designing a fuzzy logic rule base includes the following:

(i) To determine the process states and control variables.

In the set-point tracking example discussed above in Figures 7.6 and 7.7, let us suppose that the physical process is a temperature control. Thus, the set-point r is a target temperature to be reached, say r = 45°. In this application, the process state is the overall controlled system output, y(t), which is also temperature. Finally, the error signal

$$e(t) = r - y(t), \qquad \ldots(7.1)$$

as shown in Figure 7.9, is used to create the control variable u (see Figure 7.6) through the controller.

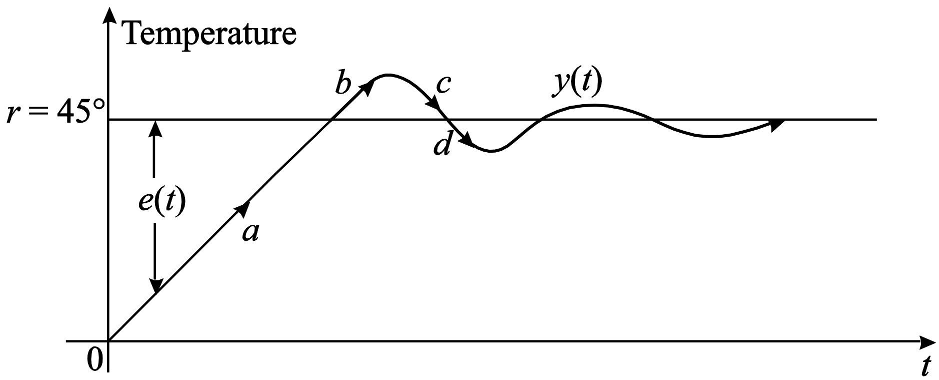


Fig. 7.9 Temperature set point tracking example

(ii) To determine input variables to the controller.

As mentioned above, the tracking error signal e(t) is an input variable to the controller (see Figures 7.6 and 7.7). Generally, we need more auxiliary input variables in order to establish a complete and effective rule base. In this temperature control example, it can be easily seen that only the error signal e(t) is not enough to write an IF-THEN control rule. Indeed, let us say $e > 0$ at a moment. Then we know $e = r - y > 0$ or $r > y$, i.e., at that moment the system output y is below the set-point. This indicates that the output y is either at position a or position d in Figure 7.9. However, this information is not sufficient for determining a control strategy that can bring the trajectory of y to approach the set-point there are two cases:

Case1. If the output y is at position a then the controller should take action to keep the trajectory going up

Case2, If y is at position d then the controller should turn the trajectory to the opposite moving direction (from pointing down to pointing up).

In the first case the controller should maintain the previous action but in the second case it should switch its previous action to the opposite. Therefore, one more input variable that can distinguish these two situations is necessary.

It may be recalled from Calculus that if a curve is moving up its derivative is positive and if it is moving down its derivative is negative. Hence, the change of the error signal, denoted $\dot{e}$ or Δe can help distinguish the two situations at points a and d as well as that at points b and c as shown in Figure 7.9. Hence, we introduce the change of error as the second input variable for this temperature set-point tracking example. More input variables can be introduced such as the second derivative $\ddot{e}$ or the sum (integral) Σe of the errors, to make the controller work more effectively. However, this will also complicate the design, and so there is a trade-off for the design engineer. In order to simplify the design and for simplicity of discussion, in this example we only use two input variables e and $\dot{e}$ for the controller.

(iii) To establish a fuzzy logic IF-THEN rule base

The input variables determined in the previous step will be used in the design of the rule base for control. As mentioned above, successful classical analysis and control strategies should also be incorporated in the design. To obtain some ideas and insights about the rule base design, we consider again the above temperature control example.

In Figure 7.9, it is clear that essentially we have four situations to consider: when the temperature output y(t) is at the situations represented by the points a, b, c, and d. We thus need at least four rules for this set-point tracking application in order to make it work and work effectively.

In principle, also intuitively, we can set up the following four control rules, where u is the output (control action) of the controller:

R^1 : IF $e > 0$ AND $\dot{e} < 0$ THEN $u(t+) = u(t)$;

R^2 : IF $e < 0$ AND $\dot{e} < 0$ THEN $u(t+) = -u(t)$;

R^3 : IF $e < 0$ AND $\dot{e} > 0$ THEN $u(t+) = u(t)$;

R^4 : IF $e > 0$ AND $\dot{e} > 0$ THEN $u(t+) = -u(t)$.

Otherwise (e.g., $e = 0$ or $\dot{e} = 0$), $u(t+) = u(t)$, until the next step.

In Figure 4.8, it is very important to observe that

$$\dot{e}(t) = \dot{r} - \dot{y}(t) = -\dot{y}(t). \qquad \ldots(7.2)$$

Hence, the rules R^1, R^2, R^3 and R^4 correspond to the situations indicated by points a, b, c, and d respectively in Figure 7.9. In these rules, the notation $u(t+) = u(t)$ means that, conceptually, the controller retains its previous action unchanged, $u(t+) = -u(t)$ means that the controller turns its previous action to the opposite (e.g., from a positive action to a negative action, or vice versa). Quantitative implementation of these actions will be further discussed below.

To this end, it can be easily realized that by following the above four rules, the controller is able to drive the temperature output y(t) to track the set-point r, at least in principle. For example, if $e > 0$ and $\dot{e} < 0$ then $r > y$ and $\dot{y} > 0$, which means that the curve y is at position a of Figure 7.9. In this case, rule R^1 implies that the controller should maintain its current action (to keep pushing the system in the same way). The other three rules can be similarly analyzed.

For technical issues let us again look at the situation at point a in Figure 7.9, which has the control rule R^1. One problem with this rule is that if the previous control action u was small, this rule would let the controller keep driving the output temperature up toward the set-point, but very slowly; so it may need a long time to complete the task. Another problem with this rule is that if the previous control action u was large, this rule would lead the controller to drive the output temperature overshooting the set-point. As a result of such rules and actions, the output temperature oscillates up and down around the set-point, which may not eventually settle at the set-point, or take a very long time to do so. Hence, these four rules have to be further improved.

It may be recalled that we have fuzzified the error signal e in the fuzzification module. If we also fuzzify the change of error, $\dot{e}$, in the fuzzification module, then we have two fuzzified input variables, e and $\dot{e}$, for the controller, and they have corresponding membership functions to describe their properties of "positive large," "negative small", etc. These properties can be used to improve the above rules. Again, let us consider the situation at point a in Figure 4.8 and its corresponding control rule R^1. It is clear that if the error $e > 0$ is small then the controller can take a smaller action, and if the error $e > 0$ is large then the controller can take a larger action. In doing so, the output trajectory y(t) will be directed to the set-point correctly and more efficiently. Therefore, we should incorporate the membership values of the two input variables e and $\dot{e}$ in the four control rules.

To further simplify the notation and discussion below, we only use the simple membership functions shown in Figure 7.10 rather than those shown in Figure 7.8 for both e and $\dot{e}$. In a real application, of course, a designer can (and likely should) employ more membership functions.

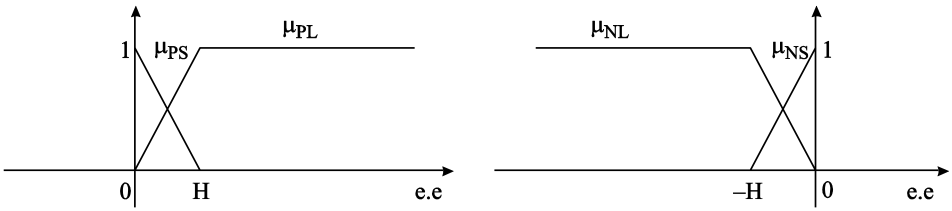


Fig. 7.10 Four membership function for both e and $\dot{e}$

Using these membership functions as weights for the control u(t), we can accomplish the following task: if y is far away from r then the control action is large, but if y is close to r then the control action is small. The improved control rule base is obtained as follows:

R^1 : IF e = PL AND $\dot{e} < 0$ THEN $u(t+) = \mu_{PL}(e) \,.\, u(t)$;

R^2 : IF e = PS AND $\dot{e} < 0$ THEN $u(t+) = (1 - \mu_{PS}(e)) \,.\, u(t)$;

R^3 : IF e = NL AND $\dot{e} < 0$ THEN $u(t+) = -\mu_{NL}(e) \,.\, u(t)$;

R^4 : IF e = NS AND $\dot{e} < 0$ THEN $u(t+) = -(1 - \mu_{NS}(e)) \,.\, u(t)$;

R^5 : IF e = NL AND $\dot{e} > 0$ THEN $u(t+) = \mu_{NL}(e) \,.\, u(t)$;

R^6 : IF e = NS AND $\dot{e} > 0$ THEN $u(t+) = (1 - \mu_{NS}(e)) \,.\, u(t)$;

R^7 : IF e = PL AND $\dot{e} > 0$ THEN $u(t+) = -\mu_{PL}(e) \,.\, u(t)$;

R^8 : IF e = PS AND $\dot{e} > 0$ THEN $u(t + 1) = -(1 - \mu_{PS}(e)) \,.\, u(t)$,

Here and below "= PL" means "is PL", etc.

These rules can be easily verified by taking into account the temperature output curve y(t) shown in Figure 7.9 the relation $\dot{y}(t) = -\dot{e}(t)$, and the four membership functions for e (and the same four membership functions for $\dot{e}$) as shown in Figure 7.10. Note that "**.**" in the formulas of u(t+1) above are algebraic multiplications.

To implement these rules on a digital computer, we actually use their discrete-time version of the form

$$u(t) = u(kT) \text{ and } u(t+) = u((k+1)T), \qquad \ldots(7.3)$$

where T is the sampling time and u(kT) is the value of the new change of the control action and $\Delta u(t)$ at t = kT, k = 0, 1, 2, ... Thus, the actual executive rules on a computer program become

R^1: IF $e(kT) = PL$ AND $\dot{e}(kT) < 0$

THEN $u((k+1)T) = \mu_{PL}(e(kT)) \,.\, u(kT)$;

R^2: IF $e(kT) = PS$ AND $\dot{e}(kT) < 0$

THEN $u((k+1)T) = (1 - \mu_{PS}(e(kT))) \,.\, u(kT)$;

R^3: IF $e(kT) = NL$ AND $\dot{e}(kT) < 0$

THEN $u((k+1)T) = -\mu_{NL}(e(kT)) \,.\, u(kT)$;

R^4: IF $e(kT) = NS$ AND $\dot{e}(kT) < 0$

THEN $u((k+1)T) = -(1 - \mu_{NS}(e(kT))) \,.\, u(kT)$;

R^5: IF $e(kT) = NL$ AND $\dot{e}(kT) > 0$

THEN $u((k+1)T) = \mu_{NL}(e(kT)) \,.\, u(kT)$;

R^6: IF $e(kT) = NS$ AND $\dot{e}(kT) > 0$

THEN $u((k+1)T) = (1-\mu_{NS}(e(kT))) \,.\, u(kT)$;

R^7: IF $e(kT) = PL$ AND $\dot{e}(kT) > 0$

THEN $u((k+1)T) = -\mu_{PL}(e(kT)) \,.\, u(kT)$;

for all k = 0, 1, 2, ..., where $\dot{e}(kT) \approx \frac{1}{T}[e(kT) - e((k-1)T)]$with the initial conditions y(0) = 0, $e(-T) = e(0) = r - y(0)$, $\dot{e}(0) = \frac{1}{T}[e(0) - e(-T)] = 0$.

We finally remark that another commonly used alternative for determining the weight of the control u(t) is to give it a value of PL, PS, NL or NS. More specifically, we put the above rule base in a tabular form as shown in Table 7.4.

Table 7.4 is sometimes called a "look-up table," which can be made more accurate by dividing both e and e& into more sub cases. Table 7.5 gives an example of a more subtle rule-base table for fuzzy logic controller whose control action is proportional to both e and $\dot{e}$, namely,

$$u((k+1)T) = a\, e(kT) + b\, \dot{e}\, (kT)$$

Table 7.4 a Rule Base in Tabular Form for Δu

e \ $\dot{e}$	< 0	> 0
PL	PL	NL
PS	PS	NS
NS	NS	PS
NL	NL	PL

Table 7.5 a Rule Base in Tabular Form for $u = ae + b\dot{e}$

e \ $\dot{e}$	NL	NM	NS	ZO	PS	PM	PL
NL	NL	NL	NL	NL	NM	ZO	PS
NM	NL	NL	NL	NM	ZO	PS	PM
NS	NL	NL	NM	ZO	PS	PM	PL
ZO	NL	NM	ZO	PS	PM	PL	PL
PS	NM	ZO	PS	PM	PL	PL	PL
PM	ZO	PS	PM	PL	PL	PL	PL
PL	PS	PM	PL	PL	PL	PL	PL

for some constants a > 0 and b > 0, where NM and ZO mean "negative medium" and "zero," respectively, and the other abbreviations are similarly understood.

Finally, we need to select membership functions for the different control outputs, given either by the formulas shown in the rule base R^1 - R^8 above, or by the linguistic terms PL, PS, NS and NL shown in Table 7.4. Figure 7.11 is a simple, yet typical choice for the membership functions of u, where P, N, and ZO indicate positive, negative, and zero, respectively, and H is a real number.

(iv) To establish a fuzzy logic inference engine

In order to complete the fuzzy logic inference embedded in the control rule base, the general fuzzy IF-THEN rule given in Chapter 3 Table 3.3 has to be applied to each rule in the rule base.

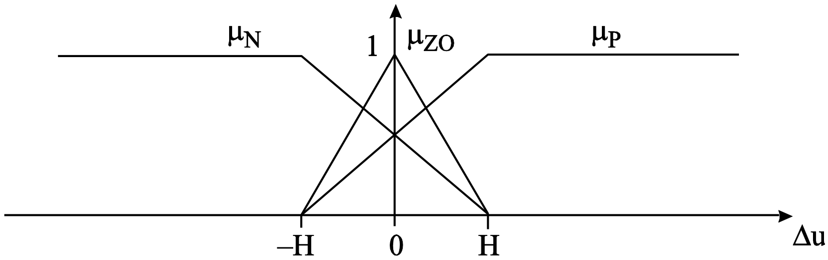


Fig. 7.11 Typical membership function for Δu

Take the first rule as an example:

R^1 : IF e(kT) = PL AND $\dot{e}$(kT) < 0THEN u((k+1)T) = μ_{PL}(e(kT)) **.** u(kT).

In this rule, e(kT) = PL has a membership function μ_{PL}(e(kT)) shown in Figure 7.10 (a),

$\dot{e}$ (kT) < 0 is non-fuzzy and so has membership values $\mu_N(\dot{e}(kT)) = 1$ and

u((k+1)T) has three membership functions as shown in Figure 7.11. Thus, the fuzzy logic inference

$\mu_{PL}(e(kT)) \wedge \mu_N(\dot{e}\ (kT)) \Rightarrow (u((k+1)T))$

yields the logical inference membership function (see Chapter 3 for more detail):

$\mu(u((k+1)T)) = \min\{\mu_{PL}(e(kT)),\ \mu_N(\dot{e}(kT))\} = \mu_{PL}(e(kT))$

without using the membership functions shown in Figure 7.11.

Another approach is to use the output membership functions shown in Figure 4.10, along with the weighted average formula given by 3.1.3 as shown in formula (7.4) below:

$$u((k+1)T) = \frac{\sum_{i=1}^{N} \mu_{U_i}(u_i(kT)).u_i(kT)}{\sum_{i=1}^{N} \mu_{U_i}(u_i(kT))} \qquad \ldots(7.4)$$

Once this logical inference has a fuzzy logic membership function, it is completed as a logical formula. A fuzzy logic control rule base with all complete logical inference membership functions is called a fuzzy logic inference engine: it activates the rules with fuzzy logic.

Remarks:The logical inference membership functions are usually not used in the design of a digital fuzzy logic controller in many control engineering applications today, where only the control rule base discussed in the last part (iv) is essential.We follow this common practice to use the rule base rather than the inference engine in the rest of the chapter. This is a design of a controller for which the design engineer has a choice of what approach to adopt. It is just like a classical controller design for which the designer usually has different options.

7.3.5 The Defuzzification Module.

The defuzzification module is in a sense the reverse of the fuzzification module and it converts all the fuzzy terms created by the rule base of the controller to crisp terms (numerical values) and then sends them to the physical system (plant, process), so as to execute the control of the system.

The defuzzification module performs the following functions:

(a) It creates a crisp, overall control signal u by combining all possible control outputs from the rule base into a weighted average formula such as

$$u((k+1)T) = \frac{\sum_{i=1}^{N} \alpha_i u_i(kT)}{\sum_{i=1}^{N} \alpha_i}, \text{ where } (\alpha_i \geq 0, \sum_{i=1}^{N} \alpha_i > 0) . \qquad \ldots(7.5)$$

In the control rule base R^1 - R^8, established above, we obtained eight different control outputs u denoted by $u_i(kT)$, i = 1, ..., 8.

Suppose that we have a fuzzy logic control rule base

R^i : IF $e(kT)$ is E_i AND $\dot{e}(kT)$ is F_i THEN $u_i((k+1)T)$ is U_i,

where i = 1, ..., N; E_i, F_i, and U_i are fuzzy subsets consisting of some bounded intervals with their associated membership functions $\mu_{Ei}(.)$, $\mu_{Fi}(.)$ and $\mu_{Ui}(.)$ respectively. Note that this control rule base may also be given in a tabular form like the one shown in Table 7.5.

There are several commonly used defuzzification formulas:

(i) The "center-of-gravity" formula:

,

It is by nature the same as the formula (3.1 – (ii) introduced in Chapter 3. The continuous-time version of this formula is

$$u(t) = \frac{\int_U \mu_U(\tilde{u}).\tilde{u}\, d\tilde{u}}{\int_U \mu_U(\tilde{u}) d\tilde{u}}, \qquad \ldots(7.6)$$

where U is the value range (interval) of the control $\tilde{u}$ in the rule base (the continuous-time version):

R : IF $e(t)$ is E AND $\dot{e}(t)$ is F THEN $u(t+)$ is U.

In applications, namely, in a practical engineering design employing digital computers, the continuous-time formulation is seldom used. However, the continuous-time formula (7.6) best explains the name of the formula:

If μ_U represents the density function over a rigid body of volume U and $\tilde{u}$ is a coordinate variable, say the horizontal x-coordinate (in a three-dimensional x-y-z Cartesian space), then formula (7.6) gives the x-coordinate of the center of mass (gravity) of the body (at time t), as is well known in Calculus and Mechanics.

(ii) The "center-of-sums" formula:

$$u((k+1)T) = \frac{\sum_{i=1}^{N} u_i(kT).\sum_{j=1}^{N} \mu_{U_j}(u_i(kT))}{\sum_{i=1}^{N}\sum_{j=1}^{N} \mu_{U_j}(u_i(kT))} \quad \ldots(7.7)$$

This formula can be computed faster than the "center-of-gravity" formula (7.5). Its continuous-time version is

$$u(t) = \frac{\int_U \tilde{u}.\sum_{j=1}^{N} \mu_{U_j}(\tilde{u})d\tilde{u}}{\int_U \sum_{j=1}^{N} \mu_{U_j}(\tilde{u})d\tilde{u}} \quad \ldots(7.8)$$

(iii) The "mean-of-maxima" formula:

For each rule R^i, i = 1, ..., N, we first find the particular control output value $u_i(kT)$ from its value range (interval) U_i at which its membership value reaches the maximum: $\mu_{U_i}(u_i(kT))$ = maximum (= 1 in the normalized case). If there are several such values $u_i(kT)$ at which their membership values reach the maxima, then we count the multiplicities. If there is a continuum (subinterval) of such values $u_i(kT)$ at which their membership values reach the maxima, then we pick the smallest and the largest values and count them just twice. Let us say the total number of such values of $u_i(kT)$, at which their membership values reach the maxima, is M. Then we take their mean (average) in the following form:

$$u(kT) = \frac{1}{M}\sum_{i=1}^{M} \tilde{u}_j(kT) \quad \ldots(7.9)$$

Its continuous-time version is

$$u(t) = \frac{1}{2}\{\inf\{u \in U \mid \mu_U(u) = \max\text{imum}\} + \sup\{u \in U \mid \mu_U(u) = \max\text{imum}\}\} \ldots(7.10)$$

Ther e are other defuzzification formulas suggested in the literature, each of which has its own rationale, either from logical argument or from geometrical reasoning, or from control principles. We will however only discuss the "center-of-gravity" formula (7.5) throughout the chapter.

(b) Just like the first step of the fuzzification module, this step of the defuzzification module transforms the after all control output, u, obtained in the previous step, to the corresponding physical values (position, voltage, degree, etc.) that the system (plant, process) can accept. This converts the fuzzy logic controller's numerical output to a physical means that can actually drive the given plant (process) to produce the expected outputs.

The overall fuzzy logic controller that combines the "fuzzification – rule base - defuzzification" modules has been shown in Figure 7.7 and is the controller block in the closed-loop set-point control system of Figure 7.6.

Finally, it is very important to recall that the design of the fuzzy logic controller that we discussed in this section does not require any information about the system (plant, process). It doesn't matter if the system is linear or nonlinear (nor the order of linearity, structure of nonlinearity, etc.) as long as its output, y, can be measured by a sensor and used for the control. In other words, the design of the fuzzy logic controller is independent of the mathematical model of the system under control. Comparing it with the conventional controllers design, its advantages are obvious.

The price that we have to pay for this success is the complexity in computation in the design: the "fuzzification - rule base - defuzzification" steps. But this is what it is supposed to be: under a worse condition (the system model is unknown, or only vaguely given). If we want to design a controller to achieve the same (or better) set-point tracking performance, then we have to do more analysis and calculation in the design. This is generally true in a sensible comparison of a fuzzy logic controller with a similar conventional one. It is likely impossible to obtain a better controller with less effort for a system under even worse conditions. However, the rapid development of modern high-speed computers has made such computational tasks affordable.

7.4. EXAMPLES OF MODEL-FREE FUZZY CONTROLLER DESIGN

Two examples are given in this section to illustrate the model-free fuzzy control approach and to demonstrate its effectiveness in controlling a complex mechanical system without assuming its mathematical model.

Example 7.1

We first consider a truck-parking control problem, as shown in Figure 7.12. The objective is to design a fuzzy logic controller, without assuming a mathematical model of the truck, to park the truck anywhere on the x-axis. Suppose that the truck can move forward at a constant speed of $v = 0.5$ m/s and it is assumed that the truck is equipped with sensors that can measure location (x, y) and orientation (angle) θ at all times. The fuzzy logic controller is to provide an input, u, to rotate the steering wheels and, consequently, to maneuver the truck.

In the simulation, the input variables are the truck angle θ and the vertical position coordinate, y, while the output variable is the steering angle (signal), u. The variable ranges are pre-assigned as $-100 \le y \le 100$, $-180° \le \theta \le 180°$, $-30° \le u \le 30°$.

Here, clockwise rotations are considered positive in θ and likewise, counterclockwise are negative. The linguistic terms used in this design are given as follows:

Angle θ	**y-position**	**Steering angle u**
AB: Above	AO: Above much	NB: Negative big
AC: Above center	AR: Above	NM: Negative medium
CE: Center	AH: Above horizontal	NS: Negative small
BC: Below center	HZ: Horizontal	ZE: Zero
BE: Below	BH: Below horizontal	PS: Positive small
	BR: Below	PM: Positive medium
	BO: Below much	PB: Positive big

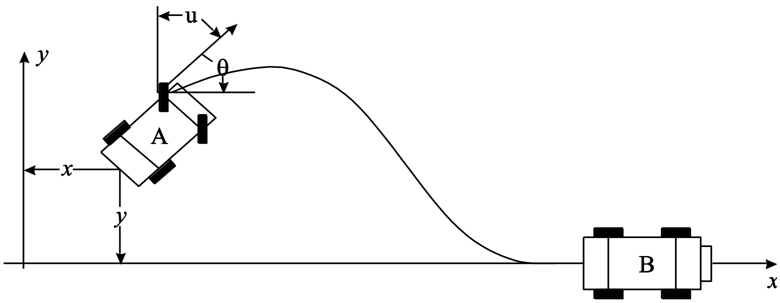


Fig. 7.12 A truck driving control example

The first step is to choose membership functions. They are as shown in Figures 7.12 – 7.15. These choices of their shapes, ranges and overlapping are somewhat arbitrary. We only note that narrow membership functions are used to permit fine control near the designated parking spot, while wide membership functions are used to perform fast controls when the truck is far away from the parking place.

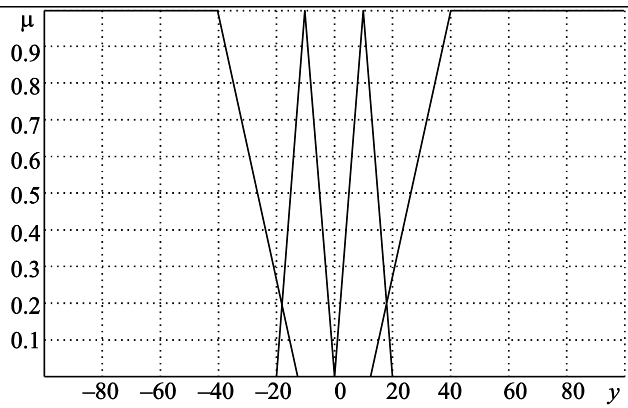


Fig. 7.13 Fuzzy membership functions for the position y

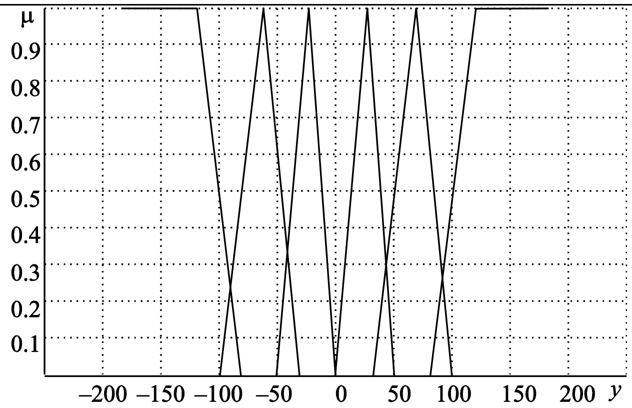


Fig. 7.14 Fuzzy membership functions for the angle θ

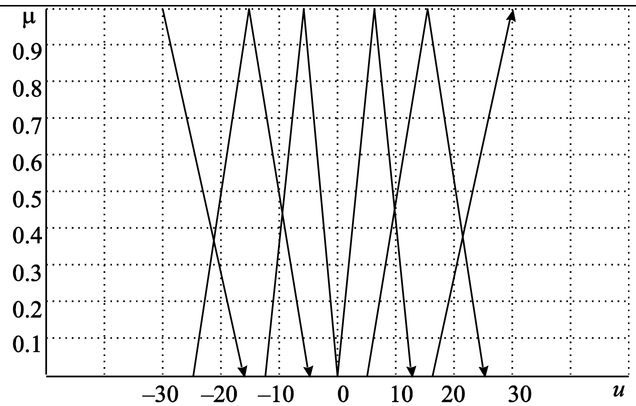


Fig. 7.13 Fuzzy membership functions for the control signal u

The rule base used in simulation is summarized in Table 7.6. Each rule has the form IF y is Y AND θ is Θ THEN u is U, as usual. A glance of these rules reveals the symmetry, an intrinsic property of this controller, which is reasonable for this parking application since the truck can move in any direction. This is not always the case, which can be seen from the next example where symmetry of control rules is not reasonable.

Table 7.6 Fuzzy Controller Rule Base

	BE	BC	CE	AC	AB
BO	PB	PB	PM	PM	PS
BR	PB	PB	PM	PS	NS
BH	PB	PM	PS	NS	NM
HZ	PM	PM	ZE	NM	NM
AH	PM	PS	NS	NM	NB
AR	PS	NS	NM	NB	NB
AO	NS	NM	NM	NB	NB

Finally, we need to obtain the output action under the given input conditions. The following standard weighted average defuzzification formula was used in simulation:

$$u = \frac{\sum_{i=1}^{4} \mu(u_i) u_i}{\sum_{i=1}^{4} \mu(u_i)}$$

A total of 12 computer simulations were performed. Their initial conditions are summarized in Table 7.7, and the corresponding parking performances are shown in Figure 7.16 (a) to (c).

Table 7.7 Initial Conditions of the 12 Simulated Cases

Case	1	2	3	4	5	6	7	8	9	10	11	12
θ	0	0	0	90	90	90	180	180	180	-90	-90	-90
y	30	20	-20	30	10	-20	30	-20	-10	-10	-20	20

The following is a simplified mathematical model for the motion of the truck. This model was used to create data for simulation butwas not used in the design of the controller. This model directly follows from the geometry of the truck (see Figure 7.12):

$$\begin{aligned} \theta(k+1) &= \theta(k) + v\,T \tan(u(k)) / L, \\ x(k+1) &= x(k) + v\,T \cos(\theta(k)), \\ y(k+1) &= y(k) + v\,T \sin(\theta(k)), \end{aligned} \qquad \text{...(7.11)}$$

where

$\theta(k)$ – angle of the truck at time k,

$x(k)$ – horizontal position of the truck rear-end at time k,

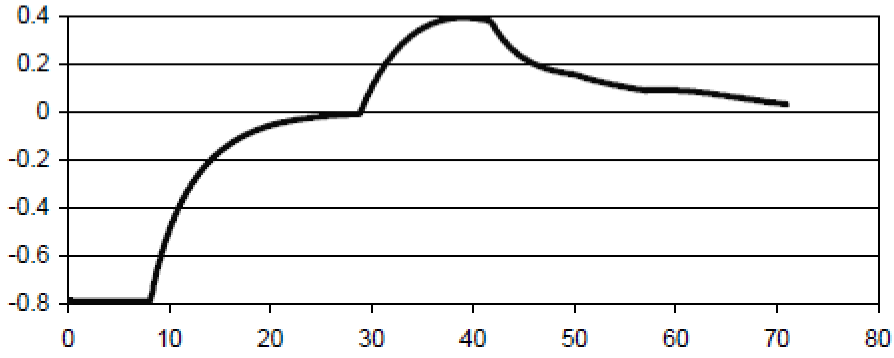
$y(k)$ – vertical position of the truck rear-end at time k,

$u(k)$ – steering angle as control input to the truck at time k,

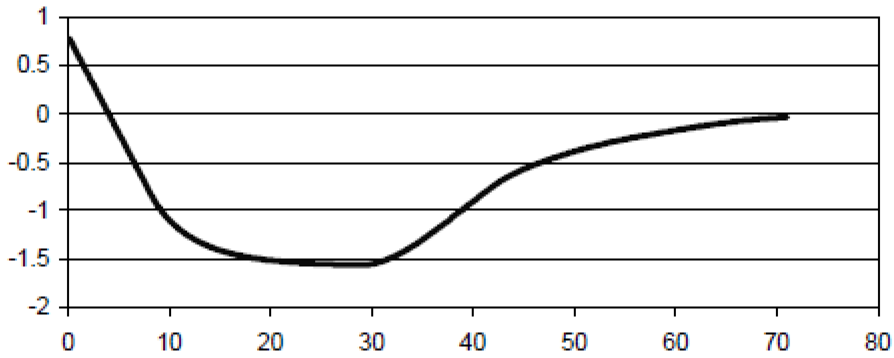
L – length of the truck (L=2.5m),

T – sampling time of the discrete model (T = 0.1s),

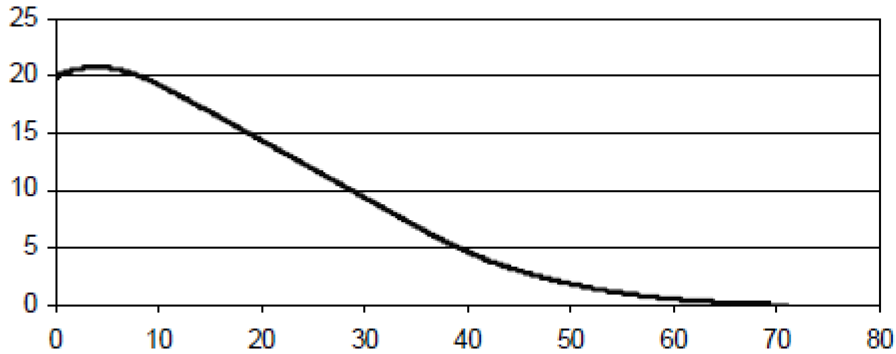
v – constant speed of the truck (v = 0.5m/s



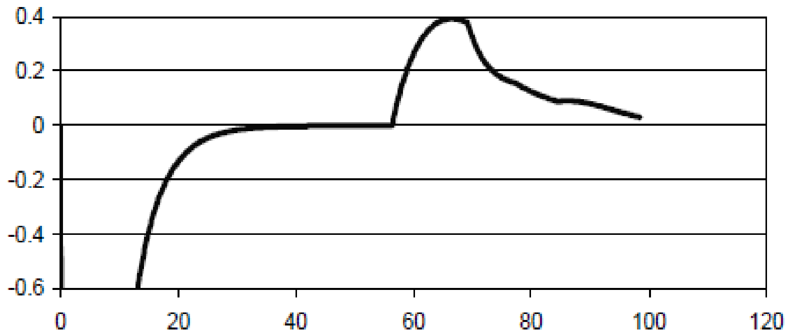
(a) profile of input $u(t)$ vs. time t



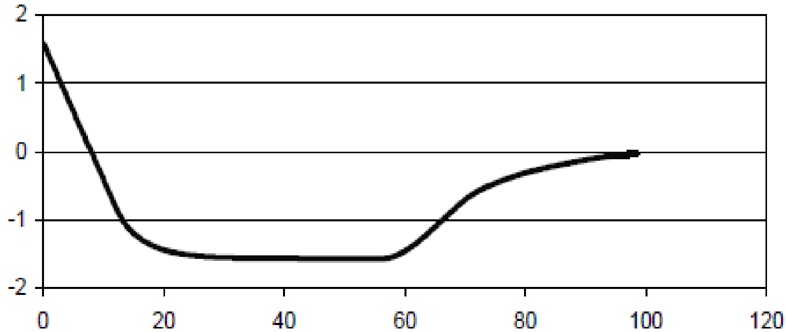
(b) profile of output $\theta(t)$ vs. time t



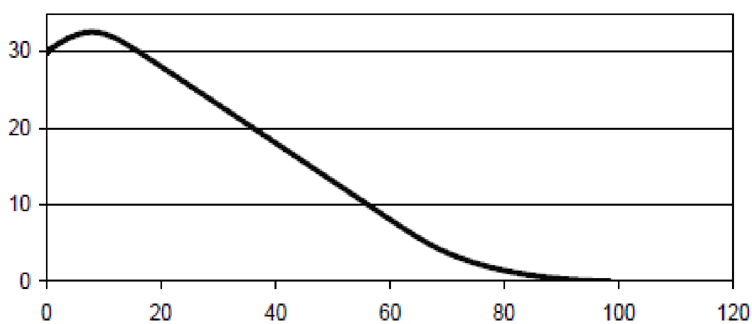
(c) profile of output $y(t)$ vs. time t



(a) profile of input $u(t)$ vs. time t

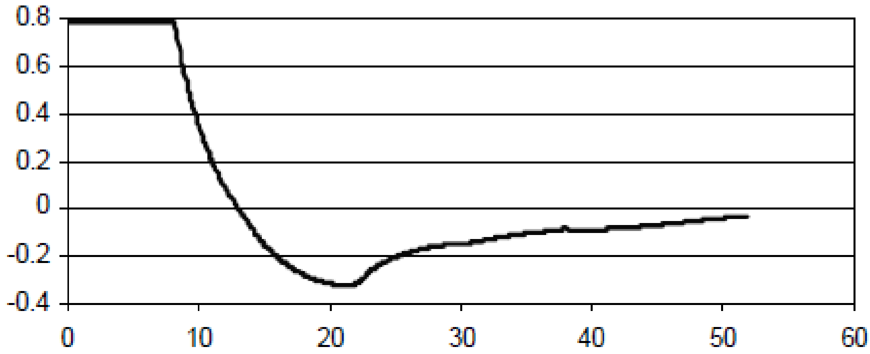


(b) profile of output $\theta(t)$ vs. time t

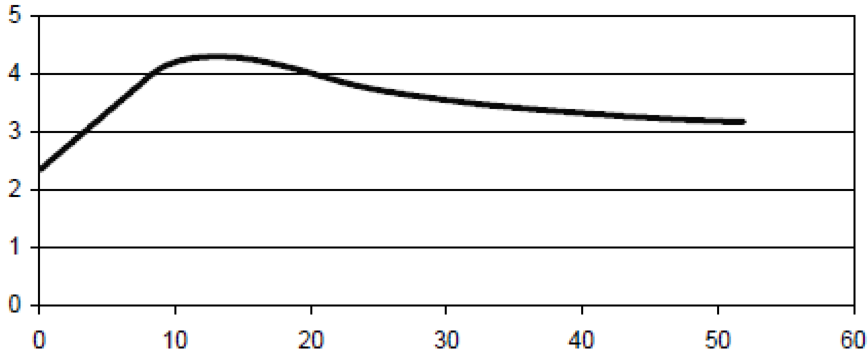


(c) profile of output $y(t)$ vs. time t

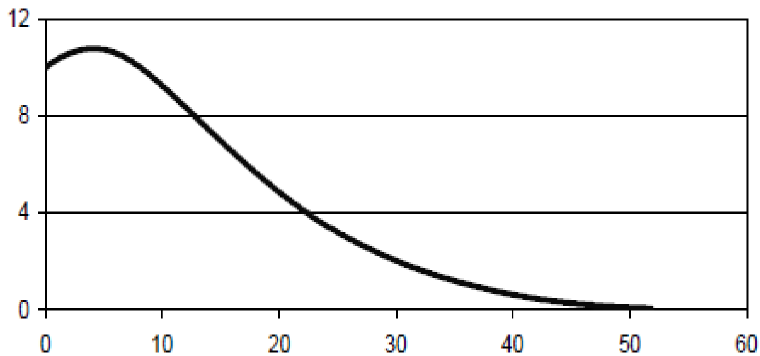
Fig. 7.16 – B Model free parking control simulation results for the angle θ



(a) profile of input $u(t)$ vs. time t



(b) profile of output $\theta(t)$ vs. time t



(c) profile of output $y(t)$ vs. time t

Fig. 7.16 – C Model free parking control simulation results for the control signal u

Example7.2

In this example, we show a landing control problem of a model plane from a model-free approach. This airplane landing problem is visualized by Figure 7.17, which differs from the truck-parking control problem of Example7.1because of the airplane is not allowed to "overshoot" the set-point (i.e., cannot hit the ground) in order to avoid crashing. Therefore, the control rule base will not have symmetry in this example.

