RÉPUBLIQUE ALGÉRIÈNNE DÉMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DE TECHNOLOGIE
DEPARTEMENT DE GENIE ELECTRIQUE

Mémoire de fin d'études en vue de l'obtention du diplôme de

MASTER

Spécialité : Electromécanique

Fuzzy modeling of the takagi-sugono type with two different techniques of clustering algorithms

---

Proposé et dirigé par :
Dr. Abdelhafid BENYOUNES

Présenté par :
Nouaoui tahar
HERIZI Kabes Eddine

Nº d'ordre : 052

In the memory of my aunt,

To my parents,

To my brothers and sisters.

Content

Content 3
Figures and tables list 6
Glossary 7
General Introduction 8
Chapter I 9
Concept of fuzzy logic 9
I.1.Introduction 10
I.1.1 Why Fuzzy Systems? 10
I.1.2 What Are Fuzzy Systems? 11
I.1.3 Where Are Fuzzy Systems Used and How? 14
I.2.The Mathematics of Fuzzy Systems and Control 14
I.2.1.Introduction 14
I.2.2.Fuzzy Sets and Basic Operations on Fuzzy Sets 14
I.2.2.1 From Classical Sets to Fuzzy Sets: 14
I .2.2.2 Basic Concepts Associated with Fuzzy Set 15
I .2.2.3 Operations on Fuzzy Sets 17
I .3. Further Operations on Fuzzy Sets 17
I .3.1.introduction: 17
I .3.2 Fuzzy Complement 18
I .4.Linguistic Variables and Fuzzy IF-THEN Rules 18
I .4.1.From Numerical Variables to Linguistic Variables 18
I .4.2.Linguistic Hedges: 19
I .4.3 Fuzzy IF-THEN Rules 20
I .4.3.1.Introduction: 20
I .4.3.2.Fuzzy propositions: 20
I .4.3.3. Interpretations of Fuzzy IF-THEN Rules 21
I .5.Fuzzy Rule Base and Fuzzy Inference Engine 23
I .5.1.Introduction: 23
I .5.2 Fuzzy Rule Base: 23
I .5.2.1 Structure of Fuzzy Rule Base: 23
I .5.2.2 Properties of Set of Rules: 24

I .5.3.Fuzzy lnference Engine: 24
I .5.3.1 Composition Based lnference:..24
I .5.3.2 Individual-Rule Based lnference: 25
I .5.3.3 The Details of Some lnference Engines:..25
I .6.Fuzzifiers and Defuzzifiers:..26
I .6.1.Introduction:...26
I .6.2.Fuzzifiers: 26
I .6.3.Defuzzifiers:...27
I .6.3.1 Center of gravity Defuzzifier: 27
I .6.3.2 Center Average Defuzzifier: 28
I .6.3.3 Maximum Defuzzifier...29
I .6.3.4. Comparison of the Defuzzifiers: 30
I .7.Summary and further readings:..31
Chapter II...32
Fuzzy modeling..32
II .1.Introduction: 33
II.2. Fuzzy Clustering:...33
II.2.1.Cluster analysis 33
II.2.1.1 classic partition and fuzzy partition...33
II.2.2. Fuzzy classification algorithms 34
II.2.2.1. Clustering algorithm of the type C-means (FCM) 34
II.2.2.2.Algorithm of Gustafson-Kessel (GK) 36
II.3. Fuzzy modeling of the Takagi-sugono type:...39
II.3.1. Introduction: 39
II.3.2. Product space clustering for identification 39
II.3.2.1 Outline of the Approach 39
II.3.2.2 Structure Selection : 42
II.3.2.2.1 The Nonlinear Regression Problem...43
II.3.2.2.2. Input-output Black-box Models’: 43
II.3.2.2.3 State-space Framework:...44
II.3.2.2.4 Semi-mechanistic Modeling: 44
II.3.2.3 Identification by Product-space Clustering: 45

II.3.2.4.Choice of Clustering Algorithms: 46
II.3.2.4.1.Clustering with Adaptive Distance Measure: 46
II.3.2.4.2.Fuzzy c-lines and c-elliptotypes: 47
II.3.2.4.3.Fuzzy c-regression Models: 47
II.3.2.5.Determining the Number of Clusters: 47
II.3.2.5.1.Cluster Validity Measures: 48
II.3.2.5.2.Compatible Cluster Merging: 48
II.3.2.6. Takagi-sugono modeling: 48
II.3.2.7.Rule Base Simplification Algorithm. 50
II.3.2.8.Model Validation. 50
II.4.Summary and further readings: 51
Chapitre 3: 52
Fuzzy modeling of SISO and MIMO systems via C-means and GK-means methods of clustering 52
III.1.Introduction. 53
III.2.Application of SISO system. 53
III.2.1.description. 53
III.2.2.simulation. 53
III.2.3. Results comparison between C-means and GK-means...55
III.3.Application of MIMO system. 55
III.3.1.description. 55
III.3.2.simulation. 56
III.3.3. Results comparison between C-means and GK-means...57
III.4.Conclusion. 57
General conclusion. 58
Bibliographical references 59
Annex A: MATLAB code of Gustafson-Kessel (GK) Algorithm..60
Annex B: MATLAB code of C-means Algorithm 61