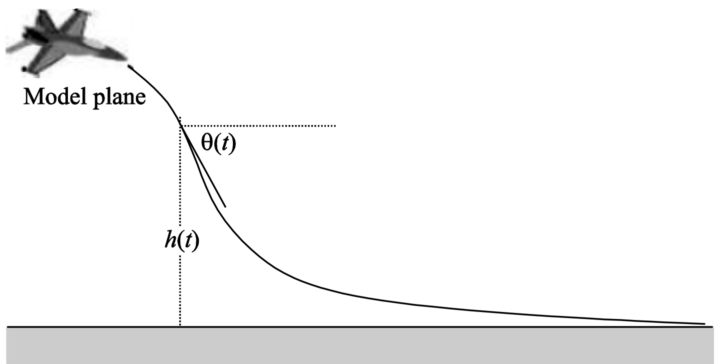


Fig. 7.17 A model airplan landing control problem

The objective is to design a fuzzy logic controller to land the plane safely, anywhere on the ground (the x-axis). Assume that no mathematical model for the plane is available for the design, but the plane is equipped with sensors that can measure the height h(t) and the angle θ of motion of the plane. Suppose that the model plane is moving forward at a constant speed of v=5 m/s, and the controller is used to steer the angle of the plane. Let the initial conditions be h(0)=1000m and θ(0)=0 deg. (horizontally). The fuzzy logic controller is used to provide an input, u, which controls the angle of the plane, so as to guide it to land on the ground safely.

The parameters involved in the control system are as follows:

θ(t)	–	angle of the plane at time t,
h(t)	–	vertical position of the plane at time t,
u(t)	–	control input to the plane at time t,
T	–	sampling time for discretization (1s),
v	–	constant speed of the plane (5m/s).

We first define the ranges of the plane position and angle, as well as that of the control input throughout the landing process:

$$-\frac{\pi}{4} \leq \theta(t) \leq 0,\ -\frac{\pi}{4} \leq u(t) \leq 0 \text{ and } 0 \leq h(t) \leq 1000.$$

Since the plane can land anywhere on the ground, we do not specify the parameter for the horizontal axis. The final state is defined by $h = 0 \pm 1$m and $\theta = 0 \pm 0.1$ deg. (horizontally) to ease the simulation.

The following values are chosen before fuzzification, in order to quantify the linguistic terms of positive small, negative large, etc.

height h

h	Z	PS	PM	PL
value	0	300	600	1000

Angle θ

θ	NL	NM	NS	Z
value	$-\pi/4$	$-\pi/6$	$-\pi/8$	0

The membership functions for the height h and the angle θ are chosen as shown in Figures 7.18 and 7.19 respectively.

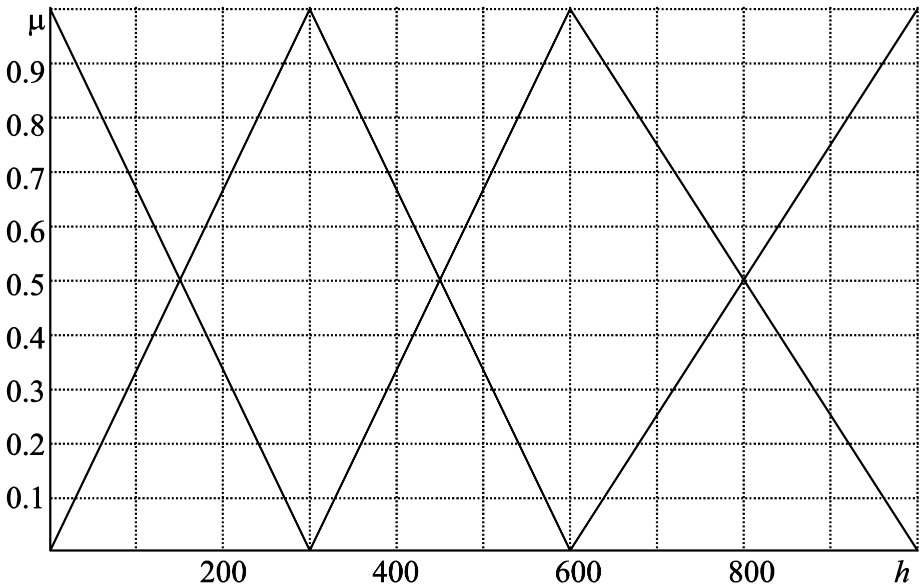


Fig. 7.18 Fuzzy membership function for the height (h)

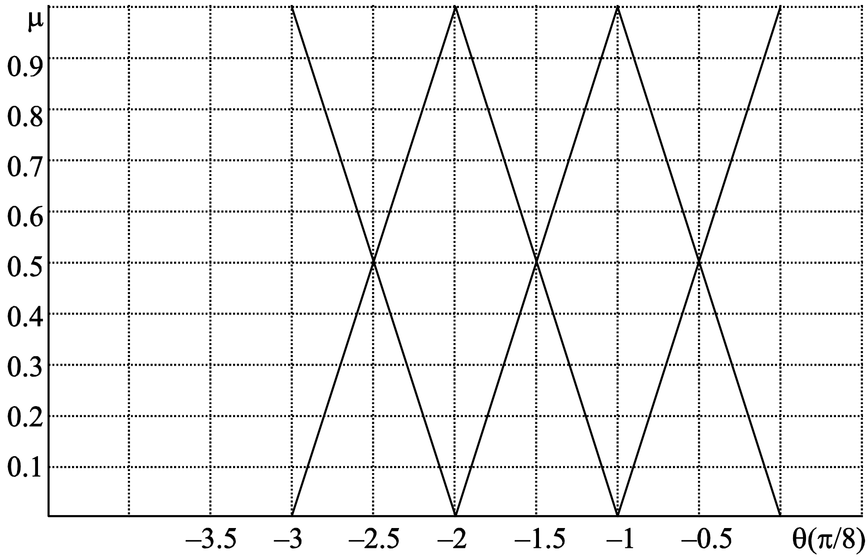


Fig. 7.19 Fuzzy membership function for the angle (θ)

The fuzzy rule base used in the simulation is shown in Table 7.8, where V means "very." It is also in the form of IF h is H AND θ is Θ THEN u is U.

Table 7.8 Fuzzy Control Rule Base

θ h	NL	NM	NS	Z
Z	PMS	PS	PVS	Z
PS	PS	PVS	Z	NVS
PM	PVS	Z	NVS	NS
PL	Z	NVS	NS	NMS

The basic idea in constructing this simple control rule base is as follows:

First, if the plane is far away from the ground, we increase the degree of the plane angle (downward). Second, as the plane approaches the ground, the desired plane angle will gradually turn from downward to horizontal, so that it can land smoothly and safely.

The defuzzfication formula used is, again, the weighted average of all control actions, where the weights are the corresponding membership values given by

$$u = \frac{\sum_{i=1}^{n} \mu(u_i) u_i}{\sum_{i=1}^{n} \mu(u_i)}$$

For the purpose of computer simulation, the following simplified mathematical model of the plane was used to create data for control (but it was not used in the controller design):

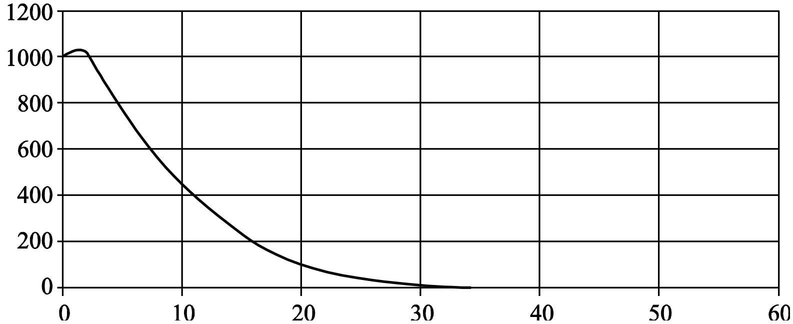


Fig. 7.20 Airplan-landing control simulation result

$$\theta(k+1) = \theta(k) + vT \tan(u(k))$$

$$y(k+1) = y(k) + vT \sin(u(k)). \qquad \ldots(7.12)$$

A simulation result is shown in Figure 7.20 and that demonstrates the safe landing process of the plane under the designed fuzzy logic controller

7.5 THE CONTROL PROBLEM (FUZZY LOGIC) MODEL-BASED APPROACH

In section 7.3, we have seen that fuzzy logic controllers can be designed and work even without any knowledge about the structure of the system (plant, process) for set-point tracking problems, if the system output can be measured and used on-line. The main idea was to use the error signal (the difference between the reference signal and the system output) to drive the fuzzy logic controller, so as to create a new control action each time.

These control actions should be able to alter the system outputs in such a way that the error signal is reduced until satisfactory set-point tracking performance is achieved.

If the mathematical model, or a fairly good approximation of it is available, one may expect to be able to design a better fuzzy logic controller with performance specifications and guaranteed stability. In this section, we discuss another general approach of fuzzy logic control that uses an approximate mathematical model of the given system (plant, process).

In the section 7.3, we used the set-point tracking control problem as a prototype for illustrating the basic ideas and methods. To facilitate our discussion in this section, we use the truck-driving example discussed in Example 7.1 above as a platform.

Consider the simplified truck-driving control problem described in Example7.1. Suppose that the truck is in the initial position A and we want to drive it to the target position B as shown in Figure 7.12. In order not to get the actuator-dynamics involved in this study of basic design principles, we again assume that the truck is driven by the actuator in a constant speed v while the control action is applied only to the steering angle of the front wheelers of the truck. Thus, a very simple mathematical model for the motion of the truck can be obtained from the geometry as shown in (7.11), which was used for simulation purposes before but is employed for controller design here. The model to be used is

$$\begin{aligned} \theta(k+1) &= \theta(k) + v\,T\tan(u(k)) / L, \\ x(k+1) &= x(k) + v\,T\cos(\theta(k)), \\ y(k+1) &= y(k) + v\,T\sin(\theta(k)), \end{aligned} \qquad \ldots(7.13)$$

with notation as defined in (7.11).

We note that this mathematical model, as it stands, is very simple and brief for describing the motion of the truck. Nevertheless, our design of a fuzzy logic controller u(k) that controls the steering angle will be based on this approximate model.

We first discuss the general approach to this control problem. To begin, we convert this mathematical model to a fuzzy model described by the following IF-THEN rules (see Section II.B of Chapter 3):

$$R_S^i: \quad \text{IF } \theta(k) \text{ is } \Theta_i \text{ AND } x(k) \text{ is } X_i \text{ AND } y(k) \text{ is } Y_i$$

$$\text{THEN } \mathbf{x}(k+1) = \mathbf{f}_i(\mathbf{x}(k)) + \mathbf{b}_i(u(k)),\ i = 1,\ldots,N, \qquad \ldots(7.14)$$

where

$$\mathbf{x}(k) = [\theta(k)\ x(k)\ y(k)\ ]^T,$$

$$f_i(x(k)) = \begin{bmatrix} \theta(k) \\ x(k) + vT\cos(\theta(k)) \\ y(k) + vT\sin(\theta(k)) \end{bmatrix},$$

$$b_i(u(k)) = \begin{bmatrix} vT\tan(u(k)) / L \\ 0 \\ 0 \end{bmatrix}$$

We then calculate the new state vector by the following standard weighted average formula:

$$x(k+1) = \frac{\sum_{i=1}^{N} w_i(k) x_i(k+1)}{\sum_{i=1}^{N} w_i(k)}, \qquad \ldots(7.15)$$

where

$w_i(k) = \min\{\mu_{\theta,i}(\theta(k)), \mu_{x,i}(x(k)), \mu_{y,i}(y(k))\}$,

$$w_i(k) \geq 0, \qquad \sum_{i=1}^{N} w_i(k) > 0,$$

in which $\mu_{\theta,i}(.)$, $\mu_{x,i}(.)$, $\mu_{y,i}(.)$ are membership functions defined on the fuzzy subsets Θ, X, Y, and $\mu_{\theta,i}$, $\mu_{x,}$i, $\mu_{y,i}$ are the corresponding membership values of θ(k), x(k), y(k) in the IF-THEN rule R^i, i = 1,..., N, respectively. Example 7.4.1discussed above shows a concrete case of this description.

We now design a state-feedback fuzzy logic controller for guiding the steering angle of the front wheelers of the truck. This controller is governed by the following fuzzy IF-THEN rules:

R_C^i : IF θ(k) is Θ_i AND x(k) is X_i AND y(k) is Y_i

THEN $u_i(k) = \mathbf{K}_i\, \mathbf{x}(k)$, i = 1,...,N, …(7.16)

where $\{\mathbf{K}_i\}$ are constant matrices (control gains) to be determined in the design with the averaged control action given by

$$u(k) = \frac{\sum_{i=1}^{N} w_i(k) K_i x_i(k+1)}{\sum_{i=1}^{N} w_i(k)}, \qquad \ldots(7.17)$$

where for simplicity we use the same weights as those obtained in the system model above. Of course, it is possible to use different weights for the controller to make it more effective in a real design.

Since this is a design, we may take different approaches. Here, to take a simple approach, we continue our design by further simplifying the models in two aspects:

(i) We only study the simple case of horizontal parking of the truck, to position B of Figure 7.12, where the final horizontal position, x(k), is not specified. Hence, we can simply delete the second equation of (7.13) without altering the control effect on the process.

(ii) We linearize the rest of the mathematical model as follows:

$$\theta(k+1) = \theta(k) + \frac{vT}{L} u(k),$$

$$y(k+1) = y(k) + vT\theta(k),$$

where we note that

$$-\theta(k) \leq \sin(\theta(k)) \leq \theta(k), \qquad -\pi \leq \theta(k) \leq \pi,$$

and

$\theta(k) \to 0 \quad \Rightarrow \quad \sin(\theta(k)) \to 0,$

$\theta(k) \to \pm\pi \quad \Rightarrow \quad \sin(\theta(k)) \to 0.$

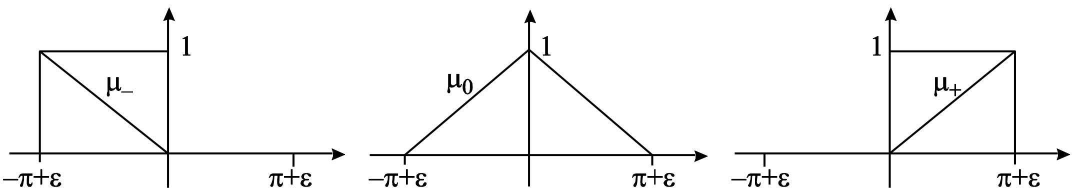


Fig. 7.21 Membership function for truck model linearization

To avoid the singular points $\pm\pi$ at which the linearization is invalid, we only consider the range

$$-\pi + \varepsilon \le \theta(k) \le \pi - \varepsilon \text{ for a small } \varepsilon > 0$$

Under these two simplifications, we obtain the following two linearized models for the truck-driving control problem:

S_1 : (when $\theta(k) \to 0$)

$$\begin{bmatrix} \theta(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ vT & 1 \end{bmatrix} \begin{bmatrix} \theta(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} vT/L \\ 0 \end{bmatrix} u_1(k);$$

S_2 : (when $\theta(k) \to \pm\pi \mp \varepsilon$)

$$\begin{bmatrix} \theta(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \varepsilon vT & 1 \end{bmatrix} \begin{bmatrix} \theta(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} vT/L \\ 0 \end{bmatrix} u_2(k)$$

Note that in these two linearized models, the data set $\{y(k)\}$ is actually not used (they are zeroed out by the coefficient matrix) for the changes of the angle $\theta(k)$. This is reasonable since the truck is moving to the horizontal direction, $\theta(k) \to 0$, independently of the vertical position of the truck.

Converting these mathematical models to fuzzy ones, we have

$\tilde{R}_S^1$: IF $\theta(k)$ is approximately 0

THEN $\mathbf{x}_1(k+1) = \mathbf{A}_1\, \mathbf{x}(k) + \mathbf{b}_1\, u(k)$;

$\tilde{R}_S^2$: IF $\theta(k)$ is approximately $\pi - \varepsilon$ OR $\theta(k)$ is approximately $-\pi + \varepsilon$

THEN $\mathbf{x}_2(k+1) = \mathbf{A}_2\, \mathbf{x}(k) + \mathbf{b}_2\, u(k)$,

where $\mathbf{x}(k) = [\theta(k)\ y(k)]^T$, $\quad A_1 = \begin{bmatrix} 1 & 0 \\ vT & 1 \end{bmatrix}$, $\quad A_2 = \begin{bmatrix} 1 & 0 \\ \varepsilon vT & 1 \end{bmatrix}$ and

$$b_1 = b_2 = \begin{bmatrix} vT/L \\ 0 \end{bmatrix}$$

with the membership functions for linearization defined by μ_-, μ_0, and μ_+ in Figure 7.21. Note that $\tilde{R}_S^2$ is equivalent to the following two standard IF-THEN rules:

$\tilde{R}_S^{21}$: IF $\theta(k)$ is approximately $\pi - \varepsilon$

THEN $\mathbf{x}_{21}(k+1) = \mathbf{A}_2\,\mathbf{x}(k) + \mathbf{b}_2\,u_{21}(k)$;

$\tilde{R}_S^{22}$: IF $\theta(k)$ is approximately $-\pi + \varepsilon$

THEN $\mathbf{x}_{22}(k+1) = \mathbf{A}_2\,\mathbf{x}(k) + \mathbf{b}_2\,u_{22}(k)$.

In the above rules $\tilde{R}_S^1$, $\tilde{R}_S^{21}$ and $\tilde{R}_S^{22}$ have the associated membership functions μ_-, μ_0 and μ_+ respectively. Hence, the averaged state vector is given by

$$x(k+1) = \frac{w_1(k)x_1(k+1) + w_2(k)x_{21}(k+1) + w_3(k)x_{22}(k+1)}{w_1(k) + w_2(k) + w_3(k)}, \qquad \ldots(7.19)$$

where $w_1(k) = \mu_0(\theta(k))$, $w_2(k) = \mu_+(\theta(k))$, and $w_3(k) = \mu_-(\theta(k))$.

The fuzzy logic controller for this system is described by the following two fuzzy IF-THEN rules:

R_C^1 : IF $\theta(k)$ is approximately 0

THEN $u(k) = \mathbf{K}_1\,\mathbf{x}(k) = [\,K_{11}\ K_{12}\,]\,\mathbf{x}(k)$;

$\bar{R}_C^2$: IF $\theta(k)$ is approximately $\pi - \varepsilon$ OR $\theta(k)$ is approximately $-\pi + \varepsilon$

THEN $u(k) = \mathbf{K}_2\,\mathbf{x}(k) = [K_{21}\ K_{22}]\,\mathbf{x}(k)$,

where the constant gains, K_{11}, K_{12}, K_{21} and K_{22} are to be determined in the design for the control task driving the truck from position A to the horizontal position B as shown in Figure 4.12.

Similar to the system rules, the second control rule is equivalent to the following two standard IF-THEN rules:

R_C^{21}: IF $\theta(k)$ is approximately $\pi - \varepsilon$

THEN $u_{21}(k) = \mathbf{K}_2\,\mathbf{x}(k) = [\,K_{21}\ K_{22}\,]\,\mathbf{x}(k)$;

R_C^{22} : IF $\theta(k)$ is approximately $-\pi + \varepsilon$

THEN $u_{22}(k) = \mathbf{K}_2\,\mathbf{x}(k) = [K_{21}\ K_{22}]\,\mathbf{x}(k)$,

and the averaged control is given by

$$u(k) = \frac{w_1(k)u_1(k) + w_2(k)u_{21}(k) + w_3(k)u_{22}(k)}{w_1(k) + w_2(k) + w_3(k)}, \qquad \ldots(7.20)$$

where as mentioned above, we use the same weights $w_1(k)$, $w_2(k)$, and $w_3(k)$ as those used in the system rules for simplicity of discussion.

Observe that our control task in this example is to drive the truck to position B where $\theta(k^*) = 0$, $x(k^*)$ is not specified ("don't care"), and $y(k^*) = 0$, for all large values of k^*. This is guaranteed if we can design the fuzzy controller $u(k)$ such that the equilibrium point $\mathbf{x}^* = \mathbf{0}$ of the linear controlled system (described by R_S^i and R_C^i for I=1, 2) becomes asymptotically stable. If so, then $\theta(k) \to 0$, $y(k) \to 0$ as $k \to \infty$.

Therefore, we next complete the design by determining the constant control gains, $\mathbf{K}_1 = [K_{11}\ K_{12}]$ and $\mathbf{K}_2 = [K_{21}\ K_{22}]$, such that the feedback controlled system is asymptotically stable about its equilibrium point $\mathbf{x}^* = [\theta^*\ y^*]T = \mathbf{0}$.

By substituting the control rules R_C^1, R_C^{21}, R_C^{22} into R_S^1, R_S^2 and using the averaged state vector (7.19), in which they have the same weights, we obtain

$$x(k+1) = \frac{w_1(k)H_1 + w_2(k)H_2 + w_3(k)H_3}{w_1(k) + w_2(k) + w_3(k)} x(k), \qquad \ldots(7.21)$$

where

$$H_1 = A_1 + b_2K_1 = \begin{bmatrix} 1 + vTK_{11}/L & vTK_{12}/L \\ vT & 1 \end{bmatrix},$$

$$H_2 = \frac{1}{2}[(A_1 + b_1K_1) + (A_2 + b_2K_2)] = \begin{bmatrix} 1 + vT(K_{11} + K_{21})/2L & vT(K_{12} + K_{22})/2L \\ (1+\varepsilon)vT/2 & 1 \end{bmatrix}$$

$$H_3 = A_2 + b_2K_2 = \begin{bmatrix} 1 + vTK_{21}/L & vTK_{22}/L \\ \varepsilon vT & 1 \end{bmatrix}$$

It then follows from "The discrete-time dynamic fuzzy system is asymptotically stable about the equilibrium point 0 if there exists a common positive definite matrix P such that, $A_iTPA_i - P$ □0, for all i=1,2,...,N."

If we can choose K_{11}, K_{12}, K_{21}, and K_{22} such that there is a common positive definite matrix $\mathbf{P}$ satisfying $\mathbf{H}_i^T\ \mathbf{P}\mathbf{H}_i - \mathbf{P} < 0$, i = 1, 2, 3,then the feedback controlled system (7.21) will be asymptotically stable about its equilibrium point $\mathbf{x}^* = \mathbf{0}$.

Table 7.9 Different Initial Positions of the Truck

	θ(0)[deg.]	y(0)[m]
1	0	30
2	0	20
3	0	10
4	0	-10
5	0	-20
6	0	-30
7	90	30
8	90	20
9	90	10
10	90	-10
11	90	-20
12	90	-30
13	180	30
14	180	20
15	180	10
16	180	-10
17	180	-20
18	180	-30
19	-90	30
20	-90	20
21	-90	10
22	-90	-10
23	-90	-20
24	-90	-30

As a numerical example, let

$L = 2.8$ m, $T = 1.0$ sec.,

$v = 1.0$ m/sec., $\varepsilon = 0.01\pi$,

$K_{11} = -0.4212, \;\; K_{12} = -0.02944,$

$K_{21} = -0.0991, \;\; K_{22} = -0.00967.$

It can be verified that

$$P = \begin{bmatrix} 989.0 & 75.25 \\ 75.25 & 26.29 \end{bmatrix} > 0\,.$$

It implies that

$$H_1^T P H_1 - P = \begin{bmatrix} -120.4 & 6.008 \\ 6.008 & -1.468 \end{bmatrix} < 0,$$

$$H_2^T P H_2 - P = \begin{bmatrix} -68.31 & -5.873 \\ -5.873 & -0.5079 \end{bmatrix} < 0 \text{ and}$$

$$H_3^T P H_3 - P = \begin{bmatrix} -100.0 & -0.000114 \\ -0.000114 & -1.0 \end{bmatrix} < 0.$$

The computer simulation results of this fuzzy control, for the linearized truck-driving system, with the target angle $\theta = 0$ and with 24 different initial positions (the position A in Figure 7.12) listed in Table 7.9 are shown in Figures 7.21 (A) - (D).

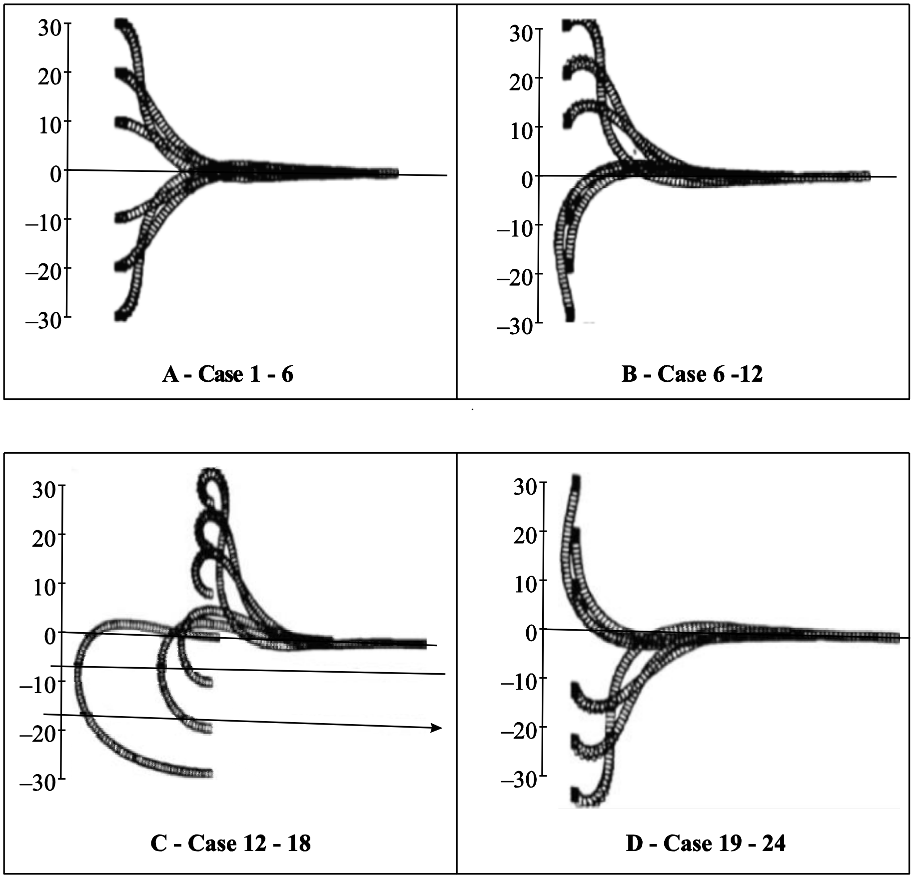


Fig. 7.22 control simulation results for Cases

EXERCISE-7

1. A conventional machine tool working in an automatic cycle is under the control of a programmable logic controller (PLC). This machine tool is loaded and unloaded by an industrial robot arm that operates according to a 2-second load cycle and then a 2-second unload cycle. There are several other activities to be performed at scheduled sequential times, some of them overlapping, as summarized in Figure 7.23

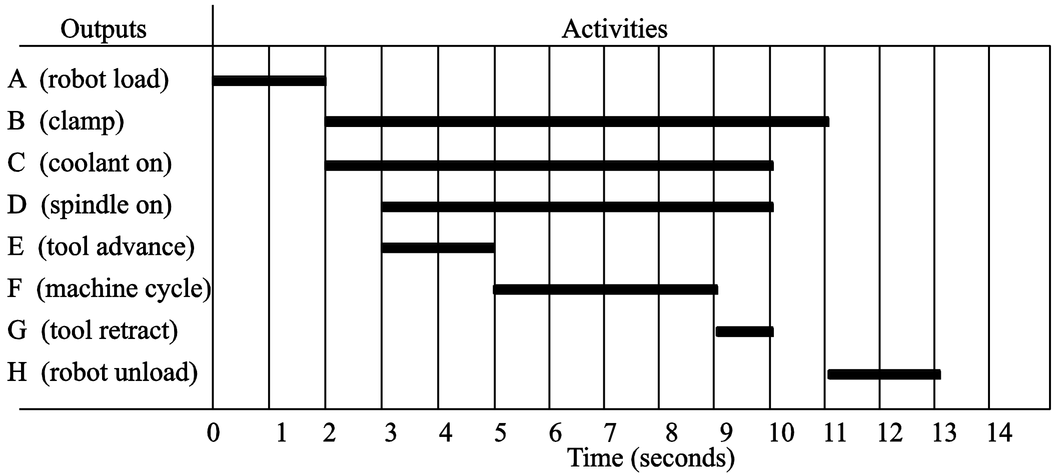


Fig. 7.23 Activities of machine tool

Fill out the drum-timer array displayed in Table 7.10 for the machine tool.

Table 7.10 PLC drum-timer array

Steps	Counts (1 count/sec)	Outputs							
		A	B	C	D	E	F	G	H
1	2								
2	1								
3	2								
4	4								
5	1								
6	1								
7	2								

2. Consider the ladder logic diagram shown in Figure 7.24. Show what the following logical operations in this diagram perform (in Boolean algebraic notation):

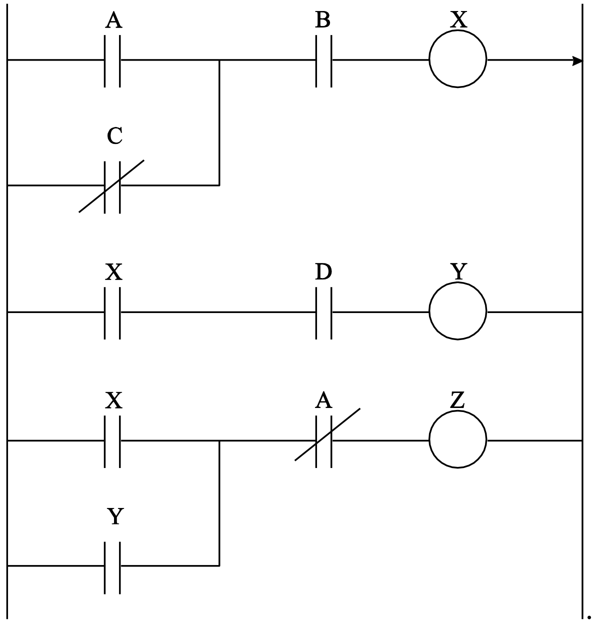


Fig. 7.24 adder logic diagram

3. (Computer programming project)

Consider a unitless set-point tracking system shown in Figure 7.6, where the set-point r = 2.0; the plant is given by the nonlinear model

$$y((k+1)T) = \frac{T}{2}[y(kT)]^2 + u(kT), \qquad \ldots(7.22)$$

where T = 0.1 is the sampling time, and u(kT) is the control input to the plant created by the fuzzy logic controller shown in Figure 7.7. Follow the general design principle discussed in Section 7.5 to design a fuzzy logic controller for this set-point tracking task (see Figure 7.9).

In the design, use the four membership functions shown in Figure 4.9 for both the error signal e(kT) and the change of error e& (kT), and the membership functions shown in Figure 4.10 for the control u(kT) and in the fuzzy logic control rule base. The common real constant H > 0 in the membership functions in Figures 7.10 and 7.11 can be used as a tuning parameter for improving the tracking performance when programming the control. Use the following initial conditions:

$$y(0) = 0,\ e(0) = r - y(0) = 2.0,$$

and

$$\dot{e}(0) = \frac{1}{T}[e(0) - e(-T)] = \frac{1}{0.1}[2.0 - 2.0] = 0\,.$$

Describe clearly the entire design procedure, write a computer program (in any computer programming language) for the described simulation, and plot the final results in a graph similar to Figure 7.9.

4. (Computer programming project)

Consider a model-based fuzzy control problem for an inverted pendulum to track a specified trajectory. Figure 7.25 shows the inverted pendulum system, which has the following mathematical model:

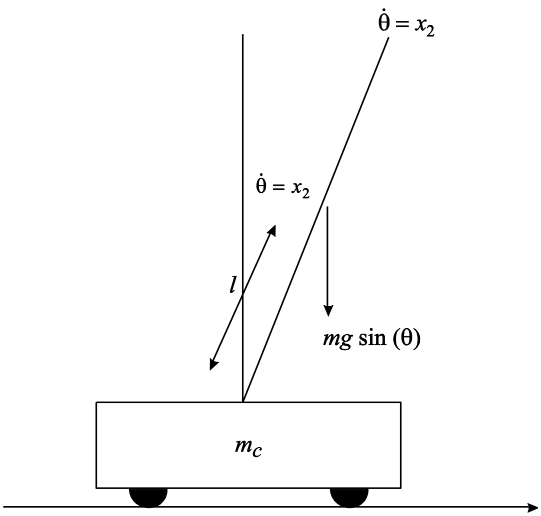


Fig. 7.25 The inverted pendulum system

$$\dot{x}_1 = x_2\,,$$

$$\dot{x}_2 = \frac{g\sin(x_1) - amlx_2^2\sin(2x_1)/2 - a\cos(x_1)u}{4l/3 - aml\cos^2(x_1)}\,, \qquad \ldots(7.23)$$

where x_1 is the angle in radians of the pendulum measured from the vertical axis, x_2 is the angular velocity in rad/sec, g = 9.8 m/sec2 is the gravity acceleration constant, m = 2.0 kg is the mass of the pendulum, $a = (m + M)^{-1}$, M = 8.0 kg is the mass of the cart, 2l = 1.0 m is the length of the pendulum, and u is the force input to the cart. A fuzzy model used to approximate this physical system is as follows:

R^1 : IF x_1 is about 0, THEN $\dot{x} = \mathbf{A}_1 x + \mathbf{B}_1 u$

R^2 : IF x_1 is about $\pm\pi/2$, THEN $\dot{x} = \mathbf{A}_1 x + \mathbf{B}_1 u$, where

$$A_1 = \begin{bmatrix} 0 & 1 \\ \dfrac{g}{4l/3 - aml} & 0 \end{bmatrix}, \qquad B_1 = \begin{bmatrix} 0 \\ -\dfrac{a}{4l/3 - aml} \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 1 \\ \dfrac{2g}{\pi(4l/3 - aml\beta^2)} & 0 \end{bmatrix}, \qquad B_2 = \begin{bmatrix} 0 \\ -\dfrac{a\beta}{4l/3 - aml\beta^2} \end{bmatrix}$$

The membership functions for Rules 1 and 2 are shown in Figure 7.26. Design a model-based fuzzy logic controller of the form $u_i = K_i x_i$ to stabilize this inverted pendulum system such that $x_1 \rightarrow 0$ and $x_2 \rightarrow 0$.

[Hint: First, discretize the model, and then follow the design procedure discussed in Section III. One possible solution is $\mathbf{K}_1$ = [–120.7 –22.7] $\mathbf{K}_2$ = [–255.6 –746.0]. But this solution is by no means unique.]

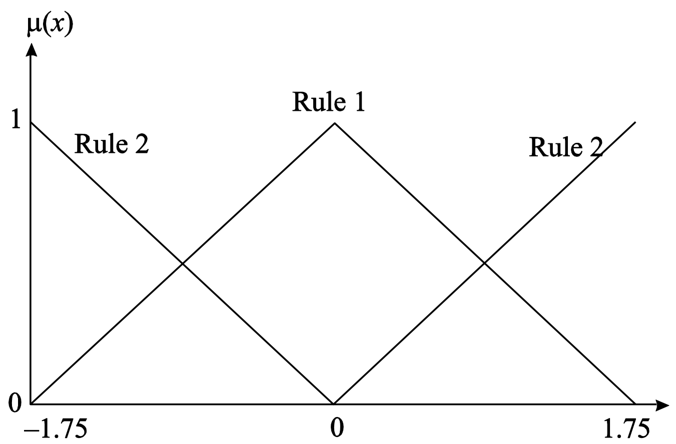


Fig. 7.26 The membership functions

BIBLIOGRAPHY

1. C. R. Asfahl. Robots and Manufacturing Automation. New York, NY: Wiley ,1992.
2. D. Drianker, H. Hellendoorn, and M. Reinfrank. An Introduction to Fuzzy Control. Berlin, Germany: Springer-Verlag,1993.
3. K. Tanaka. "Design of model-based fuzzy control using Lyapunov's stability approach and its applications to trajectory stabilization of a model car." Theoretical Aspects of Fuzzy Control (ed. H. T. Nguyen, M. Sugeno, R. Tong, and R. R. Yager), pp 31-63, New York, NY: Wiley ,1995.
4. Guanrong Chen, Trung Tat Pham. Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control System: CRC Press Boca Raton London New York Washington, D.C.
5. L. X. Wang. Adaptive Fuzzy Systems and Control, Englewood Cliffs, NJ: Prentice-Hall ,1994.
6. J.W.Webb and R.A.Reis. Programmable Logic Controllers: Principles and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1995.
7. L. A. Zadeh (1994). "The role of fuzzy logic in modeling, identification, and control", Modeling, Identification, and Control, Vol. 15, No. 3, pp 191-203

Chapter-8

Stability of Fuzzy Control System

8.1 INTRODUCTION

FKBC has proved to be a powerful tool when applied to the control of processes which are not amenable to conventional, analytic design techniques. The design of most of the existing FKBC has relied mainly on the process operator's or control engineer's experience based heuristic knowledge. Hence, the controller's performance is very much dependent on how good this expertise is. Thus from the control engineering point of view, the major effort in fuzzy knowledge based control has been devoted to the development of particular FKBC for specific applications rather than to general analysis and design methodologies for coping with the dynamic behavior of control loops. The development of such methodologies is of primary interest for control theory and engineering. In particular, stability analysis is of extreme importance, and the lack of satisfactory formal techniques for studying the stability of process control systems involving FKBC has been considered a major drawback of FKBC.

Fuzzy control systems are essentially nonlinear systems. For this reason it is difficult to obtain general results on analysis and design of FKBC. Furthermore, the knowledge of the dynamic behavior of process to be controlled is normally poor. Therefore, the robustness of the fuzzy control system must be studied to guarantee stability in spite of variations in process dynamics.

In this chapter we consider several existing approaches for stability analysis of FKBC. In the fuzzy control literature this type of analysis of FKBC is usually done in the context of the following two view of the system under control:

1. Classical nonlinear dynamic systems theory: The system under control is a "non-fuzzy" system, and the FKBC is a particular class of nonlinear controller.
2. Dynamic fuzzy systems.

The second view is associated with Zadeh`s Extension principle and so far is only of theoretical interest. Research in this area includes stability criteria based on the concept of energy or the controllability of fuzzy systems.

The work presented in this chapter corresponds to the first view. We use the control structure shown in Fig. 8.1, where the FKBC is represented by means of a nonlinear function U = φ(X). It has been shown in Garcia in-Cerezo et al. [1992] and in section 4.2.2 that the FKBC is a nonlinear transfer element represented by the function φ (X). So, the structure of Fig. 8.1 can be used to analyze the dynamic behavior of the closed-loop system.

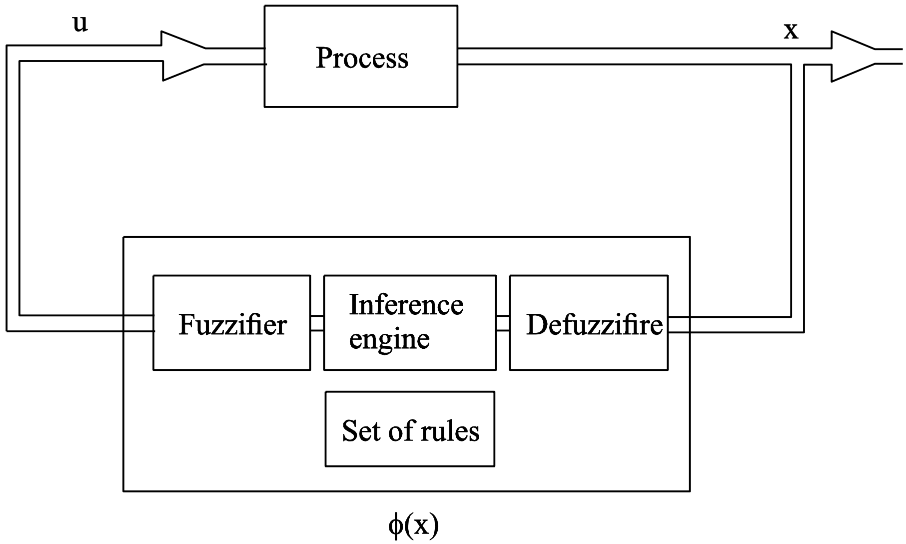


Fig. 8.1. FKBC in a closed-loop

One of the first works dealing with FKBC closed-loop analysis is that of tone [1976]. The analysis is based on the relation matrix, which is a discrete version of the fuzzy relation, Ř or μ_R representing the meaning of the rule base. The nonlinear function Φ(x) can be computed from Ř by means of fuzzification, composition-based inference and defuzzification, as shown in section 8.2 Tong [1978] shows that the relation matrix is dependent only on the set of rules. Thus, the performance of the closed-loop system is improved by modifying these rules.

Braae and Rutherford [1979] introduced the concept of linguistic trajectory of closed-loop FKBC systems. They also show the relation between the dynamic behavior of the closed-loop system and the FKBC rules and at the same time established a relation between the state space representation of the system to be controlled and the FKBC rules. This relation is related to the notion of control space (state space) partition [ref. to Gracia-Cerezo(1987)].

Analysis of the state space (the state space or phase plane in the two-dimensional case) has also been shown to be a simple but effective tool for simple systems. The “geometric method of stability analysis” [ref. to Article Aracil et al.(1988)] is based on a study of the contributions of the vector fields of the plant and controller. The interpretation is very intuitive and can be used to predict the stability under certain conditions. This method is also discussed in section 8.2.

The above geometric method is formalized with the definition of the so-called stability and robustness indices [given in Article Aracil et al.(1989)], based on concepts from the "qualitative theory of dynamical system" [Guckemheimer, J. et al. (1983)]. These indices are not only used to determine the stability and robustness of the closed-loop system, but are also used as a description of its dynamic behavior. Robustness indices are considered in section 8.3. The above methods for stability analysis are based on internal representation of dynamic systems. In the framework of the general stability theory there are two main directions: (1) stability in the Lyapunov sense which refers to internal representation (the state vector tends toward zero), and (input-output stability, which refers to the external representation (relatively small outputs with regard to the input). Section 8.4 is devoted to general concepts in input-output stability.

Vidyasagar [1978] shows that, under some restricitive conditions, internal stability implies external stability and vice versa. Safonov [1980] established a common conceptual framework for the two families of stability criteria, from which one can derive as particular cases the Lyapunov criteria, and the criteria in the input-output family: conicity, circle and popov criteria. Fihure 8.2 shows the relations among these criteria.

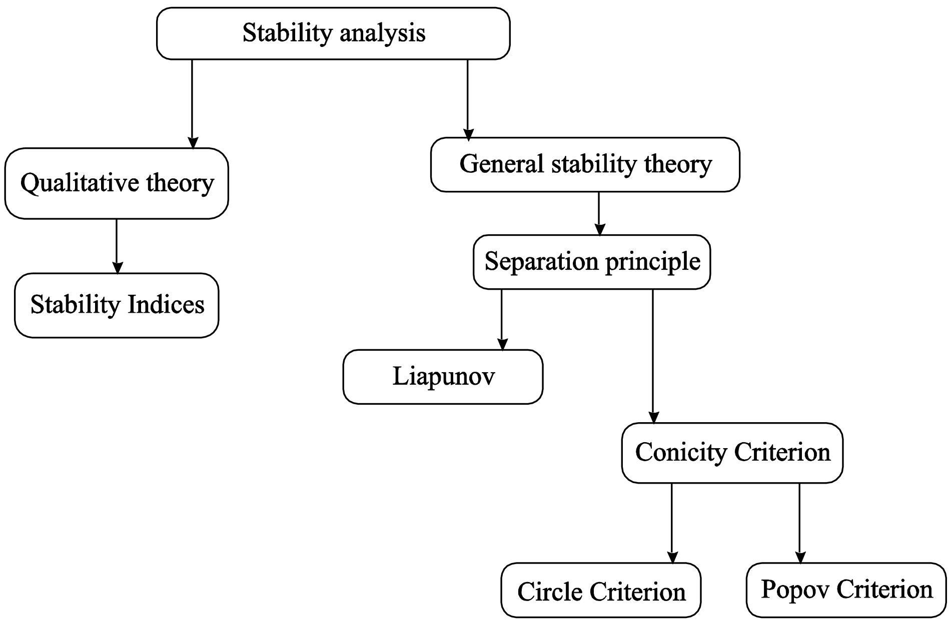


Fig.8.2. Approaches to stability analysis

The application of the classical input-output techniques such as the circle and popov criteria is well known in control theory. The work of Ray and Majumder [1984] is one of the first covering the application of input-output stability in FKBC analysis. More precisely, this is done by the application of the circle criterion. This method is described in section 8.5.

Recently, the conicity criterion has been introduced by Aracil et al. [1991] as a method for the stability analysis and design of FKBC and has been dealt in section 8.6.

It should be noted here that the approaches presented in this chapter assume that a dynamic non-fuzzy model of the process or plant to be controlled is known. In fact, some knowledge about the process dynamic is required for any stability analysis. The important point is that even if the model is only approximately knows, some conclusions about the stability of the closed-loop system can still be obtained. Furthermore, if the closed-loop system is robust enough (far from the point to stability loss) it can maintain stability even if the model does not exactly represent the actual dynamic behavior of the process under control.

8.2 THE STATE SPACE APPROACH

One of the first approaches to the stability analysis of FKBC uses the phase plane and was introduced by Braae and Rutherford [1985]. The practical applications of this method are restricted to Two-dimensional systems due to difficulties in the interpretation of higher order graphical representations of the phase plane.

Stability analysis of a FKBC requires characterizations of the relation between the rules and the state space associated with the dynamic system under control. This relationship is based on the relative influence of each rule of the rule base on the control action produced by the FKBC.
Let us consider a rule base composed by rules where the rules-antecedent contains two process state variables (plant variables), x1 and x2, representing the controller input variables. The consequent contains a single control output, u. The process state variables, x1 and x2 can take n_1 and n_2 linguistic values respectively. Under these conditions, the maximum number of rules contained in the rule base is n1 x n2. The region of study for the analysis of the FKBC in the state space is normally bounded by same finite values xmini, xmaxi where i = 1, 2.
We will now state that the crisp element (x1, x2) of the state space belongs to the subspace of the partition associated with rule j, if it holds that

$$\forall\ j \neq A(x1, x2) : \mu R, (x1, x2) \geq \mu Rk\ (x1, x2) \qquad \ldots (8.1)$$

where μR, (x_1, x_2) and μRk (x_1, x_2) are the fuzzy relations representing the meaning of the rule-antecedents of rules j and k respectively. For example, let
Rule j: if x_1 is $LX^{(j)}{}_2$ and x2 $LX^{(j)}{}_2$ Then μ is $LU^{(j)}$,
Rule k: if x_2 is $LX^{(k)}{}_2$ and x2 $LX^{(k)}{}_2$ Then μ is $LU^{(k)}$.
Thus μRj, and μRk in the case of Mamdani-type implication are defined as follows
$\forall\ x_1,x_2 : \mu Rj\ (x_1, x_2) = \min\ ({}^{\mu}LX_1{}^{(j)},\ {}^{\mu}LX_2{}^{(k)})$ … (8.2)
$\forall$ x1,x2 : $\mu Rk\ (x_1, x_2) = \min\ ({}^{\mu}LX_1{}^{(j)},\ {}^{\mu}LX_2{}^{(k)})$. … (8.3)
In this case, $xmin_1 = -4$, $xmax_1 = 4$, $xmin_2 = -10$, $xmax_2 = 10$.

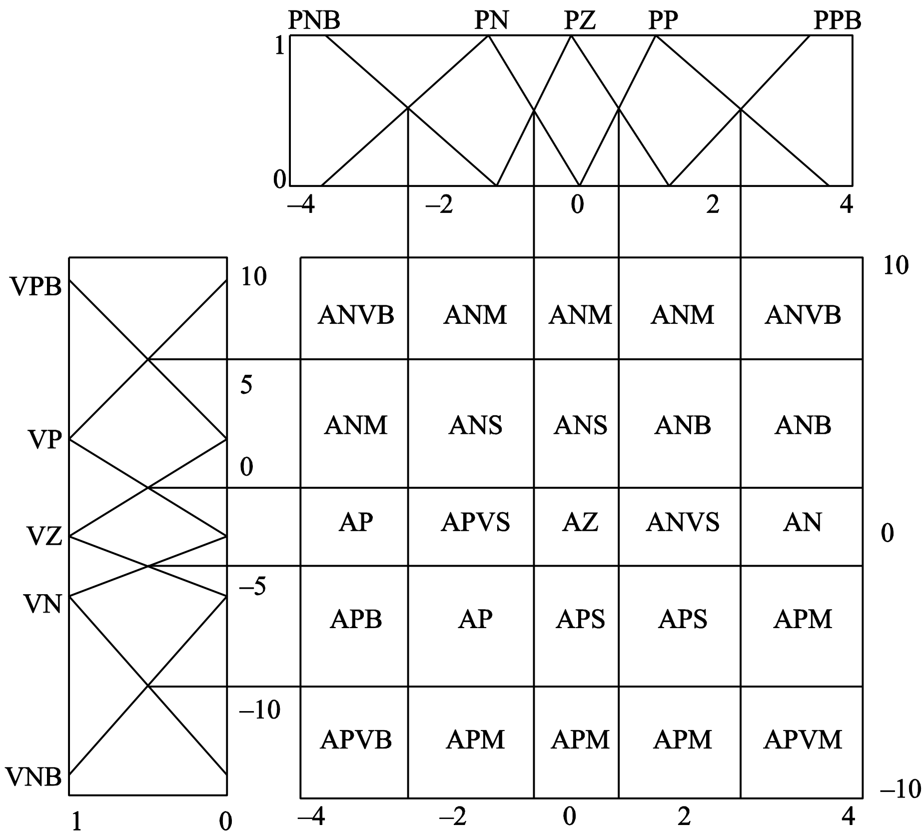


Fig.8.3. Partition of the input space.

The "partition limits" are determined through (8.1). Figure 8.3 shows the determination of the partition for a rule base defined in a table form as shown in fig. 8.4.

A closed-loop system trajectory can be mapped on the partition space as done in Fig. 8.5. It can be seen there how certain areas of the partition space (the gray partition areas, each area corresponding to a particular FKBC rule) relate to the system trajectory. A sequence of rules, obtained according to the order in which they are fired, forms the so-called linguistic trajectory which with the system trajectory in Fig. 8.5a is given by

Linguistic trajectory$_1$ = **(Rule$_{11}$, Rule$_{16}$, Rule$_{17}$, Rule$_{18}$, Rule$_{19}$, Rule$_{14}$, Rule$_9$, Rule$_8$, Rule$_{13}$).**

From a design point of view, this method provides interesting guidelines for the analysis of a FKBC. Non-operative rules (non-fired rules in a given mode of operation or working conditions) such as shown in Fig. 8.5b can be easily modified. In Fig.8.5b top left corner, only the rules centered around the axis x_1 are fired. A similar situation, this time around the axis x_2, is presented in the top right corner. In the bottom left corner of this figure only the rule covering the origin of the control space is fired. Finally, in the bottom right corner the set of rules does not cover all system trajectories. This dynamic behavior suggests.

Fuzzy sets of rules

x_2 \ x_1	PNB	PN	PZ	PP	PPB
VNB	APVB	APM	APM	APM	APVB
VN	APB	AP	APS	APS	APM
VZ	AP	APVS	AZ	ANVS	AN
VP	ANM	ANS	ANS	ANB	ANB
VPB	ANVB	ANB	ANM	ANM	ANVB

Fig.8.4. The variables x_1 and x_2 can for example represent position and velocity.

The rule base is in a table form. PNB, PN, PZ, PP and PPB represent linguistic values of x_1, such as negative big, negative, zero, positive, and positive big. VBN, VN, VZ, VP and VPB represent linguistic values of x_2, such as negative big, negative, Zero, positive, positive big. ANVB, ANB, AN, ANM, ANS, AZ, APS, APM, AP, APB and APVB represent linguistic values of the control output such as negative very big, negative big, negative, negative medium, negative small, zero, positive small, positive medium, positive, positive big, and positive very big. The membership functions for these linguistic values are defined in Fig. 8.5.

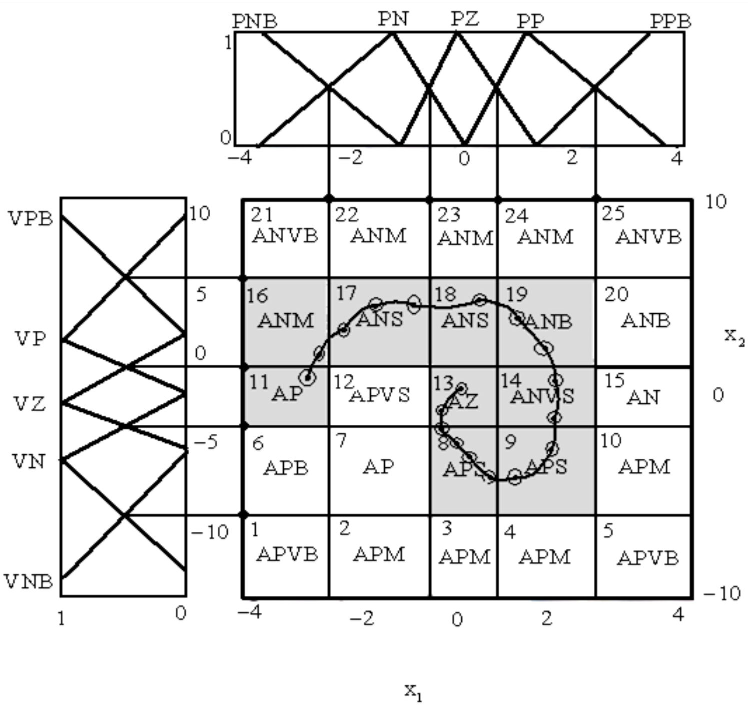


Fig.8.5. (a) The system trajectory mapped on the partitio

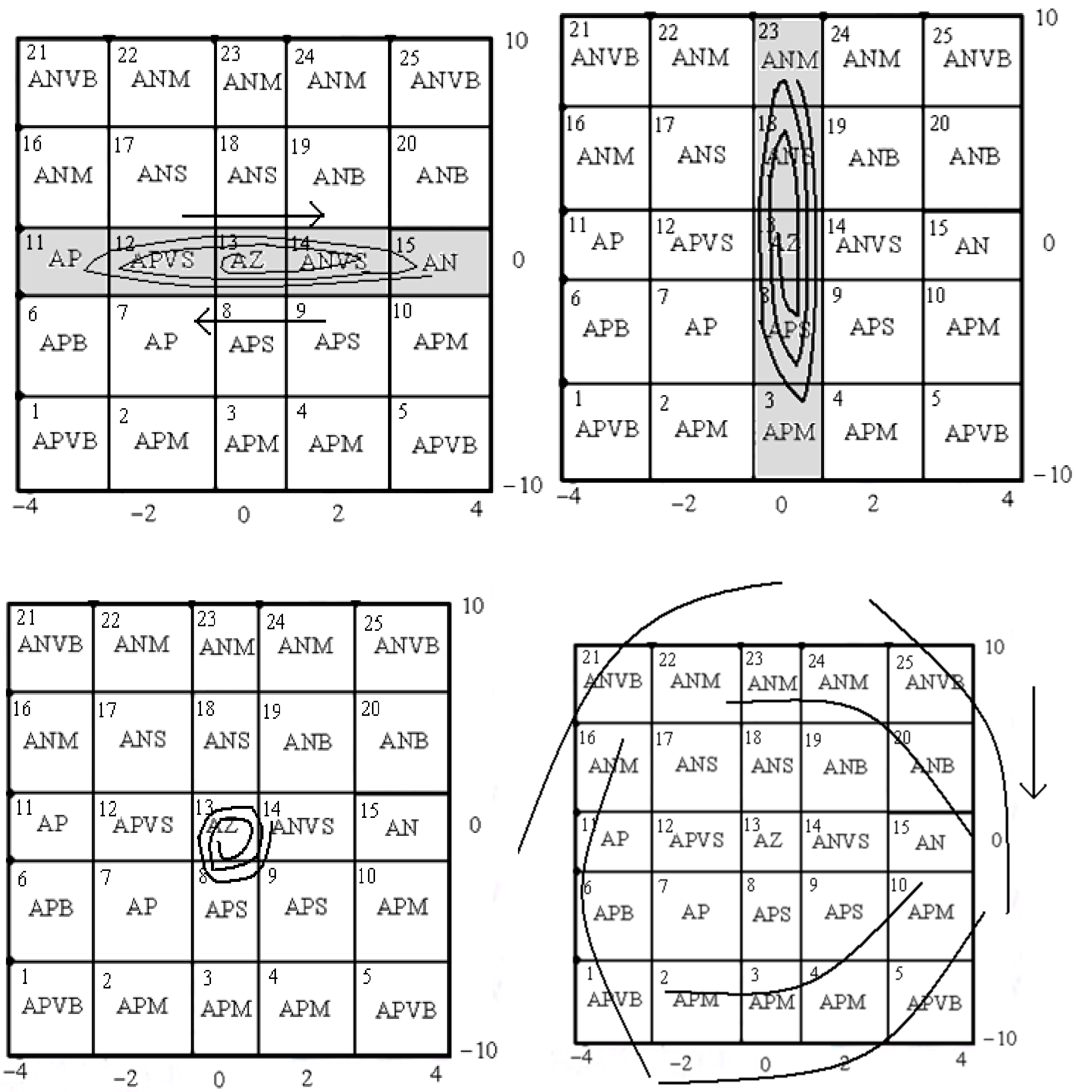


Fig.8.5. (b) Inadequate coverage of the partition space during the operation of a FKBC.

Gray areas correspond to fired rules, white areas correspond to non-fired rules and modifications in the fuzzy sets representing the meaning of the linguistic values of x_1 and x_2. Aracil et al. [1982] propose a geometric interpretation of the state map. This technique is based on the study of vector fields associated with the plant and the FKBC rule base.

Let us consider the closed-loop system from Fig.8.1 represented as

$$\frac{dx}{dt} = f(x) + b\mu,$$

$$\mu = \Phi(x), \qquad \text{…. (8.4)}$$

where f(x) is a nonlinear function which represents the plant dynamic with f (0) = 0, x and b are vectors of dimension n, μ is the scalar control variable and Φ(x) is a nonlinear function representing the FKBC with Φ(0) = 0.

Let μR (**x**, μ) be the fuzzy relation representing the meaning of the rule base let, defuzz be any defuzzification operator and let u^*_x be the fuzzified input of the controller obtained from the crisp input **x***; then closed loop behavior will depend on the nature of (x) and

$$\Phi(x) = \text{Defuzz}(\mu *_x {}^o\mu R(x, \mu)), \qquad \ldots(8.5)$$

where $x \in X_1 \times X_2 \times \ldots \times X_n$ and $u \in \mu$.

The direction of the vector field associated with the FKBC is determined by coefficients of b, and the magnitude is given by b. $\Phi(x_1)$ as shown in Fig. 8.6. One should note the effect which the subspace determined by the condition Φ(x) = 0 has on the closed loop. This subspace is usually a line (switching line) for second order System determines the separation of the state space in positive and negative control action areas (see Fig.8.7).

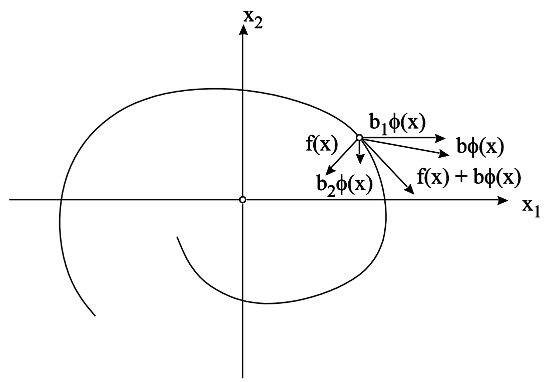


Fig.8.6. The components of the vector field.

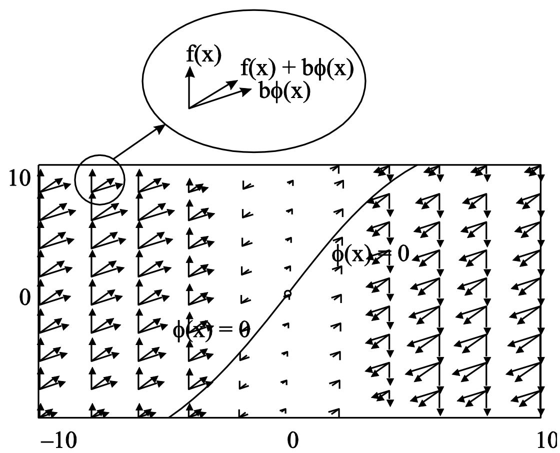


Fig. 8.7. vector field representation.

With monotone f(x) and Φ(x), state behavior of the closed-loop system can be predicted under the following conditions;

1. The open-loop system $\frac{dx}{dt}$ = f(x) is stable.
2. The vector field associated with b. Φ(x) = 0 tends towards Φ(x) = 0.

Obviously, for other situations such as unstable open-loop systems, the characteristics of the vector field must be determined to properly predict the behavior of the system.

However, the above analysis is in line with the previous concepts when considering linguistic trajectories. A simple inspection using this approach suffices to establish stability or instability, in addition to predicting other dynamic phenomena such as limit cycles, isolated areas, oscillations, etc.

To illustrate some of these dynamic phenomena, we can summarize as follows:

(a) Stability feedback system: In this case, the vector field obtained as $\Phi(x_1, x_2)$ tries to lead the system trajectories in the direction of the switching curve $\Phi(x_1, x_2)$ =0. When the trajectories approach this curve, $\Phi(x_1, x_2) = 0$, the plant component of the vector field obtains a greater influence which makes the trajectories converge to the equilibrium point (see Fig.8.8).

(b) Critical case: the nonlinear field does not lead the trajectories to the curve $\Phi(x_1, x_2) = 0$, so instability can be avoided if the plant component obtains greater influence then the controller component (see Fig. 8.9).

(c) Limit cycles: This case can be considered as a combination of the stable and critical case (see Fig.8.1.). Limit cycle also depend on the plant component characteristics.

(d) Nonlinear vector fields with isolated areas: Isolated areas are regions which behave differently from the dominated area, which is above the switching curve $\Phi(x_1, x_2) = 0$. Isolated areas are normally defined by closed generic curve $\Phi(x_1, x_2) = 0$, that do not contain the plant component equilibrium point, and do not produce field compensation. In this case, the tendency would be to go around these areas. Thus, an output trajectory appears where a large number of trajectories converge (see Fig.8.11).

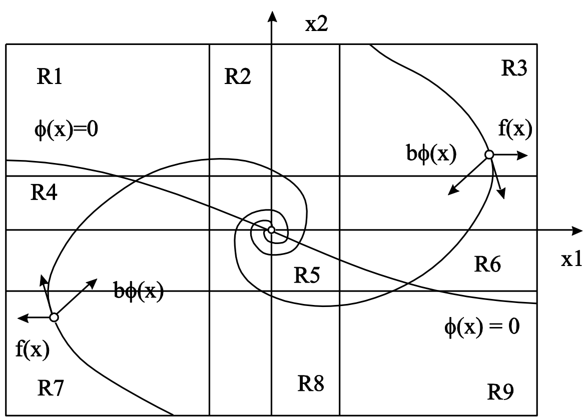


Fig.8.8. Stable feedback system.

R1: If x_1 is N and x_2 is P then u is NB, R2: If x_1 is Z and x_2 is P then u is NM
R3: If x_1 is P and x_2 is P then u is NS R4: If x_1 is N and x_2 is Z then u is PM
R5: If x_1 is Z and x_2 is Z then u is Z, R6: If x_1 is P and x_2 is Z then u is NM
R7: If x_1 is N and x_2 is N then u is PS, R8: If x_1 is Z and x_2 is N them u is PM
R9: If x_1 is P and x_2 is N then u is PB

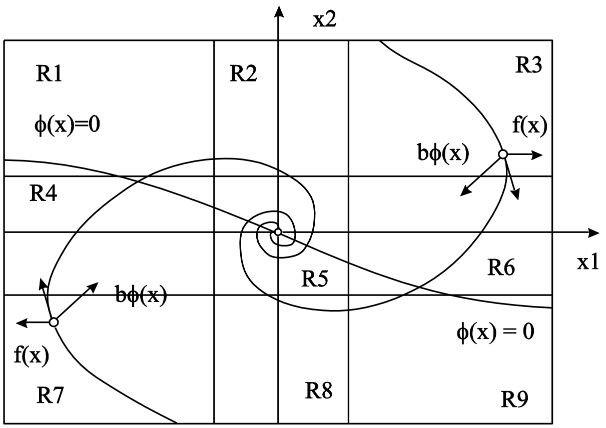


Fig 8.9 Critical Case

R1: If x_1 is N and x_2 is P then u is PB
R2: If $x_1$is Z and x_2 is P then u is PM
R3: If x_1 is P and x_2 is P then u is PS
R4: If x_1 is N and x_2 is Z then u is PM
R5: If x_1 is Z and x_2 is Z then u is Z
R6: If x_1 is P and x_2 is Z then u is NS
R7: If x_1 is N and x_2 is N then u is NS
R8: If x_1 is Z and x_2 is N them u is NM
R9: If x_1 is P and x_2 is N then u is NB
NB, NM, N, NS, Z, PS, P, PM, PB represents negative big negative medium negative small zero positive small positive positive medium and positive big respectively

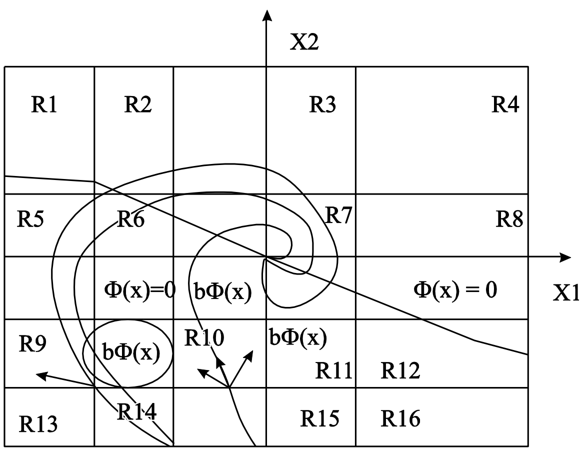


Fig. 8.10 presence of a limit cycle

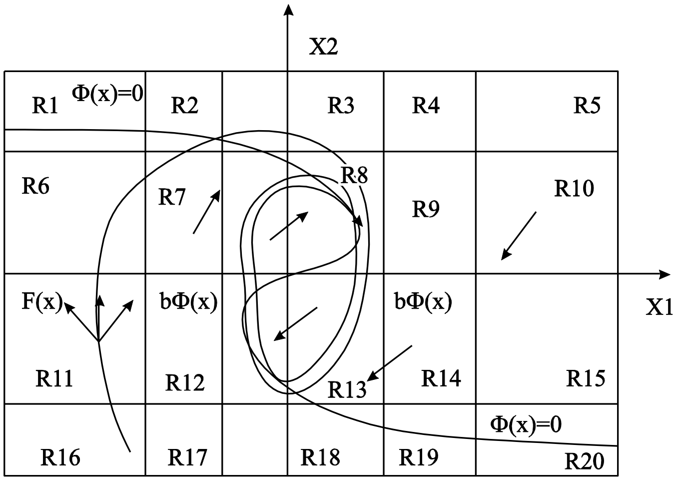


Fig.8.11. zNonlinear dynamic with isolated areas.

Rules for presence of limit

R1: If x_1 is NB and x_2 is PB then u is PS
R2: IF x_1 is N and x_2 is PB then u is PS
R3 : IF x_1 is Z and x_2 is PB then u is PM
R4 : If x_1 is P and $x_2$is PB then u is PM
R5: If x_1 is PB and x_2 is PB then u is PB
R6: IF x_1 is NB and x_2 is P then u is NM
R7: IF x_1 is N and x_2 is P then u is NM
R8: IF x_1 is Z and x_2 is P then u is PS
R9: If x_1 is P and $x_2$is P then u is PM
R10: If x_1 is PB and x_2 is P then u is PM
R11: IF x_1 NB and x_2 is N then u is NM
R12: If x_1 is N and x_2 is N then u is NM
R13 : IF x_1 is Z and x_2 is N then u is NS
R14: IF x_1 is P and x_2 is N then u is PM
R15 : IF x_1 is PB and x_2 is N then u is PM
R16: If x_1 is NB and $x_2$is NB then u is NB
R17: If x_1 is N and $x_2$is NB then u is NM
R18: If x_1 is Z and x_2 is NB then u is NM
R19: If x_1 is P and x_2 is NB then u is NS
R20: If x_1 is PB and x2 is N then u is NS

Rules for Nonlinear dynamic with isolated areas.

R1: If x_1 is NB and x_2 is P then u is PB

R2: IF x_1 is N and x_2 is P then u is PB

R3 : IF x_1 is Z and x_2 is P then u is PM

R4 : If x_1 is P and x_2 is P then u is PS

R5: If x_1 is NB and x2 is Z then u is NM

R6: IF x_1 is N x_2 is Z then u is NM

R7: IF $x_1$is Z and x_2 is Z then u is Z

R8: IF x_1 is P and x_2 is Z then u is PM

R7: IF x_1NB and x_2 is N then u is NM

R7: If x_1 is N and x_2 is N then u is PM

R8 : IF x_1 is Z and x_2 is N then u is NM

R9: IF x_1 is P and x_2 is N then u is NS

R7 : IF x_1 is NB and x2 is N then u is NB

R7: If $x_1$is N and x_2 is NB then u is NB

R8: If x_1 is Z and x_2 is NB then u is NM

R9: If x_1 is P and x_2 is NB then u is NM

NB, NM, N, NS, Z, PS, P, PM , PB represents negative big, negative medium, negative small, zero, positive small, positive, positive medium, and positive big respectively

The above conclusions about stability are qualitative, but enough to characterize the dynamic behavior of a feedback system, or serve as the bases for adequate selection or modification of rules. For example, the limit cycle in Fig.8.10 could be avoided by modifying the rule-consequents of the rules R8 and R13 in order to generate a control output with the opposite sign. Thus "μ is PS" in rule R8 should become "μ is NS" and "μ is NS" in rule R13 should become "μ is PS" the same can be done with the rule R10 in order to eliminate the isolated area shown in Fig. 8.11.

8.3 STABILITY AND ROBUSTNESS INDICES

In the previous section we have presented approximation of the problem of FKBC stability, and its geometrical interpretation in the state space. In this section we formalize the problem of stability analysis by using concept extracted from the qualitative theory of nonlinear dynamic system [Guckemheimer, J. et al. (1983)]. These concepts are used to interpret instability and bifurcations and thus, a more complete representation of the stability problem is achieved. We start with the one-dimensional case followed by the n-dimensional case.

8.3.1 The one-dimensional case

Let us consider a system with a mathematical model of the following from

$$\frac{dx}{dt} = f(x) + b.\ \mu, \qquad \text{with} \qquad \ldots(8.6)$$

$x \in X \subset IR$ and $\mu \in U \subset IR$, where f(x) is a nonlinear, monotone and increasing function such that f(0) = 0,and IR is the real line. Consider also the control law given by

$$\mu = \Phi(x), \qquad \ldots.(8.7)$$

where Φ(x) is a nonlinear function representing the FKBC and Φ(0) = 0.

Without loss of generality, we will consider the case b = 1. As we noted in section 8.2, the vector field associated with the closed-loop dynamical system (8.6) and (8.7) is

$$\frac{dx}{dt} = f(x) + \Phi(x) \qquad \ldots\ (8.8)$$

It is composed of two parts: the plant component f(x) and the controller component Φ(x). to visualize geometrically these vector field we can rotate them counterclockwise by an angle of 90^o and obtain the curves shown in Fig. 8.12.

The equilibrium point of system (8.8) are given by the values of x for which

$$\frac{dx}{dt} = 0. \qquad \ldots\ (8.9)$$

That is,

$$f(x) + \Phi(x) = 0 \text{ or } \Phi(x) = -\ f(x). \qquad \ldots\ (8.10)$$

With the graphical convention of rotation the vector field, the equilibria will be located at values of x where the curve representing the controller component Φ(x) intersects the curve representing the plant component with the sign changed to –f(x).

It is clear that system (8.8) has an equilibrium point at the origin as f(0) = 0 and Φ = 0. For this equilibrium to be stable it is necessary that

$$f'(0) = \Phi'(0) < 0. \qquad \ldots\ (8.11)$$

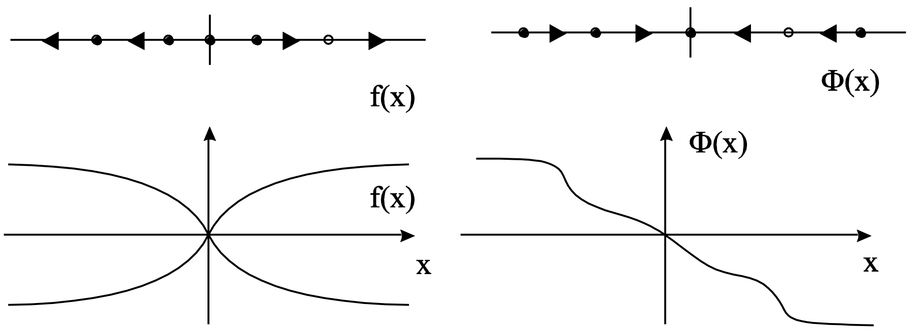


Fig 8.12. vector fields in the one-dimensional case.

That is, the eigenvalue at the origin has a negative real part.
Therefore, the system (8.8) is globally stable if it meets the following conditions

Condition 1 : $\Phi'(0) < - f'(0)$

Condition 2 : $I\Phi'(x) < If'(x)\ I\forall\ x \neq 0.$

Condition 1 ensures the equilibrium stability at the origin, while condition 2 prevents the appearance of other equilibria (no intersection of curves $\Phi(x)$ and $-f(x)$ if this condition is satisfied).
Not meeting condition 2 for some value of x produces intersection of curves $\Phi(x)$ and $-f(x)$ at point other than the origin. This is shown in Fig. 8.13 where new equilibrium point appear.
To see how these new aquilibrium points are generated, assume that some "deformation" is produced in $\Phi(x)$, and/or $f(x)$, so that an originally stable situation is changed. The question then is: under what circumstances will stability be lost? It is well known that loss of global stability may occur in at least one of the following cases:

1. The equilibrium at the origin becomes unstable.
2. New bifurcations are produced (intersections of $\Phi(x)$ with $-f(x)$).

It is possible to give a measure of how for a system is from loss of stability (how large are the "deformations" the system can be exposed to without losing stability). With this measure we define some stability indices.
To that end, let us reconsider condition 1. This condition can be rewritten

$$\Phi'(0) + f'(0) < 0 \qquad \dots (8.12)$$

or

$$-(\Phi'(0) + f'(0)) > 0. \qquad \dots (8.13)$$

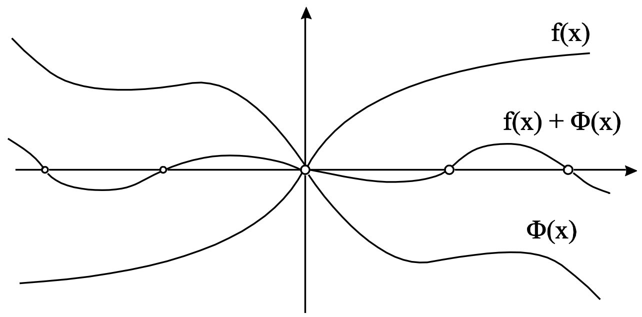


Fig. 8.13 Appearance of multiple attractors

Then the quantity $-(\Phi'(0) + f'(0))$ gives a measure of the robustness of the system against loss of stability at the origin. The higher the value of $-(\Phi'(0) + f'(0))$ the further away the system is from an unstable condition. This unstable condition is reached for $-(\Phi'(0) + f'(0)) = 0$.

In a similar way, a measure can be associated with condition 2 giving the minimum distance between $\Phi(x)$ and $-f(x)$. The greater this distance, the more robust the system is against "deformation" of $\Phi(x)$ and $-f(x)$. A natural candidate for this minimum distance is given by

$$\min |\Phi(x) + f(x)|. \qquad \text{… (8.14)}$$

However , $|\Phi(x) + f(x)|$ will take its minimum value at the origin, where it is 0. To avoid this, a certain region around the origin should be avoided. The boundary values of this region are B_1 and B_2. These are defined as the closest values to the origin such that $\Phi'(B_1) = - f'(B_1)$ and $\Phi'(B_2) = - f'(B_2)$. Figure 8.14 shows the geometrical interpretation of B_1 and B_2.

It is interesting to see what happens in the second case when new bifurcations are produced. Assume a stable situation (on intersection of $\Phi(x)$ with $-f(x)$). As $\Phi(x)$ and/or $-f(x)$ are "deformed" to approach each other, the "contact" between them will be realized when they acquire a common tangent. Then the contact is reduced to a single point and a bifurcation is produced.

As the deformation is increased two crossing points will appear. Under normal circumstances, one of these crossing points will give rise to a stable equilibrium point, and the other to an unstable one. This new equilibrium point will destroy the global stability at the origin as two attraction areas appear in the state space. In other words, the origin will no longer be the only attractor of the system.

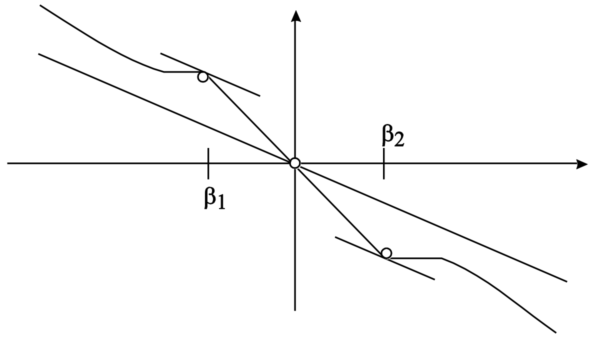


Fig. 8.14 Graphical interpretation of the region B = (β_1, β_2).

Summarizing the above discussion, two indices can be defined

$$I_1 = -(\Phi(0) + f(0)),$$

$$I_2 = -\frac{\min}{b'} I\Phi(x) + f(x)I, \qquad \dots (8.15)$$

where B′ is the complement of the region around the origin B = **(β_1, β_2)**.

8.3.2 The N-dimensional case

Let us start with n = 2. In this case the system has the form

$$\frac{dx}{dt_I} = f(x) + b^{\cdot} \mu$$

$$\mu = \Phi(x) \text{ with} \qquad \dots (8.16)$$

$x \in X \subset IR^2$ and $u \in U \subset IR$, where IR^2 is the real plane. This equation can be written in a more detailed form as

$$\frac{dx1}{dt} = f_1(x_1,x_2) - b_1 \cdot \Phi(x_1,x_2),$$

$$\frac{dx2}{dt} = f_2(x_1,x_2) - b_2 \cdot \Phi(x_1,x_2), \qquad \dots (8.17)$$

where $f_1(0, 0) = f_2(0, 0) = \Phi(0, 0) = 0$ and f_1 and f_2 are monotone functions. As in the one-dimensional case $\Phi(x_1,x_2)$ stands for the FKBC. Assume that the equilibrium at the origin is stable, i.e., the two eigienvalues of the linearized system around the origin have a negative real part.

1. The real eigenvalue crosses the imaginary axis and acquires a positive sign. This bifurcation from a stable node to a saddle point is also called static bifurcation.

A pair of complex poles crosses the imaginary axis both poles take positive real values. This bifurcation is called Hopf bifurcation. Figure 8.15 describe these two situations.

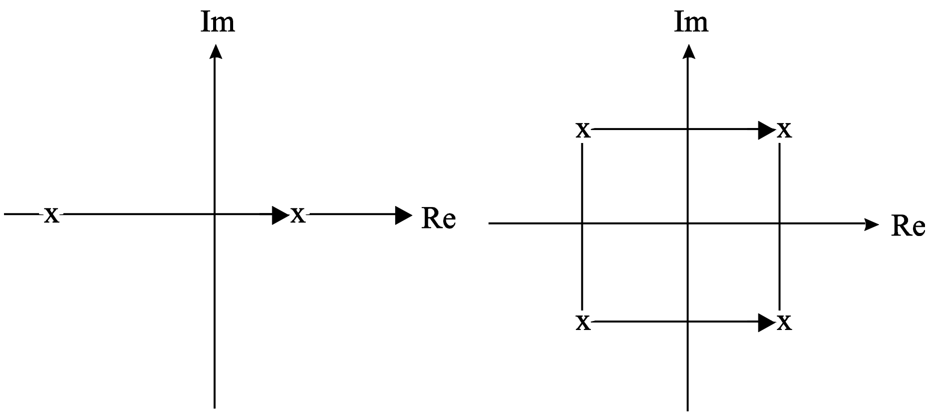


Fig.8.15 (a) Static bifurcation; (b) Hopf bifurcation.

The simplest way to linearize a nonlinear system around a point x_0 in its state space is to consider the series development

$$f(x) = f(x_0) + j(x - x_0) + \dots, \qquad \dots(8.18)$$

where j is the so-called Jacobin matrix of f(x) at x_0. This matrix is given by

$$J = \begin{vmatrix} \frac{df1}{dx1} & \frac{df2}{dx1} \\ \frac{df2}{dx1} & \frac{df1}{dx2} \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \qquad \dots(8.19)$$

If the linearized system is stable, the Lyapunov criterion guarantess stability at the origin of the nonlinear model.

The characteristic polynomial of J is defined as

$$P(s) = \det(s \cdot I - J). \qquad \dots (8.20)$$

It is easy to see that

$$\det(s \cdot I - J) = s^2 - s \cdot (a_{11} + a_{22} - a_{12} \cdot a_{21} \cdot) \qquad \dots (8.21)$$

The characteristic polynomial J can be expressed as

$$P(s) = \det(s \cdot I - J)\ s^2 + a_1 \cdot s + a_2, \qquad \dots (8.22)$$

where

$a_1 = -(a_{11} + a_{22}) = -\operatorname{tr}(J)$,

$$a_2 = a_{11} \cdot a_{22} - a_{12} \cdot a_{21} = \det(J). \qquad \dots (8.23)$$

A static bifurcation is produced when a real zeal zero (eigenvalue) of the characteristic polynornial crosses the imaginary axis (see Fig.6.15a), i.e., when one of the roots of p(s) is s=0.This only will happen in equation (8.22) when $a_2 = 0$. So this is the condition for loss of stability by a static bifurcation in a second order dynamic system. It should be noted that the larger the value of a_2 the further the system is from the bifurcation point. Therefore , the system is more robust, then the value of a_2 gives a measure of the degree of relative stability and it is quite natural to define a stability index such as

$$I_1 = a_2 = \det(J). \quad \ldots (8.24)$$

A Hopf bifurcation is produced when two complex eigenvalues cross the imaginary axis and their real parts become positive (see Fig. 6.15b). In the case of our second order system, this means that the two complex eigenvalues of the system have zero real part. The requirement for this is that $a_1 = 0$ in the characteristic polynomial. Then the condition to have a Hopf bifurcation reduces to $a_1 = -\operatorname{tr}(j) = 0$. A value of a_1 far from 0 is a guarantee for not having a Hopf bifurcation. In this way we define a second stability index as

$$I'_1 = -\operatorname{tr}(J). \quad \ldots (8.25)$$

To summarize, a statics bifurcation is be produced when $a_2 = 0$ and a Hopf bifurcation when $a_1 = 0$. The relative stability of the equilibrium point at the origin can be measured by the two indices

$$I_1 = \det(J) \text{ and } I'_1 = -\operatorname{tr}(j). \quad \ldots (8.26)$$

Let us now generalize the index I_2.Remember that in the one-dimensional case a bifurcation occurs when the vector field of the FKBC exactly compensates the vector field of the plant, thus giving rise to a global vector field of value zero. In the two-dimensional case, the vector field of the controller has, at every point of the state space, the direction given by (b1, b2). Therefore, the compensation of the vector components of the plant and of the controller can only occur in the region of the state space where the plant component has the direction (b1-b2). So we can define the auxiliary subspace as

$$\frac{F1(x1,x2)}{b1} = \frac{F2\,(x1,x2)}{b2} \quad \ldots. (8.27)$$

This expression represents the one-dimensional subspace of the state space, where the study of the occurrence of static bifurcations is to be performed. Thus the analysis is similar to the case of $n = 1$. Then an index I_2 can be defined as the minimum distance between the plant and controller components,

Calculated on the auxiliary subspace and excluding the region B around the origin:

$$I_2 = \min \mid f(x) + b.\Phi(x) \mid. \quad \ldots (8.28)$$

For $n > 2$, the generalization is straightforward. Let j be the Jacobian of the nonlinear system around the origin and let,

$$P(s) = s^n + a1.\, s^{n-1} + \ldots + a_{n-1}.s + a_n \quad \ldots (8.29)$$

be its corresponding characteristic polynomial. The generalization of the index I1 leads to

$$I'_1 = a_n = (-1)^n.\, \det(J). \quad \ldots (8.30)$$

The index I'_1 is determined using the condition for aHopf bifurcation. This condition is equivalent to the one for having two pure imaginary axes. The characteristic polynomial (8.29) can be rewritten in the form.

$$P(s) = p_1 (s) . (w^2 + s^2) + b_1 . s + b_2 \quad \text{...(8.31)}$$

Then the condition for having two pure imaginary axes is

$$b_1 = b_2 = 0 \quad \text{... (8.32)}$$

For example, consider the case n = 3. The corresponding characteristic polynomial is

$$P(s) = s^3 + a_1 . s^2 + a_2 . s + a_3 \quad \text{.... (8.33)}$$

(8.33) can be rewritten in the form of equation (8.31) as

$$P(s) = (s + a_1) (w^2 + s^2) + (a_2 - w^2) s + a_3 – a_1 . w^2. \quad \text{... (8.34)}$$

Therefore $b_1 = a_2 – w^2$ and $b_2 = a_3 – a_1 w^2$. The Hopf condition (8.32) reduces in this case to

$a_2 - w^2 = 0$,

$$a_3 – a_1 . w^2 = 0. \quad \text{... (8.35)}$$

That is, $a_1 .a_2 - a_3 = 0$. Then we can take I'_1 as

$$I'_1 = a_1.a_2 - a_3 \quad \text{... (8.36)}$$

Thus , I'_1 is the index for the measure of the “distance” between the complex poles having a negative real part and crossing the imaginary axis at points where instability occurs.

It can be demonstrated [ref. Article Ollero et al. (1992)] that the Hopf condition is equivalent to

$$\det(H_n - 1) = 0, \quad \text{... (8.37)}$$

where H_{n-1} is the minor principal of order n -1 of the Hurwitz matrix H. matrices H and H_n-1 are given as follows

$$H = \begin{bmatrix} a1 & a3 & a5 \ldots \\ 1 & a2 & a4 \ldots \\ 0 & a1 & a3 \ldots \end{bmatrix} ; H_{n-1} \begin{bmatrix} a1 & a3 & a5 \ldots \\ 1 & a2 & a4 \ldots \\ 0 & a1 & a3 \ldots \end{bmatrix} \quad \text{...(8.38)}$$

Consequently,

$$I'_1 = \det(H_{n-1}). \quad \text{...(8.39)}$$

It can also be shown that, [ref. Article Ollero et al. (1992)]

$$I_1 . I'_1 = \det(H). \quad \text{...(8.40)}$$

The generalization of the index I_2 for the case $n > 2$ is straightforward. In this case the one-dimensional auxiliary subspace will be given by

$$\frac{f1(x1, x2, \ldots, xn)}{b1} = \frac{f2((x1, x2, \ldots, xn)}{b2} = \cdots \frac{fn((x1, x2, \ldots, xn)}{bn} \quad \text{...(8.41)}$$

The remarkable fact is that for every n, that auxiliary subspace is one-dimensional (if the dimension of ʮ is one).

Example 8.1 Let us consider a nonlinear system described by the following equation

$$\frac{d^2y}{(dt)^2} + a_1.(1 - y^2)\frac{dy}{dt} + a_2.y = b_1, u \quad \ldots(8.42)$$

This system can be represented in the state space by means of

$$\dot{x_1} = x_2 \text{ and } \quad \dot{x_2} = - a_2.x_1 - a_1.x_2 + a_1.\ x^2_1.\ x_2 + b_1.u, \quad \ldots(8.43)$$

where $x_1 = y$, $b_1 = a_2 = 12.7388$, $a_1 = 2.2165$, and $|u| \le 15$.

For u = 0, The System Has An Unstable Limit Cycle (FIG. 8.16). This corresponds to a system with a hypercritical Hopf bifurcation. Considering the saturation in u, a FKBC can be designed to increase the stability domain (limited by the unstable limit cycle) as much as possible and to improve the dynamic characteristics of this system in the stable domain. Thus a robust design is obtained.

Assume we have a set of FKBC rules provided by an expert, and given in a table from as in Fig. 8.17. Additionally, Fig. 8.18 represents the FKBC as a nonlinear function Φ (**x**).

The Jacobian of the feedback system is given by

$$J_c = \begin{bmatrix} 0 & 1 \\ -a_2 + b_1.\frac{\partial\emptyset(x_1, x_2)}{\partial x_1} & -a_1 + b_1.\frac{\partial\emptyset(x_1, x_2)}{\partial x_1} \end{bmatrix} \quad \ldots(8.44)$$

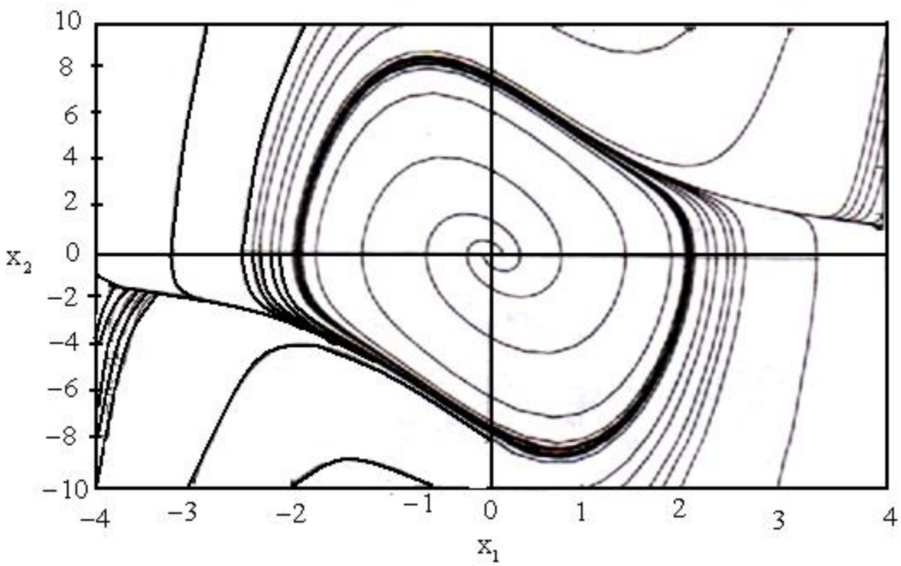


Fig. 8.16. open system response. The system has an unstable limit cycle.

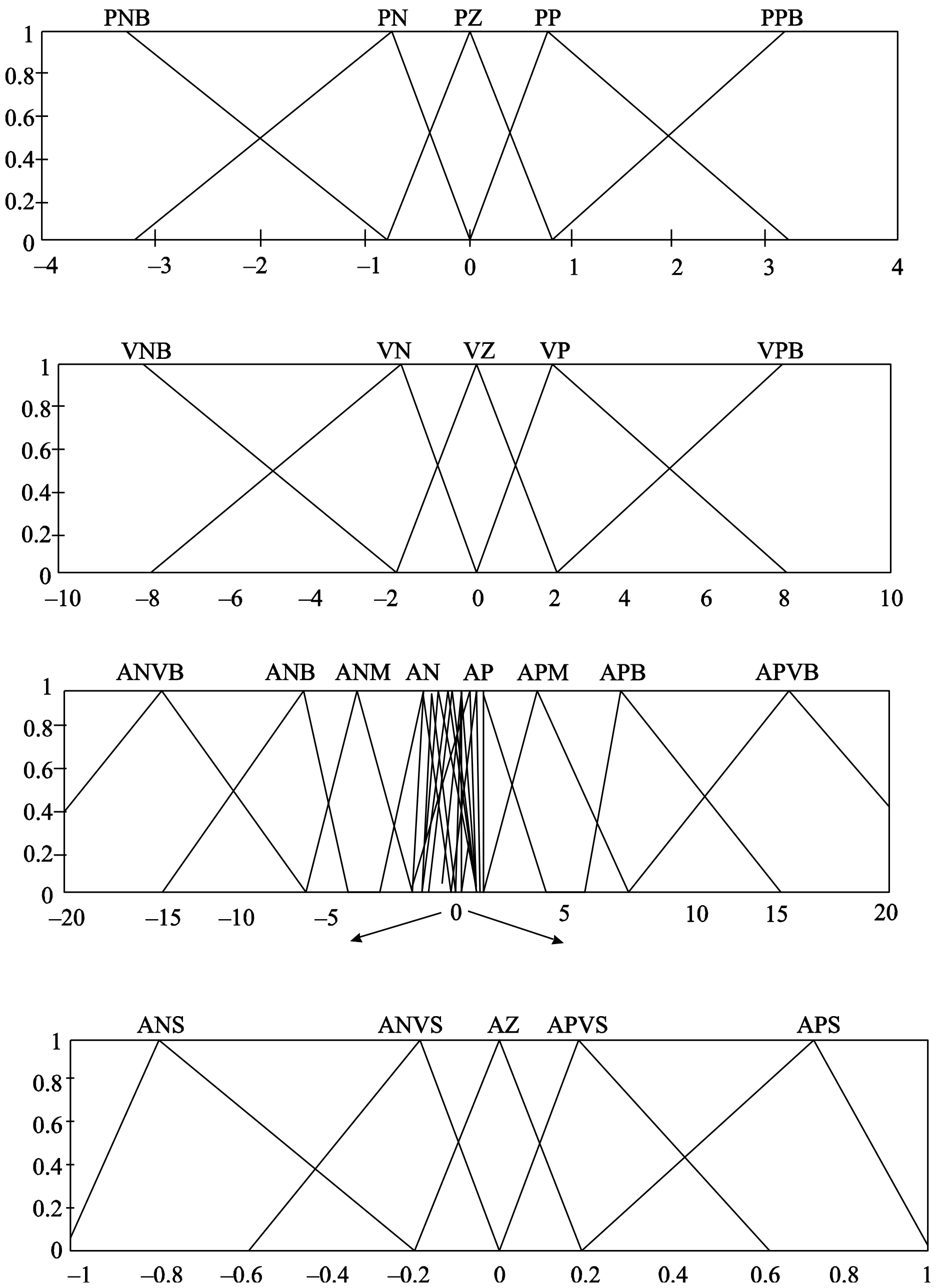


Fig. 8.17. (a) Fuzzy sets representing the meaning of linguistic values.

In (i) PNB, PN, PZ, PP and PPB represent linguistic values of x_1, such as negative big, negative, zero, positive, and positive big.

In (ii) VBN, VN, VZ, VP and VPB represent linguistic values of x_2, such as negative big, negative, zero, positive, and positive big.

X_2 \ X_1	PNB	PN	PZ	PP	PPB
VNB	1 APVB	2 APM	3 APM	4 APM	5 APVB
VN	6 APB	7 AP	8 APS	9 APS	10 APM
VZ	11 AP	12 APVS	13 AZ	14 ANVS	15 AN
VP	16 ANM	17 ANS	18 ANS	19 AN	20 ANB
VPB	21 ANVB	22 ANM	23 ANM	24 ANM	25 ANVB

Fig. 8.17 (b)

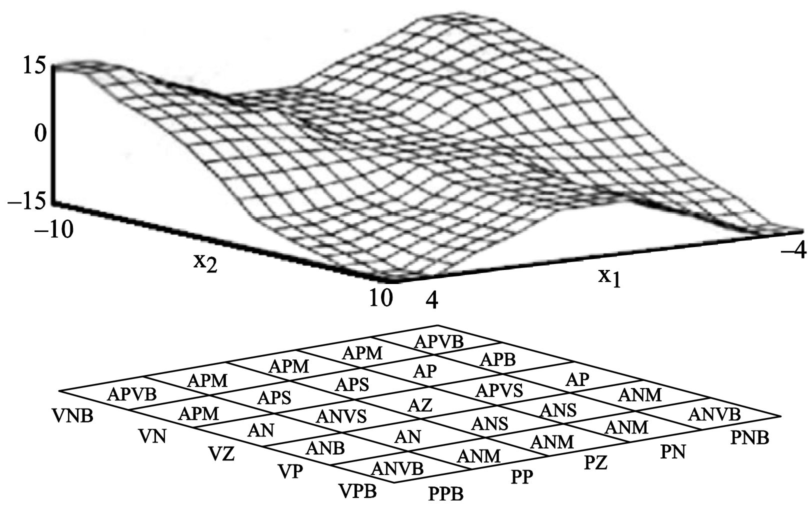


Fig. 8.18

In (iii) ANVB,ANB,AN,ANM,ANS,AZ,APS<APM,AP,APB and APVB represent linguistic values of the control output such as negative very big. Negative big, negative, negative medium, negative small, zero, positive small, positive medium, positive, positive big, and positive very big. and (iv) is a magnification of the control action interval (-1,1).

Fig. 8.17 (b) The rule base in a table form. The variables X_1 and X_2 can for example represent position and velocity.

Fig. 8.18. the nonlinear function Φ (x) representing the FKBC.

The derivatives Φ_{x1} and Φ_{x2} at the origin can be obtained by using interpolation techniques. For the case of linear interpolation and the values of a_1,a_2 and b_1 given as $b_1 = a_2 = 12.7388$, $a_1 = 2.2165$, we obtain

$$J_c = \begin{bmatrix} 0 & 1 \\ -15.9236 & -6.4204 \end{bmatrix} \qquad \text{...(8.45)}$$

Then the corresponding characteristic polynomial is given by

$$P(s) = s^2 + 6.4024.s + 15.9236. \qquad \text{... (8.46)}$$

Using equation (8.26) the indices can be obtained directly from the characteristic polynomial

$$I'_1 = 6.4024; \quad I'_1 = 15.9236. \qquad \text{...(8.47)}$$

The auxiliary subspace is determined by

$$x_2 = 0 \qquad \text{...(8.48)}$$

Figure 8.19 present the contributions of f(x) and Φ (x) on the auxiliary subspace. As shown, the index I2 is not defined, providing excellent robustness in the face of new attractors.

The system in the stable domain is more robust than the original one. Figure 6.20. gives a state diagram of the closed-loop system. As shown, the stability domain is considerably increased. Finally, Figs. 8.21 and 8.22 compare the transient responses of the closed-loop system with that of the open-loop system. A significant improvement in system behavior is also apparent.

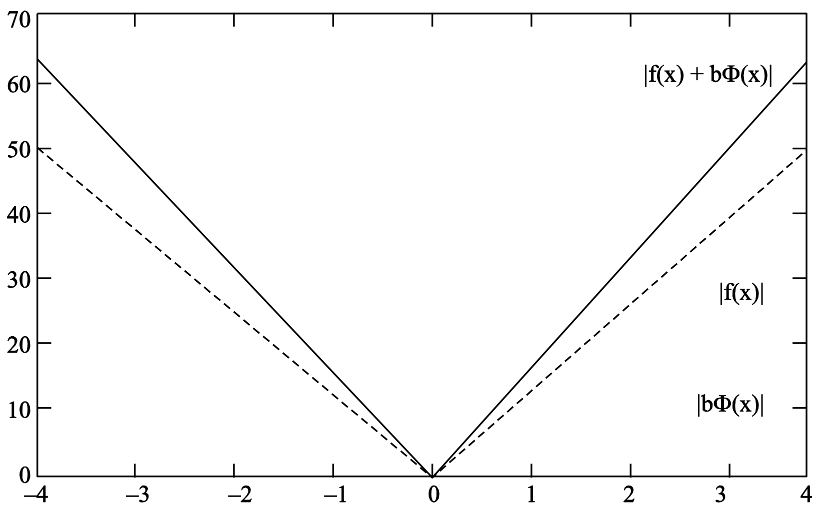


Fig. 8.19. open-loop system response.

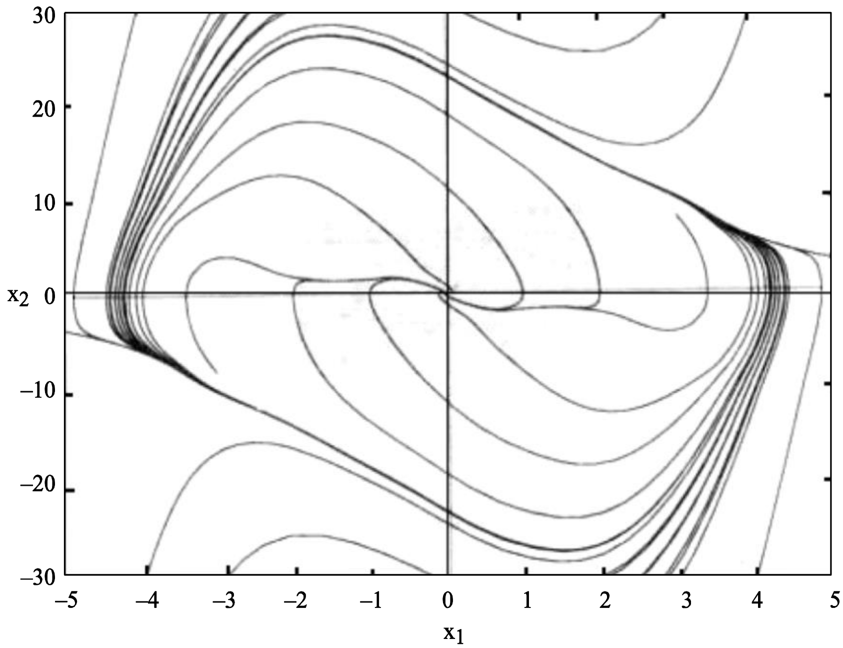


Fig. 8.20. Closed-loop system response.

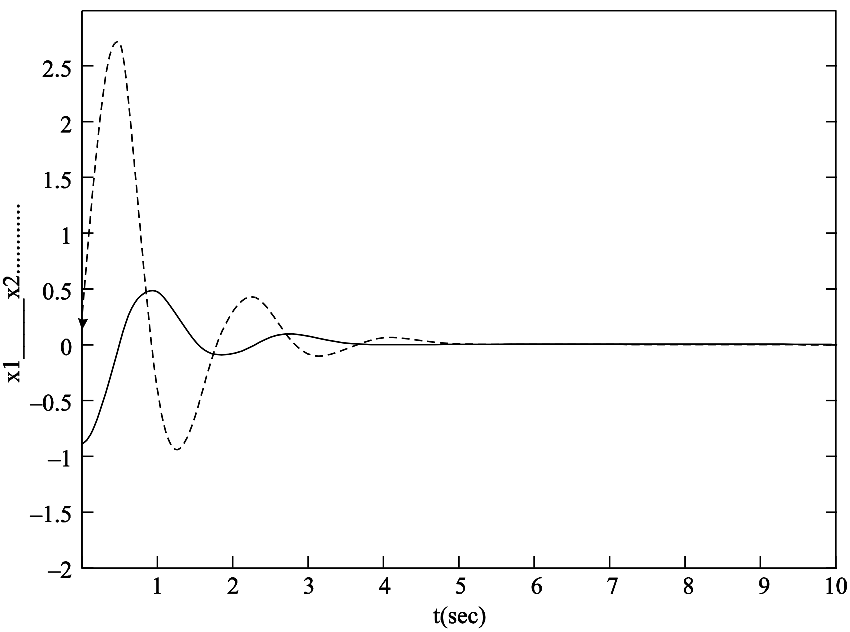


Fig. 8.21. open-loop system response.

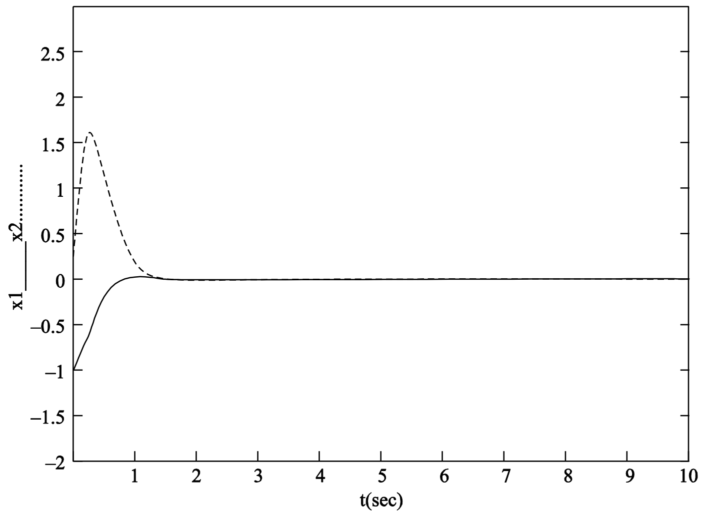


Fig. 8.22. Closed-loop system response.

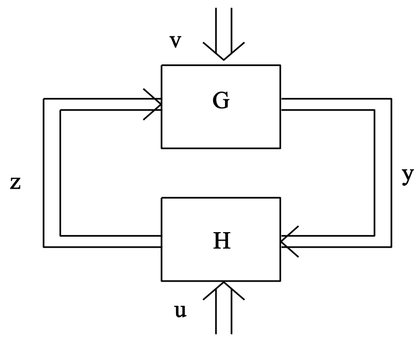


Fig. 8.23. A general feedback system.

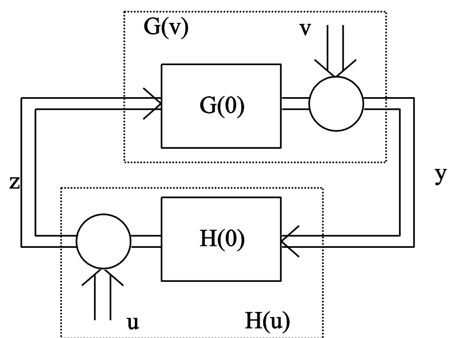


Fig. 8.24 Feedback system with additive inputs

8.4 INPUT-OUTPUT STABILITY

In this section we will present the basics of input-output stability. For a more detailed account see Safonov [1980] and Vidyasagar [1978].

Following Safonov [1980], we deal with a feedback system as shown in Fig. 8.23. Typically, G (v) represents the plant and H(u) the controller. The scalar or vector signal y will be the control variable and z the control output.

A particular case is that of Fig. 8.24, where u and v enter additively. In this case, u will be the reference variable and u may represent disturbances or initial conditions.

8.4.1 The Spaces of signals

In the input-output stability analysis, x(t) is a scalar or vector function of time, taken from a normed vector space x, i.e., a vector space with a norm on it. Two usually used norms are the 2-norm and the infinity-norm.

$$|x(t)|_{2} = \left(\int_0^\infty |x(t)|^2 dt\right)^{1/2} \quad \ldots (8.49)$$

$$|x(t)|_\infty = \text{ess sup } \{ |x(t)| : t \varepsilon [0,\infty]\} \quad \ldots (8.50)$$

where ess sup denotes the essential supremum.

We denote by L_2 and $L_{2\,\infty}$ the normed spaces of signals that have a finite 2-norm and infinity-norm, respectively. The space L_2 represents the set of all finite-energy signals, and the space $L_{2\,\infty}$ represents the set of all bounded signals.

The problem formulation must be able to cope with bounded signals usual in control:

(1) Steps with undefined 2-norm, and

(2) Ramps with undefined 2-and infinity-norms.

We define the extended space X_e in the following way. Consider the truncation $x_T(t)$ after time T, defined by: $x_T(t) = x(t)$ for $t \leq T$, and $xT(t) = 0$ for $t > T$. the extended space x_e is the space formed with all signals x(T) such that their corresponding truncation $x_T(t)$ belong to X. in this way, the extended normed spaces contain all physically conceivable signals.

8.4.2 The input-output Notion of a system

A system G that takes inputs X(t) and yields outputs Y(t) from the extended space X_e is considered as a relation G C X_e x X_e. This relation is normed by all pairs (x(t) , y(t)) such that y(t) is a possible output produced by the input x(t).

The difference with the usual notion of a system as an operator is that given an input signal X(t), an operator produces one and only one input, $Y(t) = G_x(t)$, while a replation may produce none, one, or more outputs. The advantage of using relation in this context is that one can not only cope with responses to different conditions, but can also separate the problem of existence and uniqueness of operators from the problem of stability.

8.4.3 The input-output Notion of stability

A system G is said to be finite-gain stable when the gain of G, g(G) is defined by

$$g\,(G) = \sup_{X(T)\neq 0}\left\{\frac{G_x(t))_t}{X_T(t)}\right\} < \infty. \qquad \ldots(8.51)$$

The intuitive idea behind that is that we can make the system output small provided that we make its inputs small enough. The above definition for an open-loop system G is translated to the feedback system of Fig. 8.24. Let us denote by F the closed-loop system, with closed-loop inputs (u-v) and closed-loop outputs (y, z). The intuitive idea of the closed-loop stability of F is that we can obtain small (y, z) provided that we make (u, v) small enough. For example, y may represent the error of the plant output with respect to the reference (set-point) v and u may represent disturbances to be avoided.

Other similar definitions of stability can be provided [for details see, Safonove (1980)]. The input-output stability does not clarify the asymptotic approach of the control variables to zero. This property can be ensured based on the above mentioned equivalence between input-output stability and Lyapunov internal stability. Vidyasagar (1986)].

The main result in input-output stability the small gain Theorem [in James (1966)] that states that a sufficient condition for the stability of the closed-loop system F (see Fig. 8.26) is g(G). g(H) < 1. From this theorem, one can derive the circle criterion and the conicity criterion and these can be directly applied to FKBC stability. These criteria will be considered in Sections 8.5 and 8.6.

Finally, the gain of a general system is a quantity that can be difficult to obtain. For two classes of systems it is easy to derive the gain: (i) nonlinear static (memory less) systems (NLS) and (ii) Linear time-invariant systems (LTI).

The gain of the NLS system defined by y = H (x) is given as

$$g\,(H) = \sup_{|x|\neq 0}\left\{\frac{H(x))_t}{|x|}\right\}, \qquad \ldots(8.52)$$

hwere the simplification is due to the fact that the time-dependence of the general formula disappears.

The gain of the LTI system G given by its frequency response G (jw) is given by

$$g(G) = \sup_w \bar{\sigma}\{G(jw)\} \qquad \ldots(8.53)$$

where $\bar{\sigma}(A)$ denotes the maximum singular value of the matrix A the singular values of A are the square roots of the eigenvalues of A*A, with `* denoting transpose conjugation. The two last formulas are directly applicable and extensively used in practical input-output stability analysis.

8.5 THE CRICLE CRITERION

Ray et al. [1984] proposed an analysis technique for FKBC stability based on the circle criterion. In the main study is based on considering the FKBC as a multi-level relay, as previously introduced by Kicker [1978]. Even although this appears restrictive, yet it should not be considered as such. In practice, the concepts of limiting nonlinear functions are used for every nonlinear function.

For the FKBC to behave as a multi-level relay, two conditions on the fuzzy sets describing the meaning of the linguistic values of control output u have to hold:

(a) The fuzzy sets have symmetric membership functions and
(b) The Middle-of-Maxima defuzzification method is used in obtaining the crisp value of u.

Figure 8.25 shows the input-output map of the multi-level relay FKBC.

8.5.1 Stability Analysis – SISO Case

Consider a SISO (single input single output) linear system represented by its transfer function g(s), which is rational and asymptotically stable (Fig. 8.26). The nonlinear function representing the FKBC is described using f(e), where e is the closed-loop error with the constraints

$$k_1 \leq f(e) \leq k_2 , \qquad \ldots(8.54)$$

where k_1 and k_2 are two real numbers which, for the sake of simplicity, are considered to be positive.

Under these conditions, the system is global and asymptotically stable if the Nyquist plot of g(s) does not cross or go around, in a counter clock wise direction, the circle whose diameter is defined by $(-k_1^{-1},-k_2^{-1})$ as shown in Fig.8.27a.

8.5.2 Stability Analysis – MIMO Case

Let G (s) be a transfer matrix of dimension m x m (multi-Input multi-output Case), which is rational and asymptotically stable. The nonlinear feedback is defined by the diagonal matrix F(e, t), with components f_i (e, t), and meets the restriction

$$k_1 \leq f_1 (e,t) \leq k_2 \qquad \text{where } I = 1,\ldots.,m. \qquad \ldots(8.55)$$

If $\hat{g}_{ij}(jw)$ is the inverse of $g_{ij}(jw)$, with circles of radius $\hat{r}_i(jw)$, determined by

$$\hat{r}\, i\, (jw) = \textstyle\sum_{k=1,k\neq j}\left[\left|\widehat{\hat{g}ik\,(jw)}\right|\right] \qquad \ldots(8.56)$$

In the case of row dominance (see paragraph below), or

$$\hat{r}\, i\, (jw) = \textstyle\sum_{k=1,k\neq j}\left[\left|\widehat{\hat{g}kj\,(jw)}\right|\right] \qquad \ldots(8.57)$$

In the case of column dominance the MIMO system is global and asymptotically stable if the Nyquist band of the inverse system crosses neither the origin nor the circle of diameter (- k_1, -

k_2) and encircles both of them the same number of times in a clockwise direction (see Fig. 8.27b).

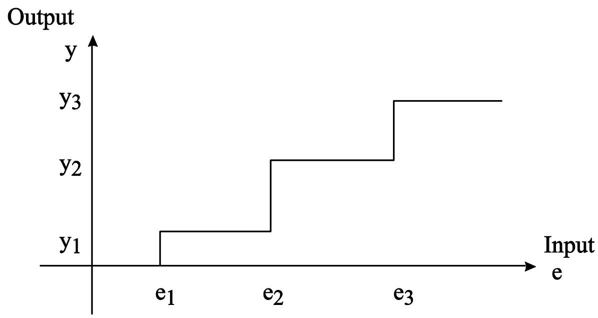
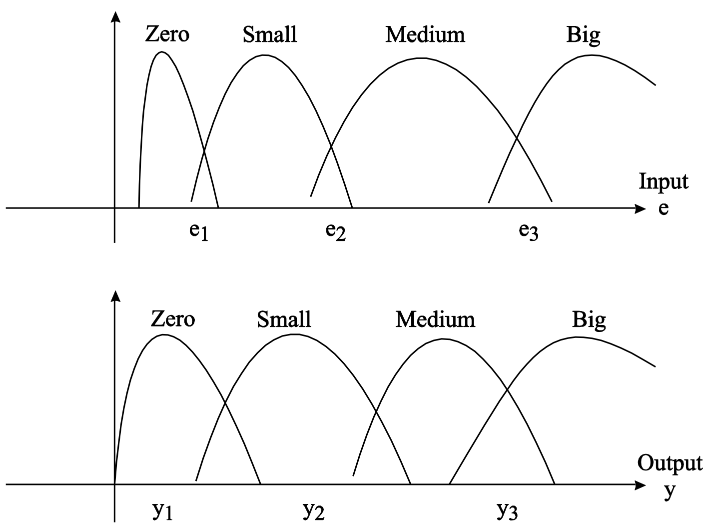


Fig. 8.25. The FKBC as a multi-level relay.



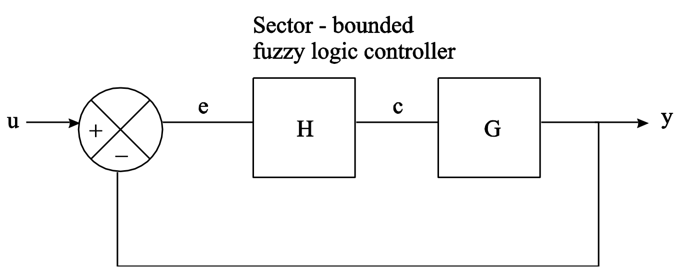
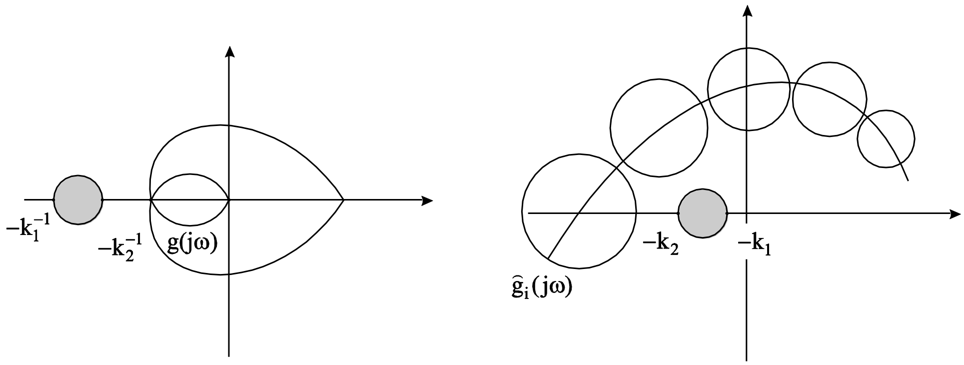


Fig. 8.26. The SISO system.



(a) **(b)**

Fig. 8.27. Nyquist plot (a) SISO case and (b) MIMO case.

Dominance Conditions for stability Analysis

Let G(s) be an m x m rational transfer function matrix defined for all s on a subset D of the complex plane. Usually D will be the Nyquist contour consisting of the imaginary axis from –jR to jR and a semicircle of large radius R in the right half plane, enclosing all poles and zeros of G(s). Let us write $G(s) = G^{-1}(s)$, if $G^{-1}(s)$ exists. Then $\widehat{g_{ii}}(s)$ are the diagonal entries of G(s) and, in general, $g_{ii}(s) \neq (g_{ii}(s))^{-1}$. The rational m x m matrix G(s) is said to be row diagonal dominant on D when for all I = 1,…, m and for all s on D, the following holds;

$$|\hat{g}_{ii}(s)| > \sum_{j=1,j\neq 1}^{m} |\hat{g}_{ij}(s)| \qquad \ldots(8.58)$$

In a dual way, G(s) is said to be column diagonal edominant on D when

$$|\hat{g}_{ii}(s)| > \sum_{j=1,j\neq 1}^{m} |\hat{g}_{js}(s)| \qquad \ldots(8.59)$$

The diagonal dominance has a simple graphical interpretation. For each i = 1,., m, the $g_{ii}(s)$ maps the imaginary axis s = jw in the contour D onto the curve $g_{ii}(jw)$ in the complex plane. Let us define the radius

$$d_i(jw) > \sum_{j=1,j\neq 1}^{m} |\hat{g}_{ij}(jw)| \qquad \ldots(8.60)$$

and the closed disk $D_i(w)$ centered at $\hat{g}_{ii}(jw)$ with radius $d_i(jw)$. Consider the union B_i of all these disks, for all w. This union is called the inverse Nyquist band and, and looks like a band in the complex plane centered along the curve $\hat{g}_{ii}(jw)$ of variable width, depending on the $d_1(jw)$. Then the system G(s) is row dominant when all the bands B_i, for i = 1,…, m, exclude the origin. A similar graphical test for column dominance can be built based on column wise defined radius

$$d'_i(jw) > \sum_{j=1,j\neq 1}^{m} |\hat{g}_{ji}(jw)| \qquad \ldots (8.61)$$

An application of diagonal dominance conditions to control systems stability was developed by Rosenbrock [1972] and Cook [1972] who presented multivariable versions of the circle criterion based on frequency response bands that were called Gershgorin bands. A complete design method can be found in Rosenbrock [1974] or in Bell et al. [1982].

Remarks: Under normal considering the FKBC as a multi-level relay leads to $k_1 = 0$. This would cause the circle to become a semiplane $(-\infty, -k_2^{-1})$ in the SISO case as shown in Fig. 8.28. Furthermore, in the MIMO case the circle has a diameter defined by $(0, k_2)$ as shown in Fig. 8.29.

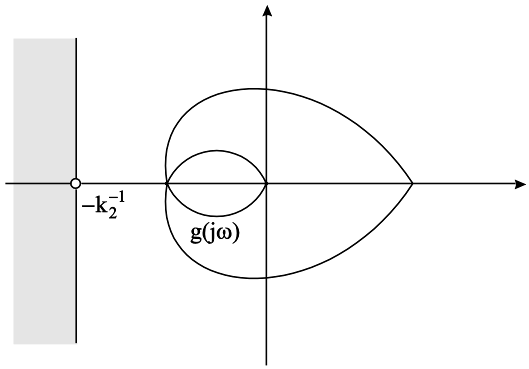


Fig. 8.28. SISO case for k1 =0

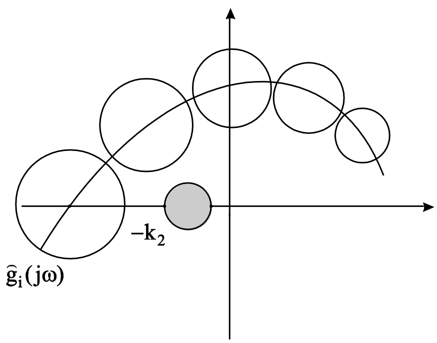


Fig. 8.29. MIMO case for k1 = 0

Table 8.1. FKBC for subsystem 1

S \ E	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NM	NM	NS	ZE	PM	PB
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NM	NM	NB	NB	NS	PS
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PM
PM	NS	NM	NS	PM	PM	PB	PB
PB	ZE	PS	PS	PM	PB	PB	PB

E represents error, S represents error sum, ZE represents Zero, PS, PB, PM, NB, NM ,NS have usual meaning

Example 8.2 Consider the system defined by the following transfer matrix [see Ray et al. (1984)]

$$G(s) = \begin{bmatrix} \frac{1}{(s+1)^3} & \frac{1}{2(s+1)^3} \\ \frac{1}{(s+2)^3} & \frac{3}{2(s+2)^3} \end{bmatrix} \qquad \ldots(8.62)$$

The inverse matrix is defined by

$$\hat{G}(s) = \begin{bmatrix} \frac{2}{3}(s+1)^3 & \frac{-1}{2}(s+1)^3 \\ (s+2)^3 & (s+2)^3 \end{bmatrix} \qquad \ldots(8.63)$$

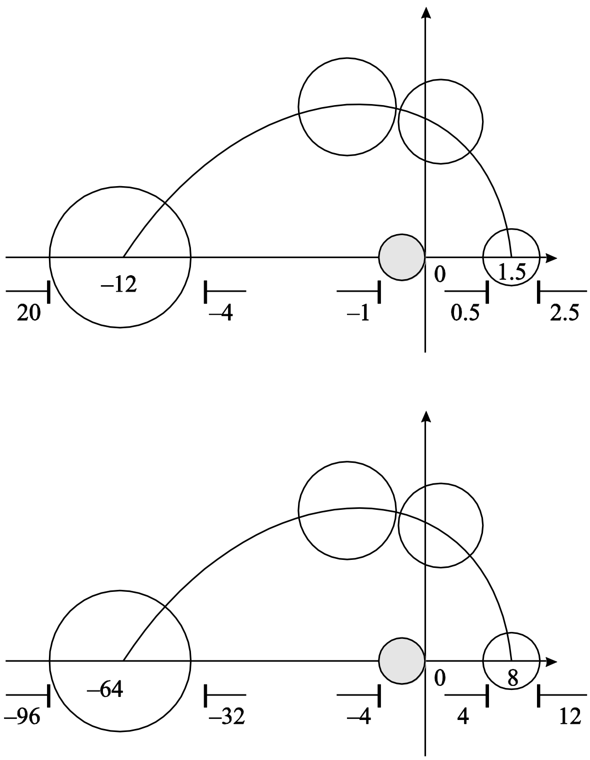


Fig. 8.30. Nyquist plot 1st and 2nd column of Example 8.2.

The open-loop system is column dominant. Thus, the original MIMO system can be considered as two relatively uncoupled subsystems. Tables 8.1 and 8.2 give the FKBC for each one of these subsystems. In both tables S denotes sum-of-error and E denotes error. The controller slopes are defined by the pairs (0, 1) and (0, 4). The feedback system is stable, as shown in Fig.8.30.

S \ E	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NM	NS	NS	ZE	PS	PS
NM	NM	NM	NM	NS	NS	ZE	PS
NS	NB	NM	NS	NM	NS	NS	ZE
ZE	NB	NM	NS	ZE	PS	PS	PM
PS	NS	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PM	PS	PM	PM	PM
PB	ZE	PM	PS	PM	PS	PM	PM

Table 8.2. FKBC for subsystem 2

E represents error, S represents error sum, ZE represents Zero, PS, PB, PM, NB, NM ,NS have usual meaning

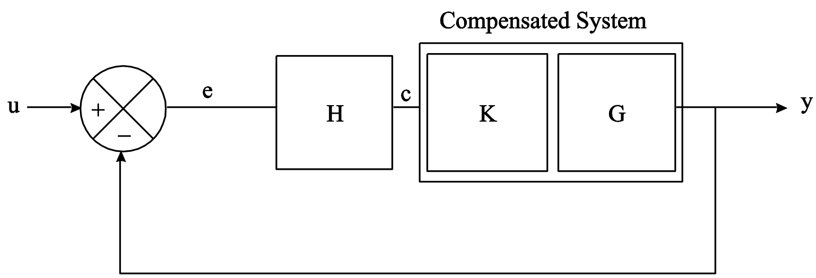


Fig. 8.31. Design of the FKBC for an unstable plant.

8.5.3 Application of the circle criterion to Design

The previously mentioned criterion requires a linear and asymptotically stable plant. [Ray and Majumdar (1984)] used this criterion as an aid in the design of a FKBC. If stability defines the admissible limits for the FKBC. On the other hand, if the plant is unstable, a linear compensator is in-troduced in such a way that the compensated plant is stable (Fig.8.31). The conditions are fixed on the circle in a similar way.

8.6 THE CONICITY CRITERION

Consider the closed-loop system F of Fig. 8.32. If the small-gain condition $_{g}(G)$. $_{g}(H) < 1$ holds, then we can ensure closed-loop stability. If this condition does not hold we can not conclude instability of the small-gain condition is to add.

One way to increase the applicability of the small-gain condition is to add and subtract from F the same block C. This gives rise to the transformed closed-loop T(F,C) of

Fig. 8.33. Under not very restrictive conditions normally met in practice, the stability conditions of F and T(F,C) are equivalent, as stated in the loop transformation theorem [Desoer and Vidysagar (1975)].

If we apply the small gain condition to the transformed loop T(F,C), we obtain the conicity criterion, formulated as follows:

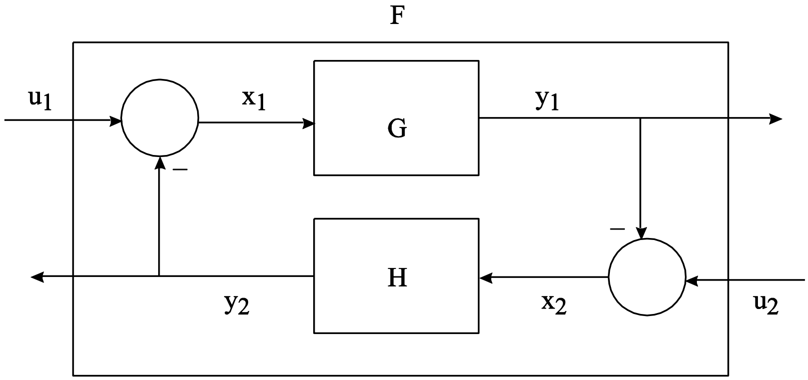


Fig.8.32. Canonical Closed-loop system

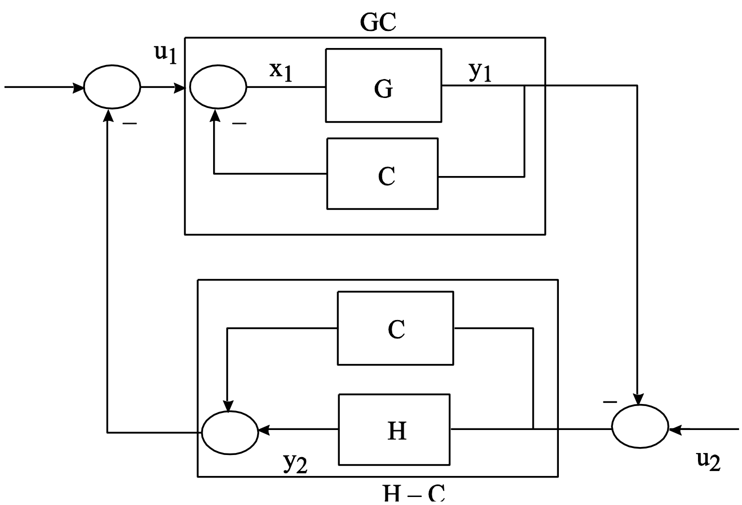


Fig. 8.33. The transformed Closed-loop system

The feedback system F is stable if there is a linear operator C and a positive number r > 0 such that

(a) g(H-C) < r and g(feedb(G,C)) ≤ 1/r.

In the above feedb(G, C) denotes the feedback configuration of the blocks G and C.
The auxiliary elements C and r are called center and radius respectively. An intuitive graphical interpretation of the conicity criterion can be given James(1966) at least in the scalar case (C scalar), in the sense that the graph of H in the plane x-y must lie inside a sector of limiting slopes C + r and C – r. Also, condition (b) can be related to the usual circle criterion condition defined on the scalar frequency response G(jw) (see Section 8.5).
The conicity criterion when stated in this abstract way contains as a particular case [compare Safonov (1980)] the circle criterion. To apply the conicity criterion in practice and thus compute the gains involved using the formulas in Section 8.4 for NLS and LTI systems, the closed-loop synamics must be organized in two blocks. One block say G, must have a LTI model and the other block say H, must have a NLS model.
In this way, the center C must be chosen: (1) static, to compute g(H-C) with the NLS formula, and (2) linear, to compute g(feedback(G,C)) with the LTI formula.
This fact force C to be a linear static (memory less) operator, i.e., matrix of appropriate dimensions. At this point the question is how to construct the center matrix C so that the conicity conditions hold. As no priori information is in general available, one possibility is to explore all centers that may verify the conicity inequalities. This leads to the definitions and criteria given by Barseina and Aracil (1992). The conic deviation dH (C) of the rnonlinear controller H from the center matrix C is given as

$$d_H\,(C) = g\,(H - C) \qquad \ldots(8.64)$$

The conic robustness $r_G(C)$ of the linear plant G with the feedback C is given by

$$r_G(C) = \frac{1}{g(G(I+C.G)^{-1})} \qquad \ldots(8.65)$$

Then d_H (C) gives a measure of the difference between C and H and r_G (C) gives a measure of the robustness of the closed-loop formed with the plant G with the feedback C. If C is chosen so that the closed-loop approaches instability, then the gain approaches infinity and $r_G(C)$ converges to zero. Using this, the conicity stability is given by

$$\exists\, C, r : d_H\,(C) < r \le r_G\,(C) \qquad \ldots(8.66)$$

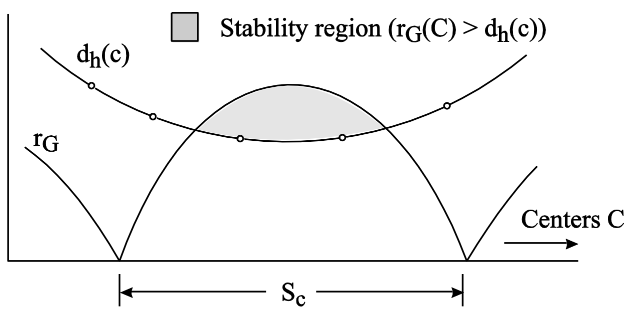


Fig. 8.34. Interpretation of the conicity criterion. Sc is a subset of stabilizing centers.

The intuitive idea behind this is that the feedback system formed with the plant G and controller H is stable if for some linear y = C_x the comic robustness $r_G(C)$ of G with the feedback c (instead of H) is greater than the deviation $d_H(C)$ between H (actual controller) and C (linear controller).

These ideas are illustrated in Fig 8.34. The x-axis is the space of linear centers C. this space is of dimension 1 when C is a scalar

dimension 2 when C is a 1 x 2 matrix

dimension n + m when C is a n x m matrix.

Then the conic deviation $d_H(C)$ and conic robustness $r_G(C)$, is the function of C represented by curve surface of hyper surface. It may happen that $r_G(C)$ is defined only on a subset S_c of stabilizing centre. The stability condition holds for the gray area Fig 8.34.

This formulation is a corollary of the conicity criterion but adds implicitly the idea of exploring among all the possible centers. As it may happen that the stability condition does not hold for some centers. But hold for other we must look for an appropriate centre. In Barreiro and Aracil (1992), techniques to organize this search are given.

If the plant G has a transfer G(s) and frequency response G(jw) the stability condition are related as follow for some rand C:

(i) Non linear conicity ($d_H(C) < r$) given by:

$$\forall\ x: \text{I}H(x) - C\ x\ \text{I} < r \cdot \text{I}\ x\ \text{I} \qquad \text{… (8.67)}$$

(ii) Linear conicity ($r < r_G(C)$ Which is Divided into the following:

(a) Stability of the linear closed loop

$$F(s) = G(s) \cdot (I + c \cdot (s))^{-1} \qquad \text{…(8.68)}$$

(b) Frequency response condition

$$\sup_{\omega} \overline{\sigma}\{F(jw)\} = \sup_{\omega} \overline{\sigma}\{G(jw)\ .\ (I + g(jw))^{-1}\} \geq 1/r \qquad \text{… (8.69)}$$

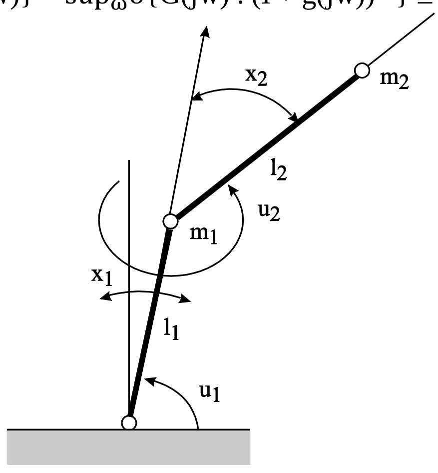


Fig. 8.35. 2-joints manipulator.

This reformulation of the conicity criterion can be easily applied to the analysis and design of FKBC. In the following we represent the application of this method to the robust controls of a two joint manipulator.'

Example 8.3 Consider the joint manipulator shown in figure 8.35 which has 2- rotational joint angle velocities and accelerations given respectively by x_i ,x_i (i =1, 2). Assuming the masses $m_1$and m_2 located at the distal ends of the links with length $l_1$and l_2 the dynamic model is given by

$$M(x) . x= u+ V(x, \dot{x}\ )+G(x) +F(x, \dot{x}\ ) + z, \qquad \ldots(8.70)$$

where x, $\dot{x}$, $\ddot{X}$ are two dimensional vectors u is a two dimensional vectors with the control torque M(x) is a 2 x 2 mass matrix is given as

$$\begin{bmatrix} l_2^2 m_2 + 2 l_1 l_2 m_2 c_2 + l_1^2 (m_1 + m_2 & l_2^2 m_2 + 2 l_1 l_2 m_2 c_2 \\ l_2^2 m_2 + 2 l_1 l_2 m_2 c_2 & l_2^2 m_2 \end{bmatrix} \qquad \ldots(8.71)$$

V (x, $\dot{x}$) is a 2 x 1 centrifugal and Coriolis terms given by

$$V(x, \dot{x}\ ) = m_2 l_1 l_2 s_2 x^2_2 + 2 m_2 l_1 l_2 s2 x_1 x_2 \qquad \ldots(8.72)$$

$$-m_2 l_1 l_2 g c_{12}$$

G(x) is a 2x1 vector of gravity terms given by

$$G(x) = -m_2 l_1 l_2 g c_{12} - (m_1 + m_2)\, l_1 c_1 \qquad \ldots(8.73)$$

$$-m_2 l_1 l_2 g c_{12}$$

F(x, $\dot{x}$) is a model of friction and z is a 2 x 1 vectors of unknown signals due to extras distribance and unmodelled dynamics. Furthermore s_2, c_1, c_2 and c_{12} represent sin (x_2) cos (x_1) Cos (x_2) and cos($x_1 + x_2$) , respectively.

Let

$$q(x, \dot{x}\ ) = V(x, \dot{x}\ ) + G(X) + F(x, \dot{x}\ ) \qquad \ldots(8.74)$$

Using the computed torque technique [ref. Article Craig J.J. (1985)] the joint dynamics are decoupled and liberalized. The control law is then given by

$$u = -q(x, \dot{x}\ ) + M(x). (h(e, e)= \ddot{X}_d) \qquad \ldots(8.75)$$

where $e = x_d - x$, $\dot{e} = \dot{x}_d - \dot{x}$; x_d, $\dot{x}_d$, $\ddot{x}_d$ are desired joint angles angular velocities and acceleration respectively and $h(e, \dot{e})$ is a feedback function.

If the estimated values of parameter in $-q(x, \dot{x})$ and M(x) are close to the real values and there are no external disturbance and unmodelled dynamic (z = o) then the computed torque control law reduce the dynamics of each joint to a double integrator

$$\ddot{e} = -u = -u, \qquad \ldots(8.76)$$

$$u = h(e, \dot{e}) \qquad \ldots(8.77)$$

In the above, the bold letter is omitted thus indicating that now we work with the scalar uncoupled dynamics of a separate joint. These dynamics can be regarded as a feedback configuration formed with a linear one-input two-output plant (input – u outputs e, ė) whose transfer Function is a double integrator

$$g(s) = \begin{pmatrix} 1/s^2 \\ 1/s \end{pmatrix} \qquad \ldots(8.78)$$

The negative feedback is given by the nonlinear statics law h (e, ė) that can be written as

$$u = h(e, \dot{e}) = k_p(e, \dot{e}).e + k_v(e, \dot{e}).\,\dot{e} \qquad \ldots(8.79)$$

In the above position and velocity gains are nonlinear function of the error and derivatives of the error ė. A fixed constant value for the position and speed gain k_p (e, ė) and k_v (e, ė) will suffice in the ideal case of the exact model estimation, however estimation errors unmodelled dynamics and other nonlinearities mainly the actuator deteriorate the control performance

It will be shown how a fuzzy design of the gains solves the actuator saturation problem. However before that we will show how the conicity criterion imposes some restriction on this design. In this way the conicity criterion can be regarded as a complementary stability tool in the fuzzy design process let us apply the conicity condition given above.

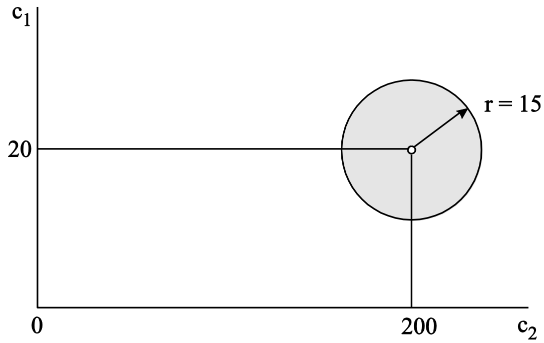


Fig 8.36 Admissible control gains.

Stabililty of Linear feedback:

For some centre C=(c_1,c_2) the feedback system formed with g(s) and C must be stable. Which gives

$$c_1 > 0 \text{ and } c_2 > 0 \qquad \ldots(8.80)$$

Following Craig (1985) we choose the central values of the position and velocity gain as follows;

$$c_1 = 200;\ c_2 = 20 \qquad \ldots(8.81)$$

Linear Conicity:

For some radius r > 0,

$$r \leq r_g(c) = 1/\sup_{w}\left(\frac{(c_1-c_2)^2+(c_2.w)^2}{1+w^2}\right)^{1/2} \qquad \ldots(8.82)$$

such that for a centre c= (200, 20) given rise to this radius $r < r_g(c) \approx 15$.

Nonlinear conicity;

From the condition

$$\forall e, \dot{e} : I\ h(e, \dot{e}) - (c_1.e + c_2.\dot{e}) I < r. |(e, \dot{e})|, \qquad \ldots(8.83).$$

we obtain

$$((k_p - c_1)^2 + (k_v - c_2)^2)^{1/2} \leq r. \qquad \ldots(8.84).$$

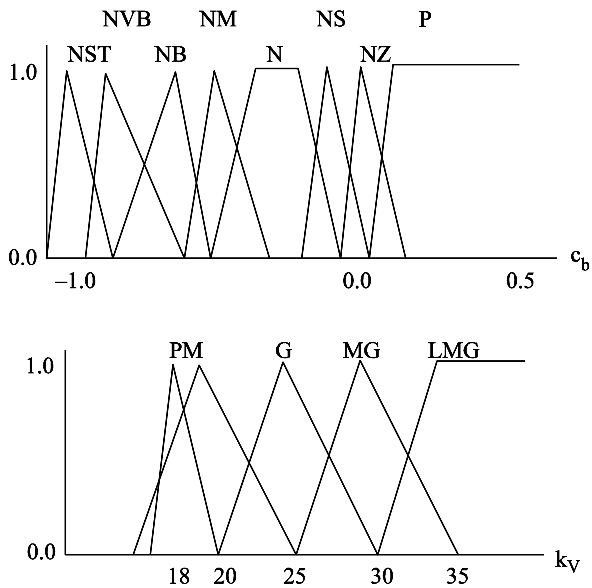
In other words, the position and the speed of gain $k_p(e, \dot{e})$ and $k_v(e, \dot{e})$ must always be interior to the circle of centre c and radius r_g in the gain plane as shown in Fig 8.36. It may be noted that if the gain were fixed then any positive value $k_p > 0$ and $k_v > 0$ ensure linear stability. The circle restriction of Fig8.36 can be regarded as the price paid for the nonlinear variability of the gains.

Normalized error

NST Strongly negative
NVB Negative very big
NB Negative big
NM Negative medium
N Negative
NS Negative small
NZ Near zero
P Positive

Gain

P Positive
M Big
VB Very Big
BS Highly big



If e_b is NST then k_v is P	If e_b is NVB then k_v is M
If e_b is NB then k_v is B	If e_b is NM then k_v is VB
If e_b is N then k_v is BS	If e_b is NS then k_v is VB
If e_b is NZ then k_v is M	If e_b is P then k_v is M

Fig 8.37 (a) Membership function (b) if then rules where e_b represent the normalized error defined by $e_b = e_v (x_d - x)$

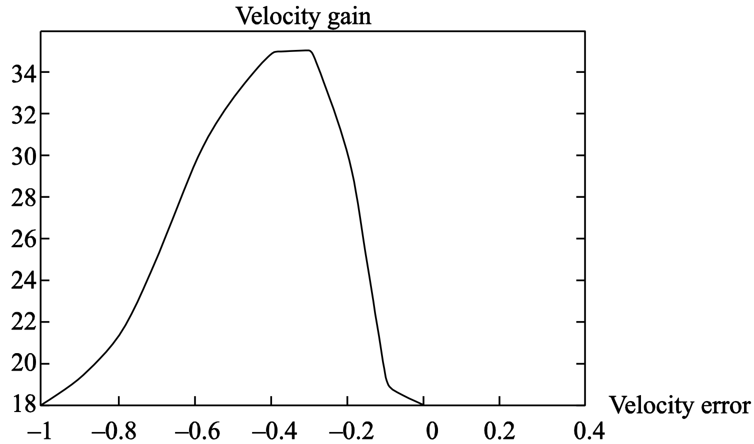


Fig. 8.38. The nonlinear function $K_v = \Phi(\dot{e})$.

Based on the conicity Stability condition the fuzzy design of the error gain can be accomplished same the following way, we assume the following working condition for the manipulator.

i. Refereneces: ramp of slope 0.5 rad / sec.,
ii. Saturation: the saturation of the variable u is assumed to be $\pm$ 20 rad / sec^2 and
iii. Design objectives : obtain a fast ramp response with absolute error as small as possible and avoiding if possible the saturation region.

Since the reference are ramp signals at the start time or the times of the slopes change the will be a large speed error hence the position gain has little effect and is kept constant at the value k_p= 200. The velocity gain will be adjusted function of the velocity error $k_v = \varphi(\dot{e})$ by mean of the set of if then rules.

It should be noted that the final conicity stability condition is

$$C_2 - r \approx 20 - 15 = 5 \leq k_v = \varphi(\dot{e}) \leq c_2 + r \approx 20 + 15 = 35. \qquad \ldots(8.85)$$

This condition can be guaranteed provided that the rule consequent member ship function for k_v be in the range [8.35]. Fig 8.37 shows the definition of rules and membership function this formulation of the FKBC and the membership function after fuzzification rules firing, Defuzzification result in a nonlinear function k_v (e) that varies the stability condition this function is shown in Fig8.38.

Finally Fig. 8.39 shows the simulation results. It may be noted that the fuzzy speed gain $k_v = \varphi(\dot{e})$ gives a better globle response than the linear gain setting k_v = 5 (with large initial error) or K_v = 35 with slow convergence. Thus the wholes model is simulated with the two joint coupled simulation Gracia-Cerezo (1987) shows that even for considerable error (50%) in the estimated M(x) and q(x, $\dot{x}$) of the actual M(x) used in Craige [1985] the fuzzy design of the speed gain work well showing inherent robustness.

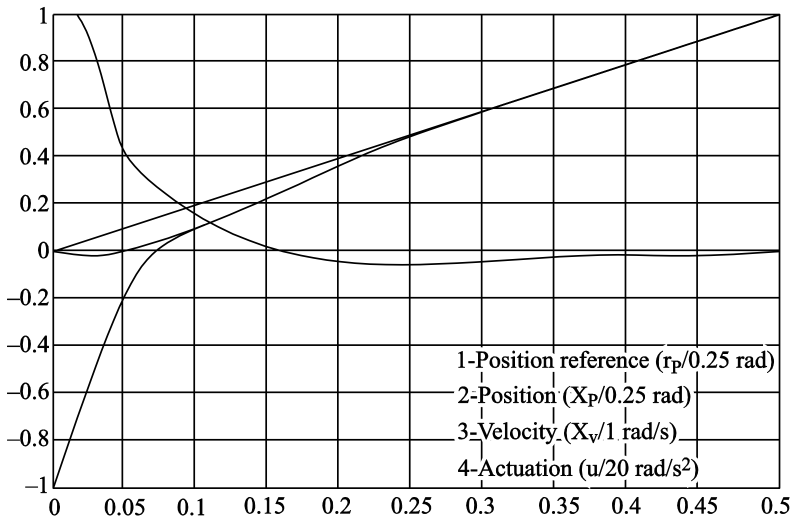


Fig 8.39 Result of the fuzzy design

EXERCISE-8

1 Consider an SISO dynamic system of the form

$$\dot{x}(t) = x^2(t) + u(t)$$
$$x(t_0) = x_0.$$

Assume that there exists a fuzzy model that accurately describes this system:

IF $x(t) \in S_n$ AND $u(t) \in S_m$ THEN $\dot{x}(t) \in S_{an+bm}$.

The definition for the fuzzy sets S_n and S_m are

$$S_n = [(n - 1)\sigma,(n+1)\sigma),$$
$$\mu_{S_n}(x) = \begin{cases} 1-|x-n\sigma|/\sigma & \text{for } x \in [(n-1)\sigma,(n+1)\sigma), \\ 0 & \text{else.} \end{cases}$$

(a) Estimate the constants {a,b} for each of the following ranges for x(t): (i) [0,0.5), (ii) [0.5,1.0), (iii) [1.0,4.0), (iv) [4.0,10.0), (v) [–0.5,0), (vi) [–1.0,–0.5), (vii) [–4.0,–1.0), and (viii) [–10.0,–4.0).

(b) Design, for each of the ranges in part (a), a simple fuzzy controller in the form of

IF $x(t) \in S_n$ THEN $u(t) \in S_{kn}$,

i.e., determine the values of k in this fuzzy expression for each range listed in part (a).

(c) Write a simulation program that combines the actual system, the fuzzy controllers in part (b), along with condition checking, to determine when x(t) falls into a particular range to activate a specific controller designed for that range. Use the following conditions $x0 = 3.0$ and $\Delta t = 0.02$ sec. Plot the output x(t) for $t \in [0.0,5.0)$ in seconds.

2. Consider an SISO dynamic system of the form

$$\dot{x}(t) = \sin(x(t)) + u(t)$$
$$x(t_0) = x_0.$$

Assume that a set of data is collected for output x(t) at the beginning of every time interval $[t_n, t_n + \Delta t)$. Denote this data set $D_{x,n}$ with

$$D_{x,n} = \{x(t_0), x(t_1), x(t_2), \ldots, x(t_n)\}.$$

Similarly, assume that a set of data $D_{u,n}$ is collected for the input u(t) that drives the system, with

$$D_{u,n} = \{u(t_0), u(t_1), u(t_2), \ldots, u(t_n)\}.$$

(a) Set up recursive formulas to update the parameters a and b in the corresponding fuzzy model

IF $x(t) \in S_n$ AND $u(t) \in S_m$ THEN $\dot{x}(t) \in S_{an+bm}$,

with fuzzy sets S_n and S_m defined as

$$S_n = [(n-1)\sigma,(n+1)\sigma),$$
$$\mu_{S_n}(x) = \begin{cases} 1-|x-n\sigma|/\sigma & \text{for } x \in [(n-1)\sigma,(n+1)\sigma), \\ 0 & \text{else.} \end{cases}$$

for the data sets $D_{x,n}$ and $D_{u,n}$.

(b) For an input signal, $u(t) = \sin(\pi t)$, write a computer program to generate input and output profiles $D_{u,n}$ and $D_{x,n}$ for the system at time $t_0 = 0.0$ to $t_n = 10.0$ sec. with $\Delta t = 0.02$ sec. Plot the input u(t) and output x(t) versus time t.
(c) Write a computer program to update the parameters a and b in the fuzzy model used to approximate the given nonlinear system. Plot the parameters a(t) and b(t) versus time t.
(d) Design a simple fuzzy controller in the form of

$$\text{IF } x(t) \in S_n \text{ THEN } u(t) \in S_{kn}$$

and determine the values of k in the above fuzzy expression as functions of parameters a and b.
(e) Write a computer program to combine the dynamic system, the update of the parameters a and b for the fuzzy model, the update of the parameter k for the fuzzy controller, and the input/output for the closed-loop system with initial condition $x(0) = 3.0$. Plot the input u(t) and output x(t) versus time t. How long does it take for the system to converge to final position $x_f = 0$? Does the output system oscillate at this value? Explain.

3. Consider an SISO dynamic system of the form $\dot{x}(t) = \sin(t)\, x(t) + \cos(t)\, u(t)$

$$x(t_0) = x_0.$$

Assume that a set of data is collected for output x(t) at the beginning of every time interval $[t_n, t_n+\Delta t)$. Denote this data set $D_{x,n}$ with

$$D_{x,n} = \{x(t_0), x(t_1), x(t_2), \ldots, x(t_n)\}.$$

Similarly, assume that a set of data $D_{u,n}$ is collected for the input u(t) that drives the system, where

$$D_{u,n} = \{u(t_0), u(t_1), u(t_2), \ldots, u(t_n)\}.$$

Repeat the steps from (a) through (e) in Exercise **2** for the present case.

4. Consider an SISO dynamic system of the form

$$\dot{x}(t) = 3\, x(t) + 2\, u(t)$$
$$x(t_0) = x_0.$$

(a) Design a Model Reference Adaptive Fuzzy System so that

$$u(t) = \gamma(t)\, x(t)$$

will drive the closed-loop system to track the fuzzy model

$$\text{IF } x(t) \in S_n \text{ AND } u(t) \in S_m \text{ THEN } \dot{x}(t) \in S_{6n+4n}.$$

(b) Write a computer program that plots the outputs and inputs of both the bounds of the fuzzy model and the dynamic system. Compare the results. Is the output of the closed-loop dynamic system tracking the fuzzy model, i.e., does it fall in between the two bounds?

5. Consider an SISO dynamic system of the form

$$\dot{x}(t) = x_2(t) + u(t)$$
$$x(t_0) = x_0.$$

(a) Design a Model Reference Adaptive Fuzzy System, so that

$$u(t) = \gamma(t)\, x(t)$$

will drive the closed-loop system to track the fuzzy model

$$\text{IF } x(t) \in S_n \text{ AND } u(t) \in S_m \text{ THEN } \dot{x}(t) \in S_{2n+n}.$$

(b) Write a computer program that plots the outputs and inputs of both the bounds of the fuzzy model and the dynamic system. Compare the results. Is the output of the closed-loop dynamic system tracking the fuzzy model, i.e., does it fall in between the two bounds?

BIBLIOGRAPHY

1. Aracil, J., Garcia Cerezo A., and ollero A., “Stability analysis of fuzzy control system ; A Geometrical Approch.” in the C.A. Kulikovsky in R.M Huber and G.A ferrate A.I expert system and language in modelling and simulation Amsterdam North Holland, 1988.
2. Aracil, J,. Ollero A,. and Garcia Cerezo, A,(1989) Stability Indices for the global analysis of expert control system “IEEE trans on System man and Cybernetics 19(5) 998-1007.
3. AracilJ,. Garcia Cerezo, A,. Barreiro, A,. and ollero A,. “Design of expert Fuzzy Controller Based on the stability Criteria ,” Proc of the If Sa 91 Brussels, 1991.
4. Barreiro, A., and Aracil, J., “stability of Uncertain Dynnamical system,” preprints of the IFAC Symposium on AI in Real-Time Control, Delft, the Netherlands, 1992.
5. Bell, D.J., Cook, P.A., Munro, N. (eds), design of Modern control systems, Peter Peregrinus, IEE Control Engineering Series, 1982.
6. Braae,M., and Rutherford, D.A., “ selection of Parameters for a Fuzzy Logic Controler,” Aytomatica, 1979.
7. Braae,M., and Rutherford, D.A.(1979) “Theoretical and Linguistic Aspects of the Fuzzy logic controller ,” Automatica, 15. 185-199.
8. Cook , P.A., “Modified Multivariable circle theorems ,” In: D.J.Bell (ed), Recent Mathematical Developments in Control, Academic Press, 1972.
9. Craig, J.J., Adaptive Control of mechanical manipulators, Addison-Wesley, 1985.
10. Desoer, C.R., and Vidyasagar,M., Feedback Systems. Input-Out properties, academic press, 1975.
11. Gracia-Cerezo, A., Aplicaciones del Razonamiento Aproximado enel Control y Supervision de Procesos, Ph.D. thesis, University of Santigo, Spain, 1987.
12. Gracia-Cerezo, A., Barreiro, A., Aracil, J., and Ollero, A., “ Design of Robust Intelligent Control of Manipulators,” proceedings of the IEEE Int. Conference on Systems Engineering, Dayton, Ohio, USA, 1991.
13. Gracia-Cerezo, A., Ollero, A., and Aracil, J., “ Stability of Fuzzy control systems by using Nonlinear systems theory,” Preprints of the IFAC Symposium on AI in Real – Time C ontrol, The Netherlands, 1992.
14. Glas, M.de, (1984) “Inveriance and Stability of Fuzzy System,” J. Math. Analysis Appl., 199299-319.
15. Guckemheimer , J., and Holmes, P., Nonlinear Oscillations, Dynamical Systems and Bifuracations of Vector Fields, New york, Springer-Verlag,1983.
16. Kickert, W.M., and Mamdani, E.H.(1978) Analysis of Fuzzy Logic Controllers,” Fuzzy Sets and Systems, 1(1) 29-44.
17. Kiszka, J.B., Gupta, M.M., and Nikiforuk, M.N.(1985) “Energetistic Stability of Fuzzy Dynamic Systems,” IEEE Trans. On Systems, Man, and Cybernetics, 15 783-792.
18. Ollero, A., Gracia-Cerezo, A., and Aracil, J., Design of Fuzzy control Systems, Dpto. Ing. Sist., University Malaga, Research Report, 1992.
19. Ray, K.S., and Majumdar, D.D.(1984) “ Application of the circle Criteria for Stability Analysis of Linear SISO and MIMO Systems Associated with Fuzzy Logic Controller,” IEEE Trans. Systems, Man, and Cybernetics, 14(2) 345-349.

20. Ray, K.S., Ghosh, A., and Majumdar, D.D.(1984) " L2-Stability and the Related Design Concept for SISO Linear System Associated with Fuzzy Logic Controllers," IEEE Trans. Systems, Man and Cybernetics, 14(6) 932-939.
21. Rosenbrock, H.H., "Multivariable Circle Theorems," In:D.J. Bell (ed), Recent Mathematical Developments in Control, London, Academic Press, 1972.
22. Rosenbrock, H.H.,Computer Aided Control System Design, London, Academic Press, 1974.
23. Safonov, M.G., Stability and Robutustness of Multivariable Feedback Systems, Cambridge, MA,MIT Press,1980.
24. Tong, R.M.(1976) "Analysis of Fuzzy Control Algorithms using the Relation Matrix," Int.J. Man-Machine Studies, 8 679-689.
25. Tong, R.M.(1978) "Synsthesis of Fuzzy Models for industrial Processes- Some Recent Results," Int. J. General Systems, 4 143-162.
26. Vidyasagar, M.,(1986) "New Directions of Research in Nonlinear System Theory," Proceeding of the IEEE, 77(8)1060-1090.

Chapter-9

Fuzzy Control System Models

9.1 INTRODUCTION

This chapter introduces some engineering applications of fuzzy logic and fuzzy control systems. The purpose of this introduction is mainly to illustrate workability and applicability of fuzzy logic and fuzzy control in real-life situations based on the fuzzy systems theories developed in the previous chapters of the book.

Three main applications are discussed in the following order: a fuzzy rule-based expert system for a health care diagnostic system monitoring vital signs of a human patient; a fuzzy control system for a diabetes system, and a fuzzy modeling and design of a fuzzy controller for a Air conditioner control system. For each application, a description of the functionality of the system is given, which is followed by the fuzzy logic rules and computer simulation results.

9.2. HEALTH MONITORING SYSTEMS

Diagnostic systems are used to monitor the behavior of a process and identify certain pre-defined patterns that associate with well-known problems. These identified problems, provides suggestions for specific treatment.

9.2.1 Introduction of Fuzzy Diagnostic Systems

Most diagnostic systems are in the form of a rule-based expert system: a set of rules is used to describe certain patterns. Observed data are collected and used to evaluate these rules. If the rules are logically satisfied, the pattern is identified and a problem associated with that pattern is suggested. Each particular problem might imply a specific treatment. In general, the diagnostic systems are used for consultation rather than replacement of human expert. Thus, the final decision is still with the human expert to determine the cause and to prescribe the treatment.

Most current health monitoring systems only check the body's temperature, blood pressure and heart rate against individual upper and lower limits. these start an audible alarm and each signal move out of its predefined range (either above the upper limit or below the lower limit). Then, human experts (nurses or physicians) examine the patient and probe the patient's body further for additional data that lead to proper diagnosis and its corresponding treatment.

Other more complicated systems normally involve more sensors that provide more data but still follow the same pattern of independently checking individual sets of data against some upper and lower limits. The warning alarm from these systems only carries a meaning that there is something wrong with the patient. Thus, attending staff would have to wait for the physician to make a diagnostic examination before they could properly prepare necessary equipment for a corresponding treatment. In a lifethreatening situation, reducing the physician's reaction time (the time between the warning and the time proper treatment is given to the patient) by preparing proper equipment for specific treatment in advance would significantly increase the patient's chance of surviving.

In the field of exploration where a team of humans is sent to a distant and isolated location, it is important to have a health monitoring system that can give early diagnostic data of each explorer's health status to prepare for the continuation of the mission. In space exploration missions, an astronaut who suddenly gets sick will be diagnosed and the data reviewed by the onboard physician as well as the physician team on earth to determine if a proper treatment onboard is possible (for the continuation of the mission) or if specific treatment on earth is required (for the abortion of the mission).

Hence, the health monitoring system is playing two roles in every space mission: to monitor the health of the astronaut and to aid in determining wheather a mission should be continued or aborted (due to serious illness). This section presents a simple implementation of a health monitoring expert system utilizing fuzzy rules for its rule base. This health monitoring expert system consists of a set of sensors monitoring three vital signs of a patient: body temperature, blood pressure and heart rate.

9.2.1. Fuzzy Rule-Based Health Monitoring Expert Systems

In this system, we deal with a fuzzy rule-based expert health monitoring system with three basic sensors: body temperature, heart rate and blood pressure. Note that the blood pressure is measured in two readings: systolic pressure (the maximum pressure that the blood exerts on the blood vessel, i.e., the aorta, when the pumping chamber of the heart contracts), and diastolic pressure (the lowest pressure that remains in the small blood vessel when the pumping chamber of the heart relaxes). For simplicity of discussion only diastolic pressure is used with the understanding that an additional reading can be easily added to the system and the number of diagnostic cases will increase accordingly. The expert system will check for combinations of data instead of individual data and thus will identify twenty-seven different scenarios instead of three in the conventional system.

Individual sensors can identify three isolated cases:

(i) high body temperature indicates high fever normally associated with the body fighting against some infectious virus or bacteria, some hormone disorder such as hyperthyroidism, some autoimmune disorder such as rheumatoid arthritis or some damage to the hypothalamus
(ii) high blood pressure indicates hypertension normally associated with some kidney disease, hormonal disorder such as hyperaldoteronism, or acute lead poisoning and
(iii) high heart rate indicates rapid heart beat normally associated with an increase in adrenaline (a hormone produced by the adrenal glands) production.

In addition, three more cases can be identified:

(iv) low body temperature indicates hypothermia
(v) low blood pressure normally associates with excess bleeding, muscle damage, heart valve disorder, or excessive sweating/urination and
(vi) low heart rate normally associates with abnormal pacemaker or with the blockage between the pacemaker and the atria where pacemaker signal is received to stimulate heartbeats.

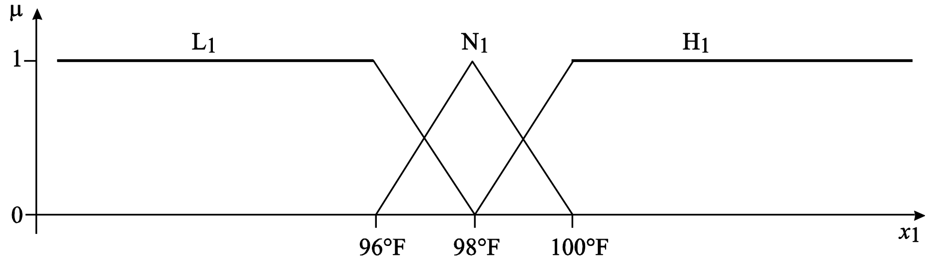


Figure 9.1 Definitions and membership functions of three different ranges for body temperatures.

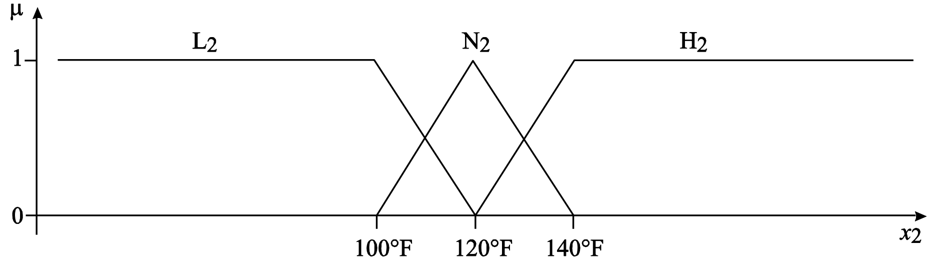


Figure 9.2 Definitions and membership functions of three different ranges for blood pressure.

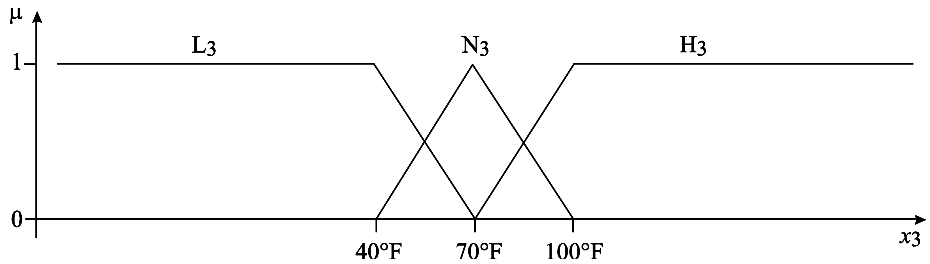


Figure 9.3 Definitions and membership functions of three different ranges for heart rate.

However, the three individual sensors, each with three settings (normal, high, and low), can be combined to give 27 different scenarios. With the perfectly normal case (where all three give normal readings) and the six cases defined above, there are additionally 20 more cases where combinations of abnormal readings can be observed. For example, low blood pressure and high body temperature might indicate severe exposure to heat and a severe loss of blood or lack of body fluid.

Let x_1 be the body temperature, x_2 the (systolic) blood pressure, x_3 the heart rate and y the diagnostic statement. Let L_i, N_i and H_i represent the three sets of low range, normal range and high range for sensor data xi, where i = 1, 2, or 3. Furthermore, let C_0, C_1, C_2, ..., C26 be the individual scenarios that could happen for each combination of the different data sets.

The low range for the body temperature can be defined as below 98°F. Similarly, the normal range is 98°F and high range is above 98°F. One can define three ranges and three membership functions as in Figure 9.1. These functions have the mathematical representation as follows:

$$\mu_{L_1}(x_1) = \begin{cases} 1 & \text{if } -\infty < x_1 \le 96^{\circ}\text{F}, \\ (98 - x_1)/2 & \text{if } 96^{\circ}\text{F} < x_1 \le 98^{\circ}\text{F}, \\ 0 & \text{if } 98^{\circ}\text{F} < x_1 < \infty, \end{cases}$$

$$\mu_{N_1}(x_1) = \begin{cases} (x_1 - 96)/2 & \text{if } 96^{\circ}\text{F} < x_1 \le 98^{\circ}\text{F}, \\ (100 - x_1)/2 & \text{if } 98^{\circ}\text{F} < x_1 \le 100^{\circ}\text{F}, \\ 0 & \text{else}, \end{cases}$$

$$\mu_{H_1}(x_1) = \begin{cases} 0 & \text{if } -\infty < x_1 \le 98^{\circ}\text{F}, \\ (x_1 - 98)/2 & \text{if } 98^{\circ}\text{F} < x_1 \le 100^{\circ}\text{F}, \\ 1 & \text{if } 100^{\circ}\text{F} < x_1 < \infty, \end{cases}$$

$$\mu_{L_2}(x_2)=\begin{cases}1 & \text{if } -\infty < x_2 \le 100\text{Hg},\\ (120-x_2)/20 & \text{if } 100\,\text{Hg} < x_2 \le 120\,\text{Hg},\\ 0 & \text{if } 120\,\text{Hg} < x_2 < \infty,\end{cases}$$

$$\mu_{N_2}(x_2)=\begin{cases}(x_2-100)/20 & \text{if } 100\,\text{Hg} < x_2 \le 120\,\text{Hg},\\ (140-x_2)/20 & \text{if } 120\,\text{Hg} < x_2 \le 140\,\text{Hg},\\ 0 & \text{else},\end{cases}$$

$$\mu_{H_2}(x_2)=\begin{cases}0 & \text{if } -\infty < x_2 \le 120\,\text{Hg},\\ (x_2-120)/20 & \text{if } 120\,\text{Hg} < x_2 \le 140\,\text{Hg},\\ 1 & \text{if } 140\,\text{Hg} < x_2 < \infty,\end{cases}$$

$$\mu_{L_3}(x_3)=\begin{cases}1 & \text{if } -\infty < x_3 \le 40\,\text{Hz},\\ (70-x_3)/30 & \text{if } 40\,\text{Hz} < x_3 \le 70\,\text{Hz},\\ 0 & \text{if } 70\,\text{Hz} < x_3 < \infty,\end{cases}$$

$$\mu_{N_3}(x_3)=\begin{cases}(x_3-40)/30 & \text{if } 40\text{Hz} < x_3 \le 70\,\text{Hz},\\ (100-x_3)/30 & \text{if } 70\,\text{Hz} < x_3 \le 100\,\text{Hz},\\ 0 & \text{else},\end{cases}$$

$$\mu_{H_3}(x_3)=\begin{cases}0 & \text{if } -\infty < x_3 \le 70\,\text{Hz},\\ (x_1-70)/30 & \text{if } 70\,\text{Hz} < x_3 \le 100\,\text{Hz},\\ 1 & \text{if } 100\,\text{Hz} < x_3 < \infty,\end{cases}$$

From this knowledge, four basic rules can be defined as follows:

$R^{(0)}$: IFx_1 is N_1 AND x_2 is N_2 AND x_3 is N_3 THEN y is C_0

$R^{(1)}$: IFx_1 is H_1 AND x_2 is N_2 AND x_3 is N_3 THEN y is C_1

$R^{(2)}$: IFx_1 is N_1 AND x_2 is H_2 AND x_3 is N_3 THEN y is C_2

$R^{(3)}$: IFx_1 is N_1 AND x_2 is N_2 AND x_3 is H_3 THEN y is C_3.

In addition, three scenarios can be identified when only one sensor provides data that are lower than the lower limit:

$R^{(4)}$: IFx1 is L_1 AND x_2 is N_2 AND x_3 is N_3 THEN y is C_4

$R^{(5)}$: IFx1 is N_1 AND x_2 is L_2 AND x_3 is N_3 THEN y is C_5

$R^{(6)}$: IFx1 is N_1 AND x_2 is N_2 AND x_3 is L_3 THEN y is C_6.

It can be seen that C_0 corresponds to normal condition, C_1 high fever, C_2 hypertension, C_3 rapid heart rate, C_4 hypothermia, C_5 low blood pressure, C_6 low heart rate. The remaining 20 cases can be defined as follows:

$R^{(7)}$: IFx_1 is L_1 AND x_2 is L_2 AND x_3 is L_3 THEN y is C_7

$R^{(8)}$: IFx_1 is L_1 AND x_2 is L_2 AND $x_3$is N_3 THEN y is C_8

$R^{(9)}$: IFx_1 is L_1 AND x_2 is L_2 AND x_3 is H_3 THEN y is C_9

$R^{(10)}$: IFx_1 is L_1 AND x_2 is N_2 AND x_3 is L_3 THEN y is C_{10}

$R^{(11)}$: IFx_1 is L_1 AND x_2 is N_2 AND x_3 is H_3 THEN y is C_{11}

$R^{(12)}$: IFx_1 is L_1 AND x_2 is H_2 AND x_3 is L_3 THEN y is C_{12}

$R^{(13)}$: IFx_1 is L_1 AND x_2 is H_2 AND x_3 is N_3 THEN y is C_{13}

$R^{(14)}$: IFx_1 is L_1 AND x_2 is H_2 AND x_3 is H_3 THEN y is C_{14}

$R^{(15)}$: IFx_1 is N_1 AND x_2 is L_2 AND x_3 is L_3 THEN y is C_{15}

$R^{(16)}$: IFx_1 is N_1 AND x_2 is L_2 AND x_3 is H_3 THEN y is C_{16}

$R^{(17)}$: IFx_1 is N_1 AND x_2 is H_2 AND x_3 is L_3 THEN y is C_{17}

$R^{(18)}$: IFx_1 is N_1 AND x_2 is H_2 AND x_3 is H_3 THEN y is C_{18}

$R^{(19)}$: IFx_1 is H_1 AND x_2 is L_2 AND x_3 is L_3 THEN y is C_{19}

$R^{(20)}$: IFx_1 is H_1 AND x_2 is L_2 AND x_3 is N_3 THEN y is C_{20}

$R^{(21)}$: IFx_1 is H_1 AND x_2 is L_2 AND x_3 is H_3 THEN y is C_{21}

$R^{(22)}$: IFx_1 is H_1 AND x_2 is N_2 AND x_3 is L_3 THEN y is C_{22}

$R^{(23)}$: IFx_1 is H_1 AND x_2 is N_2 AND x_3 is H_3 THEN y is C_{23}

$R^{(24)}$: IFx_1 is H_1 AND x_2 is H_2 AND x_3 is L_3 THEN y is C_{24}

$R^{(25)}$: IFx_1 is H_1 AND x_2 is H_2 AND x_3 is N_3 THEN y is C_{25}

$R^{(26)}$: IFx_1 is H_1 AND x_2 is H_2 AND x_3 is H_3 THEN y is C_{26}.

Physicians can provide their knowledge in medical science to label individual cases C_i for i = 7, 8, ..., 26. For example, the condition C_{26} with high body temperature, high blood pressure and high heart rate might indicate hyperthyroidism (a condition in which excess thyroid hormone is produced); the condition C_{25} with high body temperature, high blood pressure, and normal heart rate might indicate pheochromocytoma (a tumor in the adrenal gland causing overproduction of a hormone that triggers high blood pressure and raises body temperature) or the condition C_{20} with high body temperature, low blood pressure, and normal heart rate might indicate heat stroke (a condition associated with prolonged exposure to heat).

The membership function for a rule is calculated as a minimum of the memberships of individual conditions, i.e., the membership functions for rules

$R^{(0)}$, $R^{(1)}$, $R^{(2)}$ and $R^{(3)}$ are

$$\mu_R^{(0)}(x_1, x_2, x_3) = \min\{\mu_{N_1}(x_1), \mu_{N_2}(x_2), \mu_{N_3}(x_3)\},$$

$$\mu_R^{(1)}(x_1, x_2, x_3) = \min\{\mu_{H_1}(x_1), \mu_{N_2}(x_2), \mu_{N_3}(x_3)\},$$

$$\mu_R^{(2)}(x_1, x_2, x_3) = \min\{\mu_{N_1}(x_1), \mu_{H_2}(x_2), \mu_{N_3}(x_3)\} \text{ and}$$

$$\mu_R^{(3)}(x_1, x_2, x_3) = \min\{\mu_{N_1}(x_1), \mu_{N_2}(x_2), \mu_{H_3}(x_3)\}.$$

The membership functions for rules $R^{(4)}$ through $R^{(26)}$ can be similarly established according to their corresponding rules given earlier.

9.2.3 Computer Simulations

In this section, numerical simulations are created to demonstrate the workability of 27 rules given in the previous section representing a health monitoring expert system.

Example 9.2.1. In this example, the rule $R^{(1)}$ for high fever, $R^{(2)}$ for hypertension and $R^{(3)}$ for rapid heart rate are tested. The temperature profile $x_1(t)$, blood pressure profile $x_2(t)$ and heart rate profile $x_3(t)$ are generated according to the following formulas:

$$x_1(t) = 98 + 4 \sin(\frac{\pi}{20} t)\, e(t) + n(t),$$

$$x_2(t) = 120 + 40 \sin(\frac{\pi}{20} t)\, e(t-20) + n(t) \text{ and}$$

$$x_3(t) = 70 + 60 \sin(\frac{\pi}{20} t)\, e(t-40) + n(t),$$

where the function n(t) is random white noise and e(t) the rectangular envelop function of the form:

$$e(t) = \begin{cases} 1 & \text{if } 0 \le x \le 20 \\ 0 & \text{else} \end{cases}$$

These three formulas for $x_1(t)$, $x_2(t)$, and $x_3(t)$ create three separate time intervals, $I_1 \approx [7,13)$, $I_2 \approx [27,33)$, and $I_3 \approx [47,53)$. These data profiles will result in rule $R^{(1)}$ being valid in I_1, rule $R^{(2)}$ valid in I_2, and rule $R^{(3)}$ valid in I_3.

Thus, the expected membership function $\mu_R^{(1)}(x_1,x_2,x_3)$ should yield the value 1 for $t \in I_1$ and the value 0 for everywhere else. Similarly, the expected membership function $\mu_R^{(2)}(x_1,x_2,x_3)$ should yield the value 1 for $t \in I_2$ and the value 0 for everywhere else and the membership function $\mu_R^{(3)}(x_1,x_2,x_3)$ should yield the value 1 for $t \in I_3$ and the value 0 for everywhere else.

Example 9.2.2. In this example, the rule $R^{(20)}$ for heat stroke condition (high body temperature, low blood pressure, and normal heart rate) is tested. The temperature profile $x_1(t)$, blood pressure profile $x_2(t)$ and heart rate profile $x_3(t)$ are generated according to the following formulas:

$$x_1(t) = 98 + 4 \sin(\frac{\pi}{40} t)\, e(t) + n(t),$$

$$x_2(t) = 120 - 40 \sin(\frac{\pi}{40} t)\, e(t) + n(t) \text{ and}$$

$$x_3(t) = 70 + n(t),$$

where the function n(t) is random white noise and e(t) the rectangular envelop function described in the previous example. These three formulas for $x_1(t)$, $x_2(t)$, and $x_3(t)$ create an interval $I_4 \approx [10,30)$ where the rule $R^{(20)}$ is valid, i.e., the membership function $\mu_R^{(20)}(x_1,x_2,x_3)$ should yield the value 1 for $t \in I_4$ and the value 0 for everywhere else.

Example 9.2.3. In this example, the rule $R^{(26)}$ for hyperthyroidism (high body temperature, high blood pressure and rapid heart rate) is tested. The temperature profile $x_1(t)$, blood pressure profile $x_2(t)$, and heart rate profile $x_3(t)$ are generated according to the following formulas:

$$x_1(t) = 98 + 4 \sin(\frac{\pi}{40} t)\, e(t) + n(t),$$

$$x_2(t) = 120 + 40 \sin(\frac{\pi}{40} t)\, e(t) + n(t) \text{ and}$$

$$x_3(t) = 70 + 60 \sin(\frac{\pi}{40} t)\, e(t) + n(t),$$

where the function n(t) is random white noise and e(t) the rectangular envelop function described in Example 9.1. These three formulas for $x_1(t)$, $x_2(t)$ and $x_3(t)$ create an interval $I_5 \approx [10,30)$ where the rule $R^{(26)}$ will be valid, i.e., the membership function $\mu_R^{(26)}(x_1,x_2,x_3)$ should yield the value 1 for $t \in I_5$ and the value 0 for everywhere else.

9.2.4. Numerical Results

In this section, computer simulations are created for the three examples listed in the previous section. Data are generated at the rate of one sample per second for the duration of 75 seconds.

For each example three sets of data, $X_1 = \{x_1(n) \mid n = 0, 1, 2, \ldots, 74\}$, $X_2 = \{x_2(n) \mid n = 0, 1, 2, \ldots, 74\}$ and $X_3 = \{x_3(n) \mid n = 0, 1, 2, \ldots, 74\}$ are generated.

In Example 9.1, the membership functions $\mu_R^{(1)}(x_1(n),x_2(n),x_3(n))$, $\mu_R^{(2)}(x_1(n),x_2(n),x_3(n))$ and $\mu_R^{(3)}(x_1(n),x_2(n),x_3(n))$

are generated for n = 0, 1, 2, …, 74. In Examples 9.2 and 9.3, the membership functions $\mu_R^{(20)}(x_1(n),x_2(n),x_3(n))$ and $\mu_R^{(26)}(x_1(n),x_2(n),x_3(n))$ are also generated for n = 0, 1, 2, …, 74.

Figures 9.4 to 9.6 show the plot of the data $x_1(t)$ for body temperature, $x_2(t)$ for blood pressure and $x_3(t)$ for heart rate generated for Example 9.1. Figure 9.8 shows the resulting membership functions of the three conditions C_1, C_2 and C_3 tested in Example 9.1. The data are generated at the beginning of each time interval of 1 second for 75 seconds. In the first 20 seconds, $x_2(t)$ and $x_3(t)$ are in normal range and $x_1(t)$ is gradually moving toward the high range causing the membership function of condition C_1 to rise toward 1. Similarly, during the time interval [20,40], $x_1(t)$ and $x_3(t)$ are in normal range, and $x_2(t)$ is gradually moving toward the high range, causing the membership function of condition C_2 to rise toward 1 and during the time interval [40,60], $x_1(t)$ and $x_2(t)$ are in normal range, and $x_3(t)$ is gradually moving toward the high range causing the membership function of condition C_3 to rise toward 1. Therefore all three signals go back to normal, causing the membership function for condition C_0 to rise toward 1.

Figures 9.8, 9.9 and 9.10 show the plot of the data $x_1(t)$, $x_2(t)$, and $x_3(t)$ generated for Example 9.2. Figure 9.11 shows the resulting membership function for condition C_{20}. Here, conditions for $x_1(t)$ to be high, $x_2(t)$ to be low, and $x_3(t)$ to be normal are simulated. The corresponding membership for condition C_{20} is shown to confirm that the rule $R_{(20)}$ is valid.

Figures 9.12, 9.13 and 9.14 show the plot of the data $x_1(t)$, $x_2(t)$ and $x_3(t)$ generated for Example 9.3. Here, conditions for $x_1(t)$ to be high, $x_2(t)$ to be high and $x_3(t)$ to be high are simulated. The corresponding membership for condition C_{26} is shown in Figure 9.15 to confirm that the rule $R^{(26)}$ is valid.

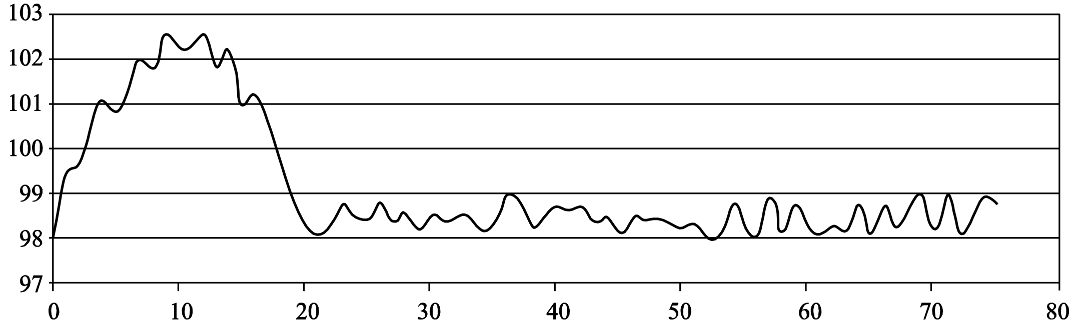


Figure 9.4 Body temperature profile for Example 9.1.

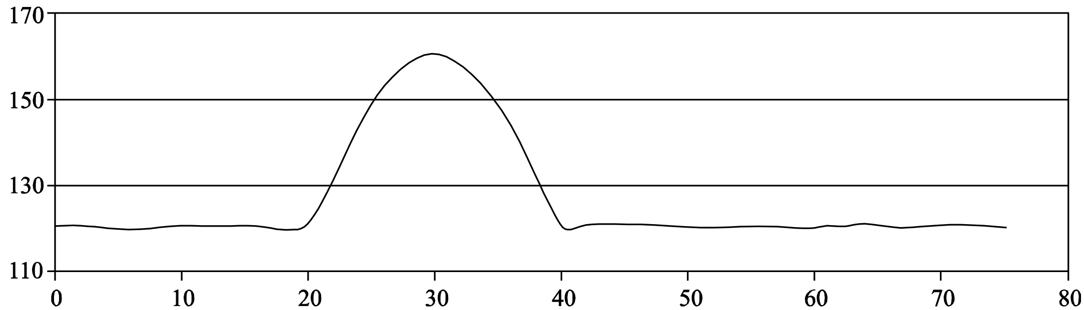


Figure 9.5 Blood pressure profile for Example 9.1.

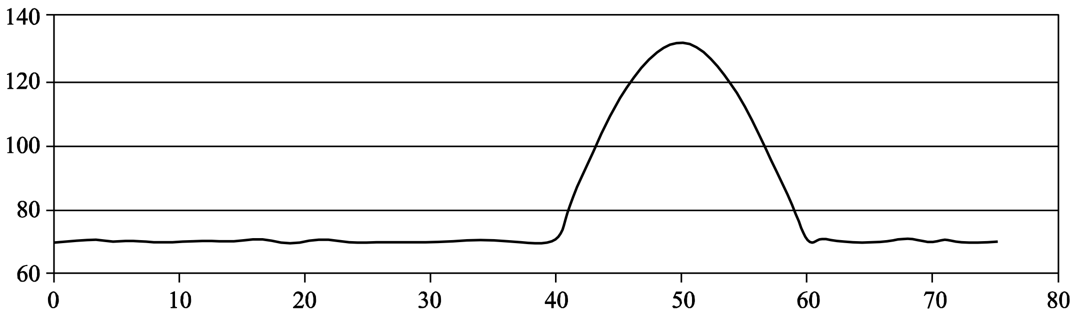


Figure 9.6 Heart rate profile for Example 9.1.

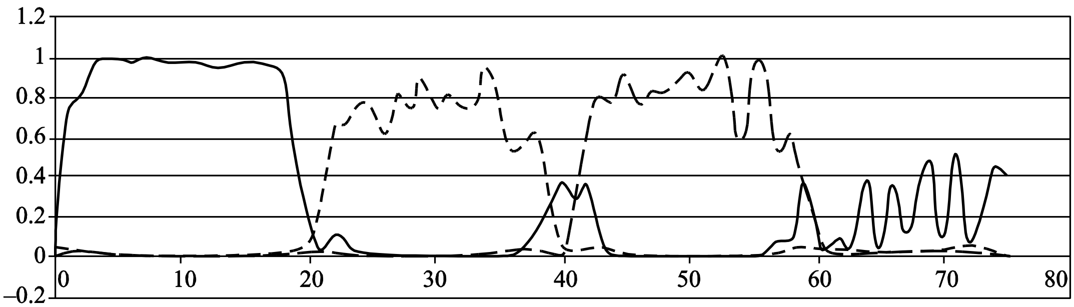


Figure 9.7 Membership functions of conditions C1, C2, and C3 for Example 9.1.

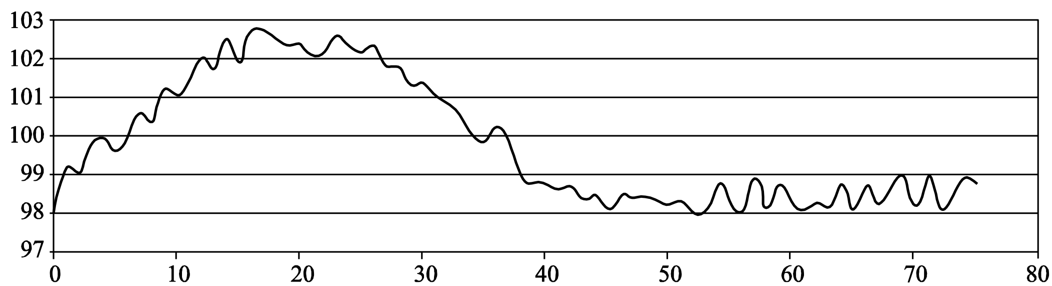


Figure 9.8 Body temperature profile for Example 9.2.

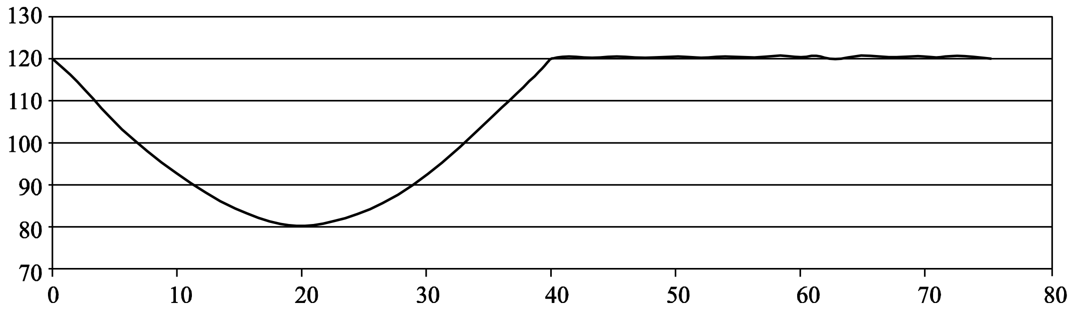


Figure 9.9 Blood pressure profile for Example 9.2.

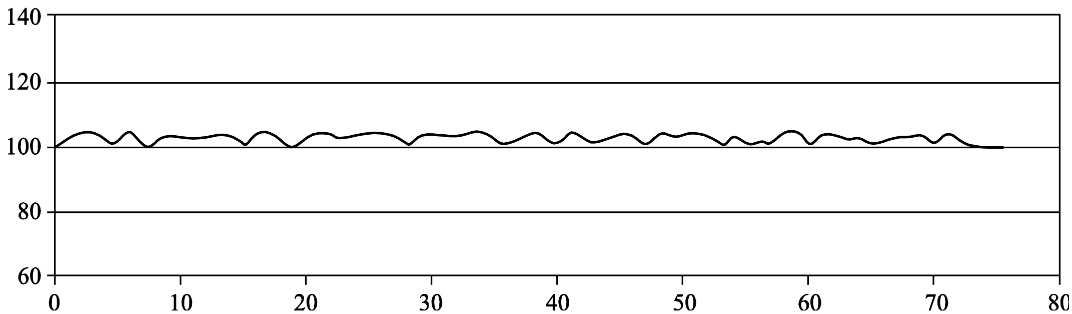


Figure 9.10 Heart rate profile for Example 9.2.

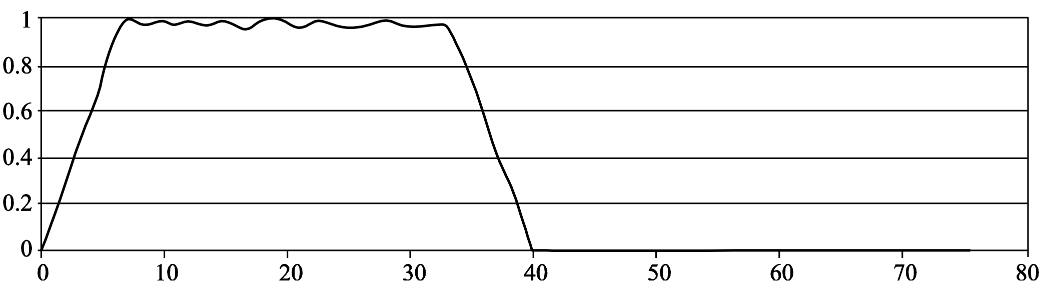


Figure 9.11 Membership function of condition C20 for Example 9.2**.**

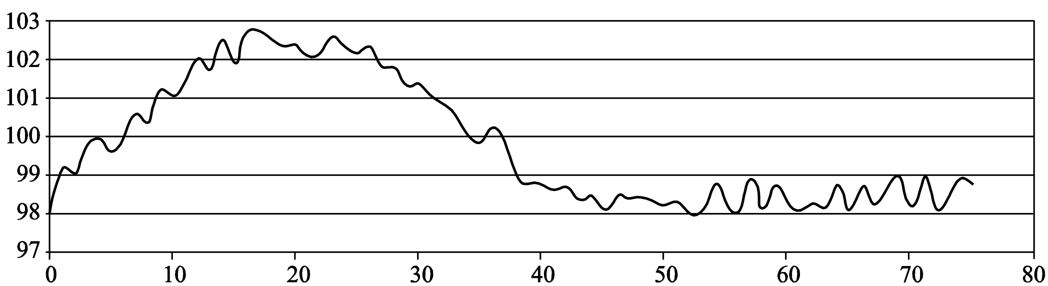


Figure 9.12 Body temperature profile for Example 9.3.

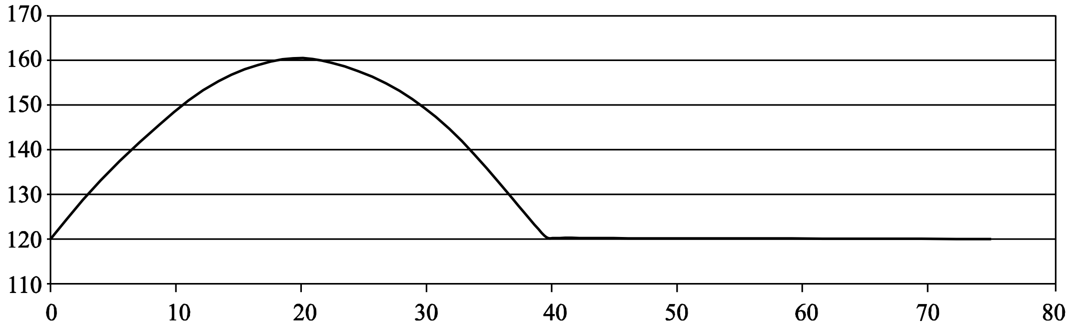


Figure 9.13 Blood pressure profile for Example 9.3.

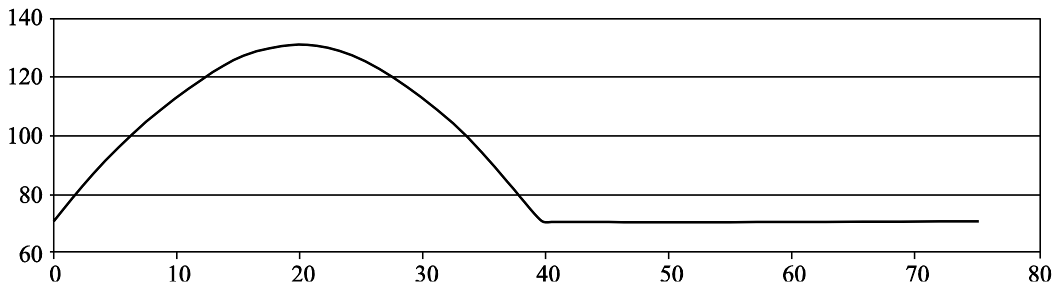


Figure 9.14 Heart rate profile for Example 9.3.

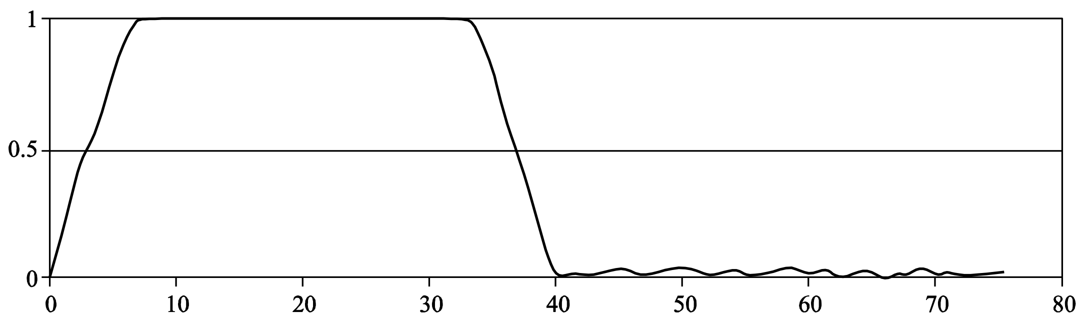


Figure 9.15 Membership function of condition C_{26} for Example 9.3.

9.3 MONITORING SYSTEM FOR DIABETES

We require energy when we walk briskly, **run** for the bus, ride a bike take an aerobic class and for our day to day chores and that is provided by the glucose in blood. When food is taken, it is broken **down** to smaller components. Sugars and carbohydrates are thus broken ("down into glucose for the body to utilize them as energy source. The liver is **also** able to manufacture glucose. In a healthy person the hormone insulin [which is made by the beta cells of the pancreas] regulates how much glucose is in the blood. When there is excess glucose in the blood, insulin stimulates cells to absorb enough glucose from the blood for the energy that they need.

9.3.1 Types of diabetes

Diabetes type-1 insulin-dependent diabetes (IDDM) or early-onset diabetes **d**evelops when the body does not produce insulin. People usually develop type 1 diabetes before their 40th year, often in early adulthood or teenage years. Patients with type 1 diabetes will need to take insulin injections for the rest of their life; Symptoms are extreme thirst, Frequent urination, Drowsiness or Lethargy, Sudden weight loss, Sugar in the urine, Fruity odor on the breath, Heavy or Labored breathing, etc. **Gestational Diabetes** is also considered as type 1 and affects females during pregnancy. Women have very high levels of glucose in their blood. Diagnosis is made during pregnancy. Between 10% to 20% of them need to take some kind of blood-glucose-controlling medications. Undiagnosed or uncontrolled gestational diabetes can raise the risk of complications during childbirth. The baby may be bigger than he/she should be. Scientists from the National Institutes of Health and Harvard University found that women whose diets before becoming pregnant were high in animal fat and cholesterol had a higher risk for gestational diabetes

Diabetes Type-2 Adult-onset diabetes mellitus or non-insulin dependent-diabetes mellitus (NIDDM) develops when the body does not produce enough insulin for proper function or the cells in the body do not react to insulin. The body does not produce enough insulin for proper function of the cells in the body do not react to insulin (insulin resistance). Approximately 90% of all cases of diabetes worldwide are of this type. Some people may be able to control their type 2 diabetes symptoms by losing weight, following a healthy diet, doing plenty of exercise and monitoring their blood glucose levels. However, type 2 diabetes is typically a progressive disease-it gradually gets worse and the patient takes insulin, usually in tablet form. People usually develop type 2 diabetes after their 40th year.

Symptoms of type 2 diabetes are **i**ncreased hunger, increased thirst, frequent urination, Blurred vision and Slow-healing sores or frequent infections.

Risk factors are high blood pressure, History of gestational diabetes, sedentary lifestyle, Age factor, polycystic ovary syndrome, Tri-glyceride level and Overweight.

Polat and Sumer used an expert system approach based on principal component analysis and adoptive neuro fuzzy inference system diagnosis of diabetes [refer to Berkan and Trubetch(1997). This model is designed for diagnosis of Type -2 diabetes patients using if then rules. The process of fuzzfication is also introduced here.

Recently, a work in this direction is to design a proposition for using mathematical models based on a fuzzy system with application, while some author’s recent work is to design a mathematical structure of fuzzy modeling of medical diagnoses by using clustering models.

9.3.2 Modeling Process

(i) Fuzzification of Impaired fasting glucose test (Do not eat or drink anything except water for 8 – 10 hours before a fasting blood glucose test)

Consider N= Normal, M = Medium, H = High and V.H. = very high.

$$\mu_N(x_1) = \begin{cases} 1 & 80 \le x_1 \le 100 \\ \frac{110-x_1}{10} & 100 \le x_1 \le 110 \end{cases}$$

$$\mu_{V.H}(x_1) = \begin{cases} \frac{x_1-140}{10} & 140 \le x_1 \le 150 \\ 1 & x_1 \ge 150 \end{cases}$$

$$\mu_H(x_1) = \begin{cases} \frac{x_1-120}{10} & 120 \le x_1 \le 130 \\ 1 & 130 \le x_1 \le 140 \\ \frac{150-x_1}{10} & 140 \le x_1 \le 150 \end{cases}$$

$$\mu_M(x_1) = \begin{cases} \frac{x_1-100}{10} & 100 \le x_1 \le 110 \\ 1 & 130 \le x_1 \le 140 \\ \frac{150-x_1}{10} & 140 \le x_1 \le 150 \end{cases}$$

(ii) Fuzzification of Triglyceride level (It is the form of the chemical glycerol- tri = three molecules of fatty acid + gleceride = glycerol)

Consider N= Normal, M = Medium, H = High and V.H. = very high.

$$\mu_N(x_2) = \begin{cases} 1 & 100 \le x_2 \le 150 \\ \frac{170-x_2}{25} & 150 \le x_2 \le 175, \end{cases}$$

$$\mu_{V.H}(x_2) = \begin{cases} \frac{x_2-425}{25} & 425 \le x_2 \le 450 \\ 1 & x_2 \ge 450, \end{cases}$$

$$\mu_H(x_2) = \begin{cases} \frac{x_2-200}{25} & 200 \le x_2 \le 225 \\ 1 & 225 \le x_2 \le 425 \\ \frac{450-x_2}{25} & 425 \le x_2 \le 450 \end{cases}$$

and

$$\mu_M(x_2) = \begin{cases} \frac{x_2-150}{25} & 150 \le x_2 \le 175 \\ 1 & 175 \le x_2 \le 200 \\ \frac{150-x_1}{10} & 200 \le x_2 \le 225 \end{cases}$$

(iii) Fuzzification of Blood Pressure (It is the pressure of blood in our arteries (blood vessels). It is measured in mm Hg it is recorded two sides 120 - 80

Consider N= Normal, M = Medium, H = High and V.H. = very high.

$$\mu_N(x_3) = \begin{cases} 1 & 100 \le x_3 \le 120 \\ \frac{120-x_3}{10} & 120 \le x_3 \le 130\text{'} \end{cases}$$

$$\mu_{V.H}(x_3) = \begin{cases} \frac{x_3-160}{10} & 1605 \le x_3 \le 170 \\ 1 & x_3 \ge 170, \end{cases}$$

$$\mu_H(x_3) = \begin{cases} \frac{x_3-140}{10} & 140 \le x_3 \le 150 \\ 1 & 150 \le x_3 \le 160 \\ \frac{170-x_3}{10} & 160 \le x_3 \le 170, \end{cases}$$

and

$$\mu_M(x_3) = \begin{cases} \frac{x_3-120}{10} & 120 \le x_3 \le 130 \\ 1 & 130 \le x_3 \le 140 \\ \frac{150-x_3}{10} & 140 \le x_3 \le 150 \end{cases}$$

(iv) Fuzzification of Bodyweight (It is according to BMI which is calculated as weight/(height)^2

Consider N= Normal, M = Medium, H = High and V.H. = very high.

$$\mu_N(x_4) = \begin{cases} 1 & 18 \le x_4 \le 20 \\ \frac{25-x_4}{5} & 20 \le x_4 \le 25 \end{cases}, \qquad \mu_{V.H}(x_4) = \begin{cases} \frac{x_4-40}{5} & 40 \le x_4 \le 45 \\ 1 & x_4 \ge 45, \end{cases}$$

$$\mu_H(x_4) = \begin{cases} \frac{x_4-30}{5} & 30 \le x_4 \le 35 \\ 1 & 35 \le x_4 \le 40 \\ \frac{45-x_4}{5} & 40 \le x_4 \le 45, \end{cases} \quad \text{and} \quad \mu_M(x_4) = \begin{cases} \frac{x_4-20}{5} & 20 \le x_4 \le 25 \\ 1 & 25 \le x_4 \le 30 \\ \frac{35-x_4}{5} & 30 \le x_4 \le 35 \end{cases}$$

9.3.3 Defuzzification

The center of gravity (COG) method is used for defuzzification process.

$$\text{Output data} = \frac{\sum_{i \epsilon x_{min}}^{x_{max}} x_i \mu(x_i)}{\sum_{i \epsilon x_{min}}^{x_{max}} x_i \mu(x_i)}$$

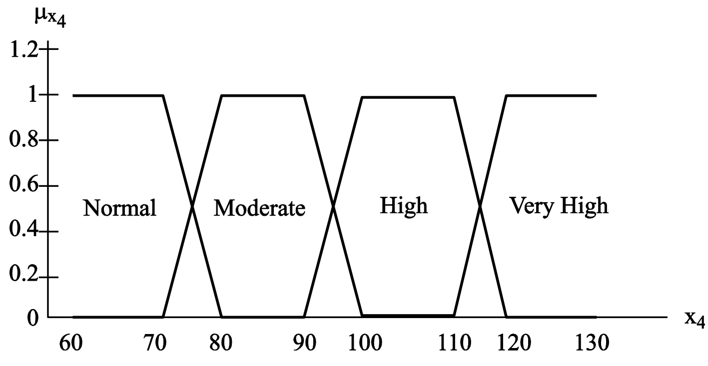


Fig. 9.16 Body weight x_4

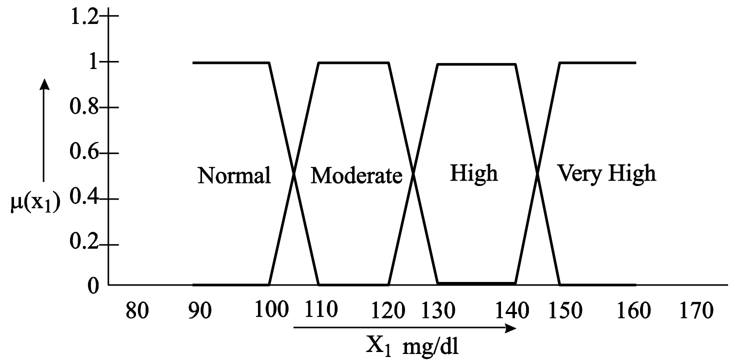


Fig. 9.17 Impaired fasting glucose x_1

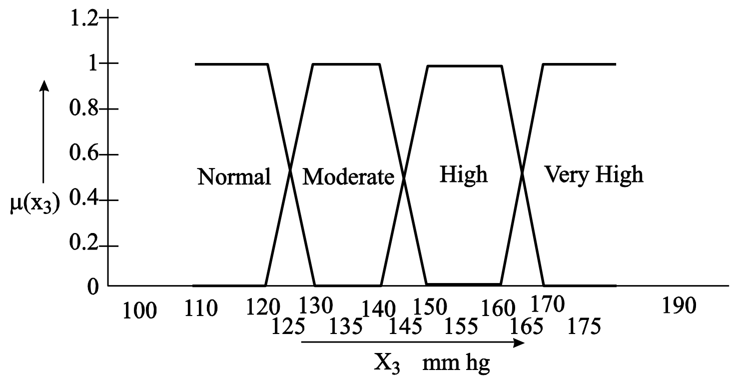


Fig. 9.18 Blood pressure x_3

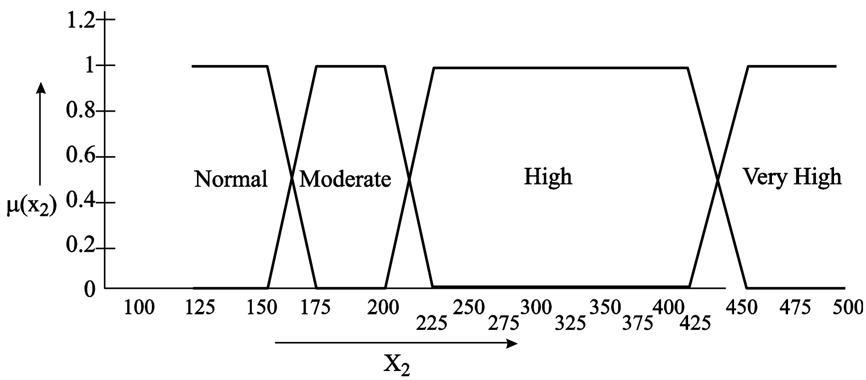


Fig. 9.19 Triglyceride level x_2

9.3.4 Case Study:

Patients A1 Age 62 date 15.2.2015 Suffering with polydipsia, polyuria, irritation in urination swelling on feet, weakness

X_1 = Imapaired glucose fasting value – 165 mg/dl

X_2 = Triglyceride test value - 212 mg/dl

X_3 = Blood pressure – 165 mm Hg systolic

X_4 = BMI – 34.2 kg/m^2

The fuzzified value of the crisp inputs by the use of membership functions defined for each fuzzy set for each linguistic variable as shown in figure 9.16 – 9.19 is as below

$\mu_N(x_1) = 0$ $\mu_M(x_1) = 0$ $\mu_H(x_1) = 0$ $\mu_{V.H.}(x_1) = 1$

$\mu_N(x_2) = 0$ $\mu_H(x_2) = 0.52$ $\mu_M(x_2) = 0.42$ $\mu_{V.H.}(x_2) = 0$

$\mu_N(x_3) = 0$ $\mu_M(x_3) = 0$ $\mu_H(x_3) = 0.5$ $\mu_{V.H.}(x_3) = 0.5$

$\mu_N(x_4) = 0$ $\mu_M(x_4) = 0.16$ $\mu_H(x_4) = 0.84$ $\mu_{V.H.}(x_4) = 0$

9.3.5RULE BASE

Rule-	IFGT	Triglyceride	Blood pressure	BMI	Output
1	V.H.	Moderate	High	Moderate	Level 2
2	V.H.	Moderate	High	High	Level 2
3	V.H.	Moderate	V.H.	Moderate	Level 2
4	V.H.	Moderate	V.H.	High	Level 3
5	V.H.	High	High	Moderate	Level 2
6	V.H.	High	High	High	Level 3
7	V.H.	High	V.H.	Moderate	Level 3
8	V.H.	High	V.H.	High	Level 4

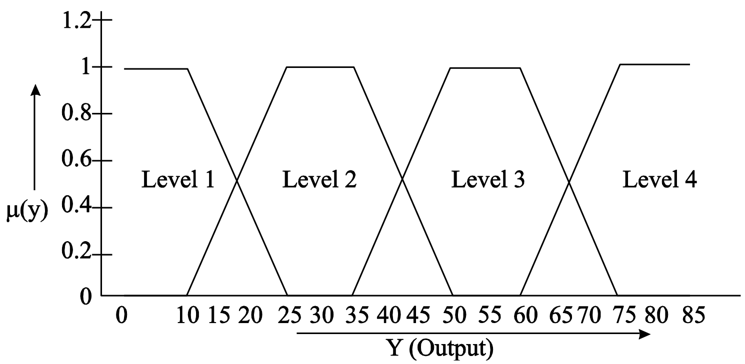


Fig. 9.20 Output of the system

9.3.6 Execute the inference system:

We use root sum square (RSS) method to combine the effect of all applicable rules.

Level 2 = $\sqrt{\sum_{j \in L_2} (\mu R_i)^2} = \sqrt{(0.16)^2 + (0.5)^2 + (0.16)^2 + (0.16)^2} = 0.5766$

Similarly Level 3 = 0.8430 and Level 4 = 0.48

Output of the decision of the expert system is given by

$= \frac{0.5766 \text{ x } 0.30 + 0.8480\text{x } 0.55 + 0.48\text{x}0.8}{0.5766 + 0.8430 + 0.48} = 0.53728$

This output shows that the patient A1 is at level 3 diabetes with 53.728 % degree of precision.

9.4 AIR CONDITIONER CONTROLLER

9.4.1 Air Conditioner overview

The system comprises a dial to control the flow of warm/hot or cool/cold air and a thermometer to measure the room temperature (0°C). When the dial is Turned positive, warm/hot air is supplied from the air conditioner; If it is turned negative, cool/cold air is suppliedand if it is set to zero, no air is supplied.

A person now notices the difference in temperature (ΔT°C) between the room temperature (T°C) as measured by the thermometer and the desired temperature (T_0 °C) at which the room is desired to be kept (set-point).

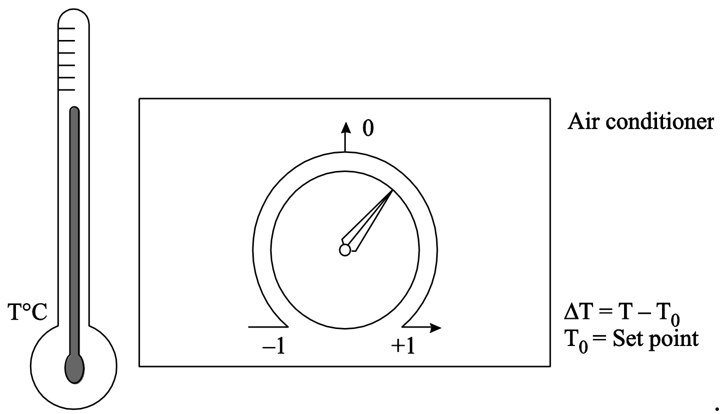


Fig. 9.21 Air conditioner control system

9.4.2 Air conditioner control problem

The problem how to determine and what extent the dial should be turned so that the appropriate supply of air (hot/ warm/cool/cold) may nullify the change in temperature is discussed as follows.

(i) Rule Base

Table 9.1: Fuzzy rule base for air conditioner control

S.No.	**Fuzzy rule (Descriptive)**	**Fuzzy rule (Notational)**
1.	If the room temperature is approximately equal to the set point T_0 °C, ΔT is approx. zero (ZE)and the temperature is rapidly changing higher, i.e., dΔT/dt is PL	If ΔT is ZE and dΔT/dt is PL then dial should be NL.
2.	If the room temperature is high and there is no change in temperature, i.e., ΔT is positive large (PL) and dΔT/dt is approximately zero (ZE) then blow cold air at an intermediate level, i.e., turn the dial negative medium (NM)	If ΔT is PL and dΔT/dt is ZE then dial should be NM.
3.	If the room temperature is a little bit higher than the set point and the temperature is gradually decreasing, i.e., ΔT is positive small (PS) and dΔT/dt is negatively small (NS) then there is no need to blow hot or cold air, i.e., turn the dial to approximately zero (ZE)	If ΔT is PS and dΔT/dt is NS then dial should be ZE.

(ii) **Fuzzification**:

The fuzzy sets for the system inputs namely ΔT and dΔT/dt, and the system output, namely turn of the dial are as shown in the following figures 8.11 to 8.13:

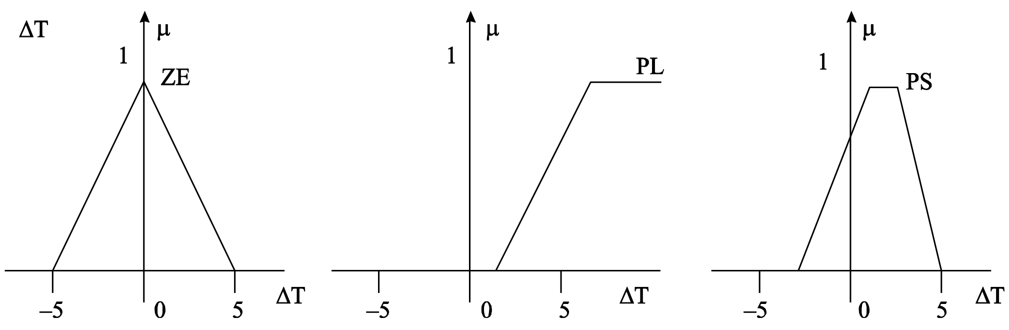


Fig. 9.22 Temperature Difference

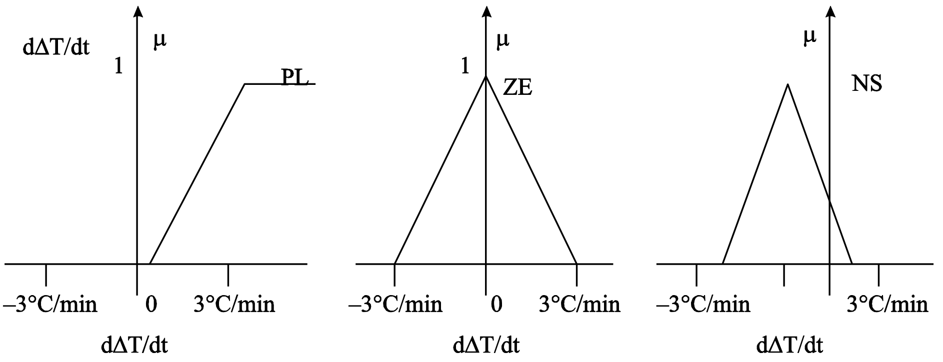


Fig. 9.23 Rate of Change in temperature difference

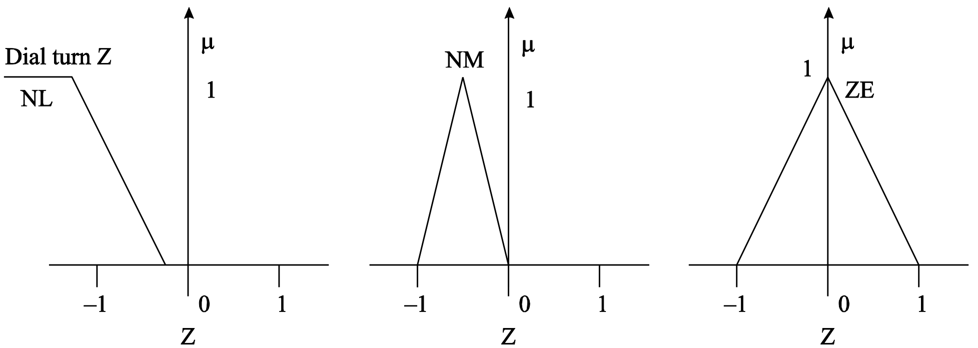


Fig. 9.24 Turn dials output

9.4.3 Case study

Consider the system inputs, ΔT = 2.5°C and dΔT/dt = -1°C/min.

Here the fuzzification of system inputs refer to Figure 9.25 has been directly done by noting the membership value corresponding to the system inputs. The rule strengths of rules 1, 2, 3 choosing the minimum of the fuzzy membership value of the antecedents are 0, 0.1 and 0.6 respectively. Figure 9.25 The defuzzification of the fuzzy output shown in figure 9.26.

Z = -0.2 for ΔT = 2.5°C and y = -1°C/min.

Hence, the dial needs to be turned in the negative direction, i.e., - 0.2 to achieve the desired temperature effect in the room.

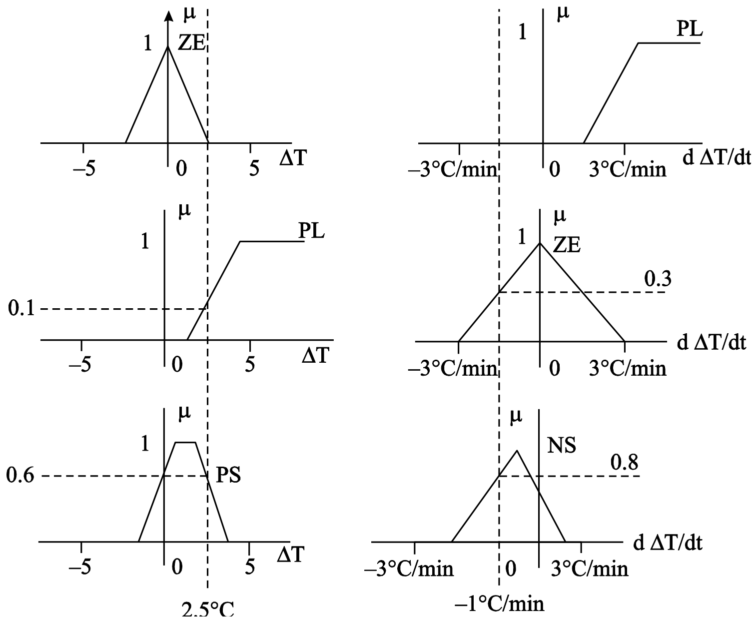


Fig. 9.25: Fuzzfication of inputs ΔT = 2.5°C, dΔT/dt = -1°C/min

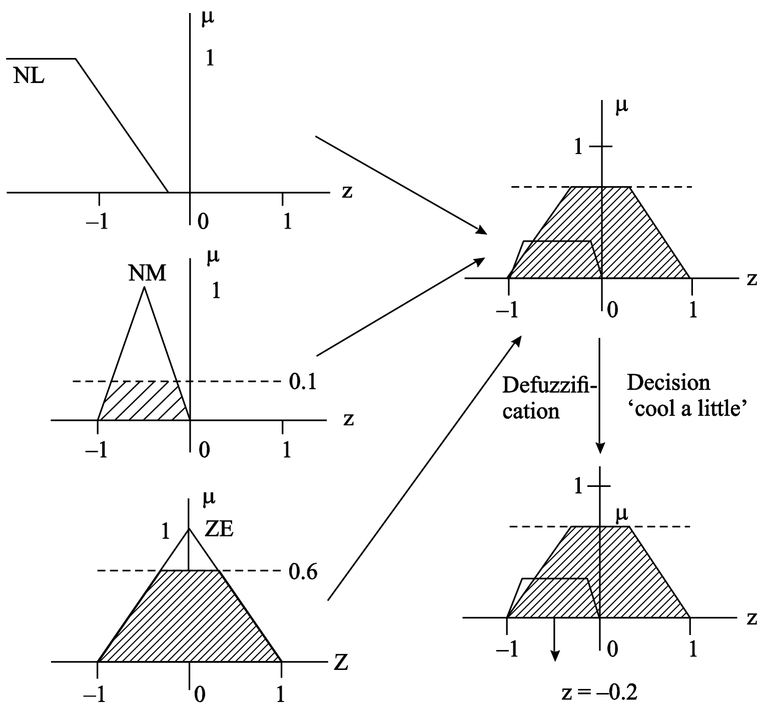


Fig. 9.26 Defuzzification of fuzzy outputs for z (turn of the dial)

9.5 BLOOD PRESSURE CONTROLLER DURING ANESTHESIA

9.5.1 Anesthesia overview

Here, a fuzzy logic controller was used to control mean arterial pressure (MAP), which was taken as a measure of the depth of anesthesia. A biological process like anesthesia has a non-linear, time varying structure and time varying parameters modelling it suggests the use of rule-based controllers like fuzzy controllers. The design process here was iterative and the reference points of the membership functions as well as the linguistic rules were determined by trial and error. The control loop has the following structure.

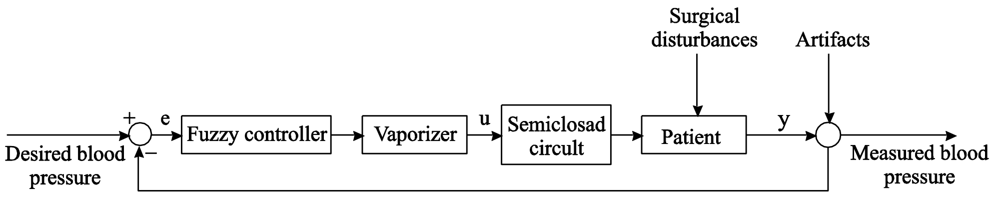


Fig. 9.27: Control loop for the control of depth of anesthesia

There are two different kinds of disturbances:

(1) Surgical disturbances

(2) Measurement noise and artifacts

The depth of anesthesia is controlled by using a mixture of drugs that are injected intravenously or inhaled as gases. Most of these agents decrease MAP. Among the inhaled gases, isofluorane is widely used, most often in a mixture of 0 to 2 percent by volume of isofluorane in oxygen and/or nitrous oxide. The relationship between
the inflow concentration of isofluorane u(t) and the resulting blood pressure y(t) is modeled as the sum of two first order terms, each with a pure time delay.

The step response h(t) corresponding to a unit step input can be written as follows:

$$h(t) = f_1(t) + f_2(t), \qquad \dots (9.1)$$

$$f_1 = k_1\alpha_1 \exp[-\alpha_1(t - t_1)], \text{ for } t > t_1 \qquad \text{and} \qquad \dots (9.2)$$

$$f_2 = k_2\alpha_2 \exp[-\alpha_2(t - t_2)], \text{ for } t > t_2 \qquad \dots (9.3)$$

where, $k_1 = -3$, $k_2 = -7.3$, $t_1 = 23$ s, $t_2 = 1.1$ s, $\alpha_1 = 0.01$, $\alpha_2 = 0.006$

Assume that $t_1 = c_1T$ and $t_2 = c_2T$ and use z-transform

Let Y(z) and U(z) are z-transforms of the output blood pressure and input isofluorane concentration respectively. Then

$$Y(z) = \sum_{i=1}^{2} k_i\alpha_i[1 - z^{-1}\exp(-\alpha_i T)]^{-1} z^{-C_i}\, u(z)$$

$$\frac{b_1 z^{-C_1} + b_2 z^{-1-C_1} + b_3 z^{-C_2} + b_4 z^{-1-C_2}}{1 + a_1 z^{-1} + a_2 z^{-2}} u(z) \qquad \dots (9.4)$$

where, $b_1 = k_1\alpha_1,$

$b_2 = -k_1\ \alpha_1 \exp(-\alpha_2 T),$

$b_3 = k_2\ \alpha_2,$

$b_4 = -k_2\ \alpha_2 \exp(-\alpha_1 T),$

$a_1 = -\exp(-\alpha_1 T) - \exp(-\alpha_2 T)$ and

$a_2 = \exp(-\alpha_1 T) \exp(-\alpha_2 T)$

The equation (9.4) in recursive form can be written as

$T = 10s$, $a_1 = -1.221$, $a_2 = 0.335$, $b_1 = -0.030$, $b_2 = 0.048$, $b_3 = -0.017$ and $b_4 = 0.041$.

It implies $y(kT) = -a_1 y((k - 1)T) - a_2 y((k - 2)T) + b_1 u((k - C_1)T)$

$$+ -b_2 u[(k - c_1 - 1)T] + b_1 u[(k - c_2)T] \quad + b_4 u[(k - c_2 - 1)T] \qquad \dots (9.5)$$

For the fuzzy controller, the error e(t) and the interval of error $\int e(t)$ are used for determination of the control variable u(t). The error e(t) is defined as

$$e(t) = d(t)-y(t), \qquad \dots (9.6)$$

where d(t) is the desired blood pressure.

The linguistic rules that describe the anesthetist's action are given in the table 9.2.

Table 9.2 Linguistic rule

Rule number	**Input e**	**Input $\int$ e**	**Output u**
1	NS	-	PB
2	PS	-	PS
3	NB	-	PV
4	PB	-	ZE
5	ZE	ZE	PM
6	ZE	PS	PS
7	ZE	NS	PB
8	-	NB	PV
9	-	PB	ZE

9.5.2 Rule Base

Considering ,

NB = Negative big NS = Negative small

ZE = Zero PS = Positive small

PB = Positive big PV = Positive very big, we draw the following Table.

Table 9.3 The FAM table for this rule is as follows

e/$\int$ e	**NB**	**NS**	**ZE**	**PS**	**PB**
NB	PV	PV	PV	PV	PV/ZE
NS	PB/PV	PB	PB	PB	PB/ZE
ZE	PV	PB	PM	PS	ZE
PS	PS/PV	PS	PS	PS	PS/ZE
PB	ZE/PV	ZE	ZE	ZE	ZE

The membership function used in this case are bell shaped functions that can be described by the following equation

$$\eta = \exp\left[-k\,(\xi - \lambda)^2\right],$$

where ξ is the input value and λ is the shifting of the function in relation to zero.
The factor k gives the width of the bell.
The reference points for the membership functions are given in the following table 9.4.

Table 9.4

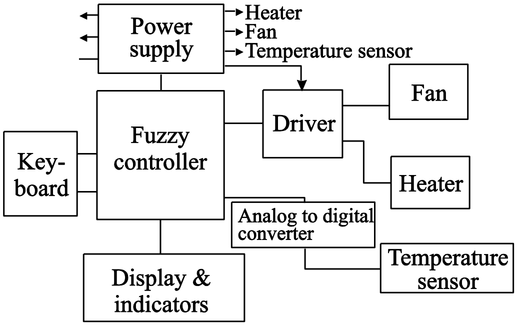
	Input e (mmHg)	**Input $\int e$ (mm Hg.s)**	**Output**	**Output u %**
NB	-10	-160	ZE	0
NS	-5	-90	PS	1
ZE	0	0	PM	2
PS	5	90	PB	3
PB	10	160	PV	4

9.6 LOW COST TEMPERATURE CONTROL

The fuzzy temperature controller is designed and implemented in microcontroller without using any special software tool. Unlike some fuzzy controllers with hundreds or even thousands of rules running on computer systems, a unique FLC using a small number of rules and simple implementation is demonstrated to solve a temperature control problem with unknown dynamics or variable time delays commonly found in industry. Also the final hardware is stand-alone system rather than a PC (personal computer/laptop computer) based system that takes control decision based on special software tools running on it and hence the design approach presented in the Table 9.5 for this example minimizes the total cost of hardware and software design. The control result can be improved by resizing the fuzzy sets and finer tuning for the membership functions.

9.6.1 System model description

Low cost temperature control using fuzzy logic system block diagram shown in the figure 9.28 and 9.29 in this system set point of the temperature is given by the operator using 4X4 keypad. LM35 temperature Sensor senses the current temperature. Analog to digital converter convert analog value into digital value and give to the Fuzzy controller. Controller calculates error between set point value and current value and consider as Input function of fuzzy logic. By fuzzification process controller calculate it membership. After in rule base and inference system output membership value calculated. Defuzzification process calculates actual value of PWM for heater and fan which is output of the temperature control system. The process comprises of a heater, fan and a temperature sensor. The amount of current passing through the coil decides the temperature of the thin metal plate. Temperature detection of this metal plate can be done by dedicated temperature sensors. A fan is placed near to the heating mechanism. Amount of power delivered to both heater and fan can be controlled by passing a command through serial port via microcontroller. Now, microcontroller generate PWM (Pulse Width Modulation) signal for the MOSFET to deliver desired amount of power to fan and heater. It could thus be used as a small plant readily available for various experimentation and study purpose.



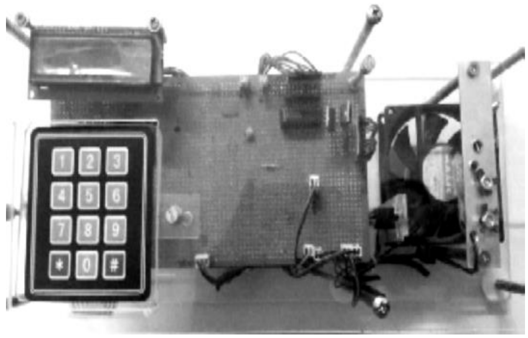


Fig 9.28: Block diagram of temperature control system Fig:9.29 Temperature control system

9.6.2 Fuzzification of Input

In the fuzzification process, a real scalar value changes into a fuzzy value. Arrangements of Fuzzy variables ensure that real values get translated into fuzzy values. After translating those real values into fuzzy values, the possible outcome is called "linguistic terms". The input linguistic variables for Fuzzy Logic Temperature Controller (FLTC) suggest two things. First it shows linguistically the difference between the set point and second it also express the measured and calculated signals from a temperature sensor. Input to FLTC is Error= (Set point-Temperature sensed). For fuzzified input, two functions including trapezoidal and triangular are used. To determine the range of fuzzy variables according to the crisp inputs is the primary requirement for proper running of the fuzzier program.Temperature difference which was sensed previously, is restricted to positive value. The following fuzzy sets are used: NEG =negative, SNEG=small negative, ZERO= zero, SPOZ=small positive, POZ= positive. Table suggests the Membership function for input linguistic variable. Membership function for input linguistic variable. To include the linguistic variable negative (Neg) to a microcontroller, transformation of the Pictorial representation into substantive code is needed.

Table 9.5: Input linguistic variables

No.	Crisp Input Range (Error = Set Point – Current Temperature)	Fuzzy Variable Name
1	-15 to -50	NEG
2	0 to +30	SNEG
3	-15 to +15	ZERO
4	0 to +30	SPOZ
5	+15 to +50	POZ

9.6.3 Fuzzy Membership Functions for Outputs

The output linguistic variables express linguistically the applied values to the FLTC actuators for temperature control. Present study considered typically one output variable, which is a PWM Wave for fan and Heater. In this case it is essential to attribute fuzzy memberships to yield variable which has to be identical to the input variable. The fuzzy sets used for PWM Wave are Z = zero, L = large, M = medium, H = high, VH = very high.

Table 9.6: Output linguistic variables

No	Fuzzy variable	Range output	Corresponding Fuzzy variable name
1	165.75 to 255	65% to 100%	VH
2	127 to 204	50% to 80%	H

3	165.75 to 89.25	65% To 35%	M
4	127 to 51	50% to 20%	L
5	89.25 to 0	35% to 0%	Z

9.6.4 Rule base

Once the current values of the input variables are fuzzified, the fuzzy controller continues with the phase of "making decisions" or deciding what actions to take to bring the temperature to its setpoint value. For the action to be initiated the measures are minimal time as well as minimal temperature. The restraint policy of a Fuzzy Control System is comprised by the rule blocks. In the rules 'IF' part depicts the situation for which the rules are projected. The following 'THEN' part delineates the reaction of the fuzzy system in this state. The control policy of heater is structurally formulated according to fuzzy rules. For example, rule 1 "If error is NEG, then firing angle is Z".

Table 9.7: Fuzzy rules

No	Fuzzy variable	Range output	Corresponding Fuzzy variable name
1	165.75 to 255	65% to 100%	VH
2	127 to 204	50% to 80%	H
3	165.75 to 89.25	65% To 35%	M
4	127 to 51	50% to 20%	L
5	89.25 to 0	35% to 0%	Z

9.6.5 Defuzzification

The outcome of defuzzification has to be in a numeric form so that it defines the PWM Wave of the MOSFET which is used to force the fan and heater. Out of the number of ways to execute defuzzification; in the given scenario, weighted average defuzzification is the best technique to obtain the crisp output. It can be further described by following equation;

$$D = \frac{\sum_{i=1}^{7} P(i).W(i)}{\sum_{i=1}^{7} W(i)}, where$$

P[i] = The extremum value of i^{th} output membership function and W[i] =The weight associated with i^{th} rule.

The fuzzy variable can be converted into crisp output using C code fragment. One example of that is given below

$\{ \; z = \frac{((f_1 * d_1) + (f_2 * d_2) + (f_3 * 127) + (f_4 * 89) + (f_5 * d_5) + (f_6 * d_6) + (f_7 * d_7))}{(f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7)}$ PWM=z;}

9.6.6 Outcome of temperature control using fuzzy logic

The Temperature in the case study is as follow:

Set temperature: 45^0 C, Current temperature: 46^0 C, then

Error = SP – CV = 45 - 46 = -1

Rule base Follow: Rule-2(SNEG) and Rule-3(ZERO)

Therefore, Fuzzilization value f_2=0.04, f_3=0.933and Defuzzilization value Z * = 130.95

The Duty Cycle of FAN speed = 51% and The Duty Cycle of HEATING COIL current = 49%

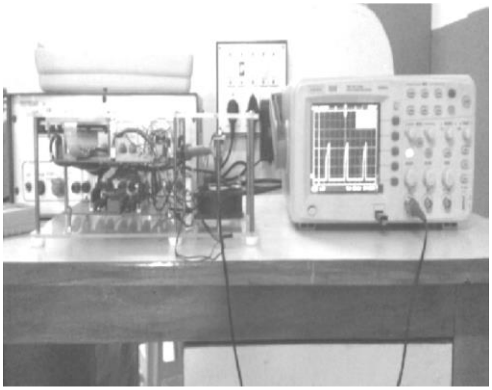


Fig 9.29: Overall view of system

Fig 9.30: LCD display

9.7 AIRCRAFT LANDING CONTROL PROBLEM

We will conduct a simulation of the final descent and landing approach of an aircraft. The desired downward velocity is proportional to the square of the height Thus, at higher altitudes a large downward velocity is desired. As the height (altitude) diminishes, the desired downward velocity gets smaller and smaller. In the limit, as the height becomes vanishingly small, the downward velocity also goes to zero. In this way, the aircraft will descend from altitude promptly but will touch down very gently to avoid damage.

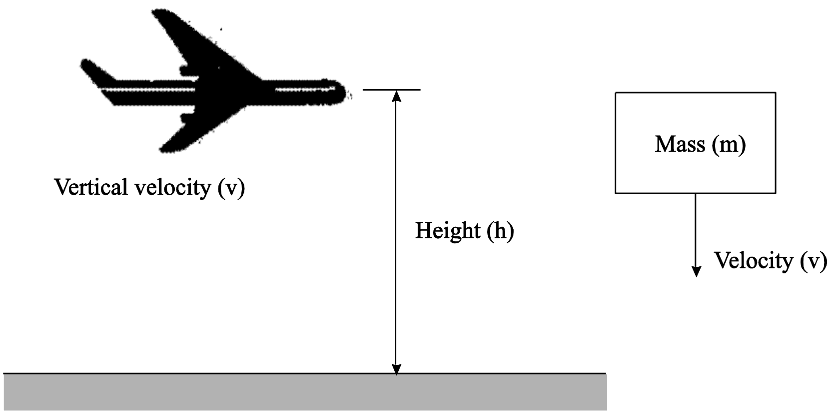


Fig. 9.31: Aircraft landing control problem

The two state variables for this simulation are be the height above ground, h, and the vertical velocity of the aircraft, v. The control output is a force that when applied to the aircraft, it alters its height, h, and velocity v. The differential control equations are loosely derived as follows:

Mass m moving with velocity v has momentum p = mv. If no external forces are applied, the mass continues in the same direction at the same velocity, v. If a force f is applied over a time interval Δt, a change in velocity of $\Delta v = f\Delta t/m$ is affected.

If we let $\Delta t = 1.0$ (sec) and $m = 1.0$ (N-sec^2/m), we obtain $\Delta v = f(N)$, or the change in velocity is proportional to the applied force.

In differential notation, we get $v_{i+1} = v_i + f_i$ and $h_{i+1} = h_i + v_i(1)$
where v_{i+1} is the new velocity, v_i is the old velocity, h_{i+1} is the new height and h_i is the old height.
These two "control equations" define the new value of the state variables v and h in response to control input and the previous state variable value. Now, following the same procedure, we construct membership functions for the height, h, the vertical velocity, v and the control force, f :
Step 1. Define membership functions for state variables as shown in the tables and figure given below

Table 9.8 Membership values for height

	Height (m)										
	0.0	100	200	300	400	500	600	700	800	900	1000
Large (L)	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.8	1.0
Medium (M)	0.0	0.0	0.0	0.0	0.2	0.4	0.6	0.8	1.0	0.8	0.6
Small (S)	0.4	0.6	0.8	1.0	0.8	0.6	0.4	0.2	0.0	0.0	0.0
Near zero (NZ)	1.0	0.8	0.6	0.4	0.2	0.0	0.0	0.0	0.0	0.0	0.0

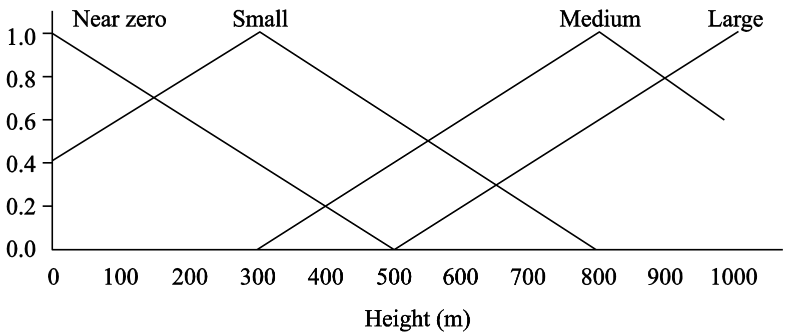


Fig. 9.32: Height, h, partitioned Membership values for velocity

Table 9.9 Vertical Velocity

	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
Up Large (UL)	0	0	0	0	0	0	0	0	0	0.5	1	1	1
Up Small (US)	0	0	0	0	0	0	0	0.5	1	0.5	0	0	0
Zero(Z)	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
Down Small (DS)	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
Down Large (DL)	1	1	1	0.5	0	0	0	0	0	0	0	0	0

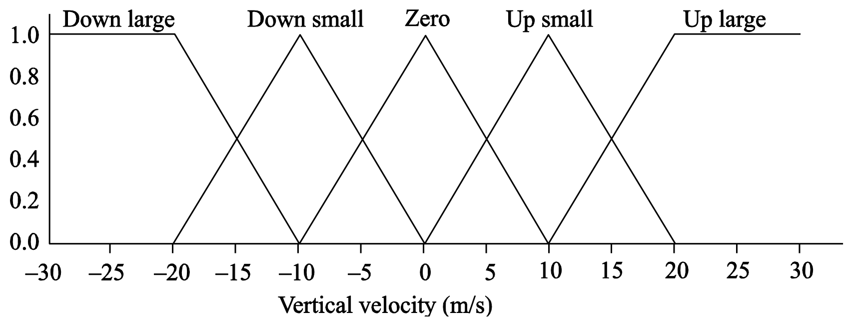


Fig. 9.33: Velocity, v, partitioned

Step2. Define a membership function for the control output, as shown in table and figure below.

Table 9.10 Membership values for velocity

	Control force (N)												
	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30
Up Large (UL)	0	0	0	0	0	0	0	0	0	0.5	1	1	1
Up Small (US)	0	0	0	0	0	0	0	0.5	1	0.5	0	0	0
Zero(Z)	0	0	0	0	0	0.5	1	0.5	0	0	0	0	0
Down Small (DS)	0	0	0	0.5	1	0.5	0	0	0	0	0	0	0
Down Large (DL)	1	1	1	0.5	0	0	0	0	0	0	0	0	0

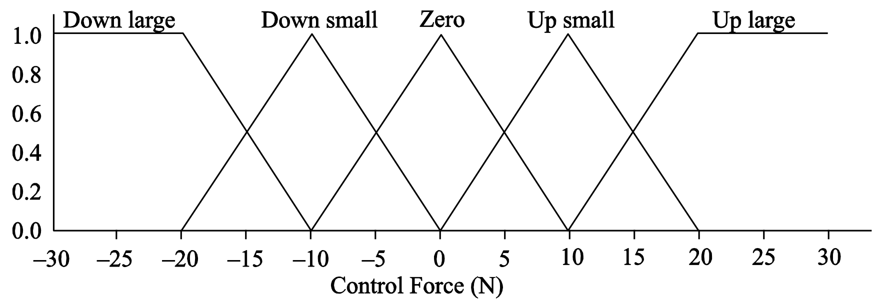


Fig. 9.34: Control force, f, partitioned

Step3. Define the rules and summarize them in an FAM table. The values in the FAM table, of course, are the control outputs.

Table 9.11 FAM table

Height	**Velocity**				
	DL	DS	Zero	US	UL
L	Z	DS	DL	DL	DL
M	US	Z	DS	DL	DL
S	UL	US	Z	DS	DL
NZ	UL	UL	Z	DS	DS

Step 4. Define the initial conditions and construct a simulation for four cycles. Since the task at hand is to control the aircraft's vertical descent during approach and landing, we will start with the aircraft at an altitude of 1000 feet, with a downward velocity of -20 ft/s. Then, using the following equation Control f_0 to be computed.

$$v_{i+1} = v_i + f_i \, , \qquad h_{i+1} = h_i + v_i$$

Initial height, h_0 = 1000 m and Initial velocity, v_0 = 20 m/s

Height h fires L at 1.0 and M at 0.6. and Velocity v fires only DL at 1.0.

Height	Velocity	Output
L(1.0) AND	DL(1.0) ⇒	Z(1.0)
M (0.6) AND	DL(1.0) ⇒	US (0.6)

We defuzzify using the centroid method and get f_0 = 5.8 N. This is the output force computed from the initial conditions.

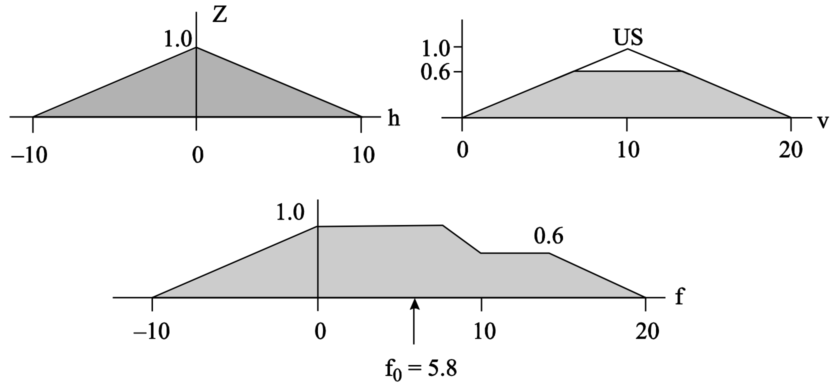


Fig. 9.35: Truncated consequents and union of fuzzy consequent for cycle 1

Now, we compute new values of the state variables and the output for the next cycle.

$h_1 = h_0 + v_0 = 1000 + (-20) = 980$ m

$v_1 = v_0 + f_0 = -20 + 5.8 = -14.2$ m/s

Height, h_1 = 980 m fires L at 0.96 and M at 0.64.

Velocity, v_1 = -14.2 m/ s fires DS at 0.58 and DL at 0.42.

Height	Velocity	Output
L(0.96) AND	DS(0.58) ⇒	DS(0.58)
L(0.96) AND	DL(0.42) ⇒	Z(0.42)
M (0.64) AND	DS(0.58) ⇒	Z(0.58)
M (0.64) AND	DL(0.42) ⇒	US (0.42)

We find the centroid to be f_1 = - 0.5 N.

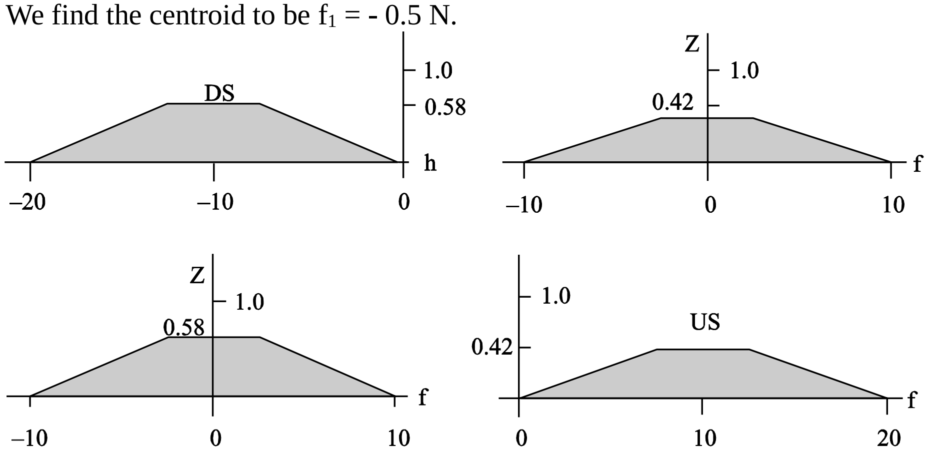


Fig. 9.36: Truncated consequents for cycle 2

We compute new values of the state variables and the output for the next cycle.

$h_2 = h_1 + v_1 = 980 + (-14.2) = 965.8$ m

$v_2 = v_1 + f_1 = -14.2 + (-0.5) = -14.7$ m/s

$h_2 = 965.8$ m fires L at 0.93 and M at 0.67.

$v_2 = -14.7$ m/s fires DL at 0.43 and DS at 0.57.

Height	Velocity	Output
L (0.93) AND	DL (0.43) $\Rightarrow$	Z(0.43)
L(0.93) AND	DS (0.57) $\Rightarrow$	DS(0.57)
M (0.67) AND	DL(0.43) $\Rightarrow$	US (0.43)
M (0.67) AND	DS(0.57) $\Rightarrow$	Z(0.57)

We find the centroid for this cycle to be f_2.= -0.4 N.

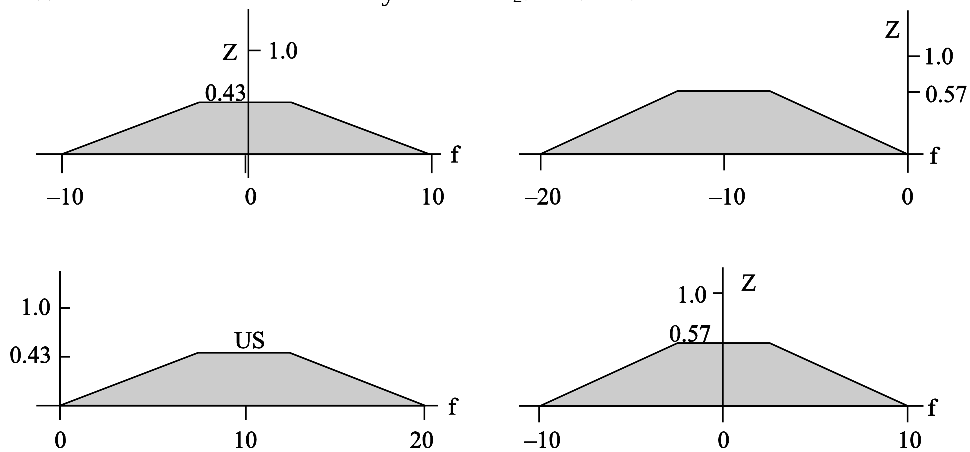


Fig. 9.37: Truncated consequents for cycle 3.

Again, we compute new values of state variables and output:

$h_3 = h_2 + v_2 = 965.8 + (-14.7) = 951.1$ m

$v_3 = v_2 + f_2 = -14.7 + (-0.4) = -15.1$ m/s

and for one more cycle, we get

$h_3 = 951.1$ m fires L at 0.9 and M at 0.7.

$v_3 = -15.1$ m/s fires DS at 0.49 and DL at 0.51.

Height	Velocity	Output
L(0.9) AND	DS(0.49) $\Rightarrow$	DS(0.49)
L(0.9) AND	DL(0.51) $\Rightarrow$	Z(0.51)
M(0.7) AND	DS(0.49) $\Rightarrow$	Z(0.49)
M(0.7) AND	DL(0.51) $\Rightarrow$	US (0.51)

Next, we compute the final values for the state variables to finish the simulation.

$h_4 = h_3 + v_3 = 951.1 + (-15.1) = 936.0$ m

$v_4 = v_3 + f_3 = -15.1 + 0.3 = -14.8$ m/s

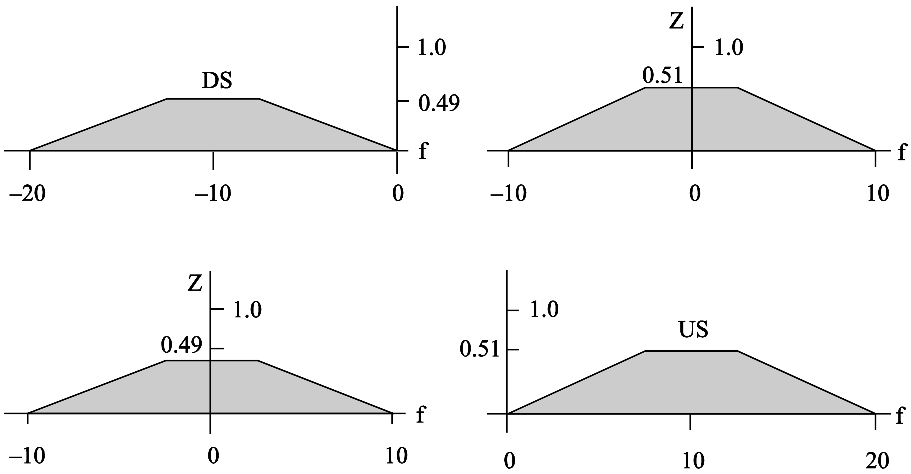


Fig. 9.38: Truncated consequents for cycle 4

The summary of the four-cycle simulation results is presented in table 9.12 given below.

Table 9.12: Summary of four-cycle simulation results

	Cycle 0	**Cycle 1**	**Cycle 2**	**Cycle 3**	**Cycle 4**
Height, m	1000.0	980.0	965.8	951.1	936.0
Velocity, m/s	-20	-14.2	-14.7	-15.1	-14.8
Control force	5.8	0.5	-0.4	0.3	

EXERCISE- 9

1. Given the health monitoring diagnostic system described in Section 9.1 of this chapter, write a computer program to simulate the following sensor data for the time $t \in [0,300.0]$ for every interval $\Delta t = 1$ second:

   (a) $x_1(t) = 98 + 4 \sin(\frac{\pi}{80} t)e(t)$

   $x_2(t) = 120 + 40 \sin(\frac{\pi}{60} t)e(t)$

   $x_3(t) = 70 + 60 \sin(\frac{\pi}{40} t)e(t)$

   (b) Implement the 27 rules mentioned and plot out the membership value for each rule against time t. What diagnostic conclusion can we draw from the plot of these 27 membership values?

   (c) Change the data in (a) to include additive random noise of Gaussian distribution. Repeat both steps (a) and (b).

   $x_1(t) = 98 + 4 \sin(\frac{\pi}{80} t)e(t) + n(t)$

   $x_2(t) = 120 + 40 \sin(\frac{\pi}{60} t)e(t) + n(t)$

   $x_3(t) = 70 + 60 \sin(\frac{\pi}{40} t)e(t) + n(t)$

2. Obtain a header format of digital image data stored under the bitmap format (with extension .bmp). Write out the format of this header in the following table:

File type (2 words)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)
data (1 word) data (1 word)	data (1 word)	data (1 word)

3. Obtain a digital image stored in the binary bitmap format (whose filename ended with the extension .bmp) described in **Exercise 2**. Write a computer program that simulates the off-focused data using the smooth (average) filter. Each off-focused image will be saved to the same binary bitmap format as the original image. Assuming that for each unit away from the correct focal point, the radius of the smooth filter is larger. The smooth filter is given as

$$I_{smooth}(n,m) = \frac{1}{(2R+1)^2} \sum_{i=n-R}^{n+N} \sum_{j=m-R}^{m+R} I_{original}(i,j).$$

4. Given a sequence of digital images generated in Exercise**3**, for each digital image, write a computer program to calculate the normalized sharpness index as

$$\sigma = \frac{1}{NM}\sum_{n=1}^{N}\sum_{n=1}^{M} e(n,m)$$

Is there another algorithm to calculate the sharpness index of a digital image? Outline and implement your own algorithm and compare the results to that of the above formula.

5. Implement the fuzzy control for the autofocus camera using the simulation results from Problems **Exercise 3** and **Exercise4**. Set the control step $\Delta\theta$ arbitrarily large and adjust the control steps using various adjustment parameters γ

BIBLIOGRAPHY

1. K. R. Ames. (1984)A Relational Approach to the Development of Expert Diagnostic Systems.
   Hampton, VA: NASA Langley Research Center.
2. N. Ayache (editor). Computer Vision, Virtual Reality, and Robotics in Medicine. Berlin, Germany: Springer-Verlag, 1995.
3. R. C. Berkan and S. L. Trubatch. Fuzzy Systems Design Principles:Building Fuzzy IF-THEN Rule Bases. New York, NY: IEEE Press, 1997.
4. K. S. Fu, R. C. Gonzalez, and C. S. G. Lee. Robotics: Control,Sensing, Vision, and Intelligence. New York, NY: McGraw-Hill, 1987.
5. S. E. Gordon. Expert Systems: Their Impact on Performance and Cognitive Strategies in Diagnostic Inference. Brooks Air Force Base,TX: Air Force Human Resources Laboratory, Air Force Systems Command Report,1987.
6. M. Klein and L. B. Methlie. Expert Systems: a Decision Support Approach: with applications in Management and Finance. Reading, MA: Addison-Wesley Publishing Company, 1990.
7. F. A. Lootsma. Fuzzy Logic for Planning and Decision Making. Boston, MA: Kluwer Academic Publishers, 1997.
8. R. W. Miller. Servomechanisms: Devices and Fundamentals. Reston, VA: Reston Publishing Company ,1977.
9. C. V. Negoita. Expert Systems and Fuzzy Systems. Menlo Park, CA: Benjamin/Cummings Publishing Company, 1985.
10. M. J. Patyra and D. M. Mlynek (editors). Fuzzy Logic: Implementation and Applications. New York, NY: Wiley, 1996.

Index