Figures and tables list

Figure I.1 horizontal axis represents the speed of the car and the vertical axis represents the membership value for "high." 11
Figure. I. 2 horizontal axis represents the force applied to the accelerator and the vertical axis represents the membership value for "less." 12
Figure . I.3 Basic configuration of pure fuzzy systems. 12
Figure . I.4 Basic configuration of Takagi-Sugeno-Kang fuzzy system. 13
Figure. I. 5 Basic configuration of fuzzy systems with fuzzifier and defuzzifier. 13
Figure .I.6 Membership function for fuzzy set "several." 15
Figure . I.7 possible membership function to 16
Figure . I.8 Centers of some typical fuzzy sets 16
Figure .I.9 The speed of a car as a linguistic variable 19
Table I.4.1 Interpretations of Fuzzy IF-THEN Rules 21
Figure . I.10 A multi-input-multi-output fuzzy system can be decomposed into a collection of multi-input-singleoutput fuzzy systems. 23
Figure I. .11. A graphical representation of the center of gravity defuzzifier. 28
Figure I. .12. A graphical representation of the center average defuzzifier 29
Table I.6.1Comparison of the center of gravity, center average, and maximum defuzzifiers with respect to plausibility, computational simplicity, and continuity. 30
Figure .I.13. A graphical representation of the maximum defuzzifiers. In this example, the mean of maxima defuzzifier gives a result that is contradictory to the maximummembership intuition 30
Figure .II.1.The clusters and its centers 36
Figure. II.2. The clusters and its centers 38
Figure . II.4...Overview of the identification approeach based on fuzzy clustering 41
Figure . II.5. Merging of compatible clusters. 48
Figure . II.6 49
Figure .III.1.Input data:Valve position 53
Figure . III.2. Output data (pressure) 54
Figure .III.3. the real output and the estimated outputs. 54
Figure . III.4. Membership functions of siso system with C-means and GK-means 54
Figure . III.5. Four cascaded tanks. 55
Figure . III.6. Input data (Water flow rate) 56
Figure . III.7. Ouput data (Tanks water levels) 56
Figure . III.8. Real and estimated outputs 57

Glossary

TS	Takagi-sugono fuzzy modeling type
VAF	Percentile variance accounted between two signals
RMS	Root-mean-squared error between two signals

General Introduction

This memoire addresses the best approach of the modeling of complex, nonlinear, or partially unknown systems by means of techniques based on fuzzy set theory and fuzzy logic. This approach, termed fuzzy modeling, is shown to be able to cope with systems that pose problems to conventional techniques, mainly due to nonlinearities and lack of precise knowledge about these systems….in Chapter.1.we will describe the meaning and the use of the fuzzy systems and go through its different fundamentals starting with the mathematics and the control of the fuzzy systems, the transition from classical to fuzzy sets and then the different fuzzy sets and some of the operations that can be applied on it including(the intersection, union and fuzzy compliment….), also we will learn the concept of linguistic variables and how it can describe the numerical variables in a more suitable way by using some terms like the linguistic hedges, Then we will explain the concepts of the fuzzy IF-THEN Rules and its different fundamentals(Fuzzy rule base structure, fuzzy inference engines…), Lastly we will explain in details the concept and the different types of Fuzzifiers and Defuzzfiers and we will end this chapter with a brief summary of its information.

In the Chapter.2.we will learn how to employ the concepts of fuzzy-set theory and fuzzy logic in the modeling of the different complex systems by explaining its different fundamentals and steps starting with the fuzzy clustering and some of its different clustering algorithms like the C-means and GK-means algorithm and then we will go through the detailed steps approached in the Takagi-sugono fuzzy modeling type including (data selection, structure selection ,fuzzy clustering, choice of the number of clusters, rule base simplification Linguistic approximation, modal validation…)

Then in chapter.3.we will present and compare the results of the fuzzy modeling of two different systems each individually (SISO and MIMO) by using two different ways of data clustering(C-means and GK-means).

Chapter I

Concept of fuzzy logic

I.1.Introduction

I.1.1 Why Fuzzy Systems?

According to the Oxford English Dictionary, the word "fuzzy" is defined as "blurred, indistinct; imprecisely defined; confused, vague." We ask the reader to disregard this definition and view the word "fuzzy" as a technical adjective. Specifically, fuzzy systems are systems to be precisely defined, and fuzzy control is a special kind of nonlinear control that also will be precisely defined. This is analogous to linear systems and control where the word "linear" is a technical adjective used to specify "systems and control;" the same is true for the word "fuzzy." Essentially, what we want to emphasize is that although the phenomena that the fuzzy systems theory characterizes may be fuzzy, the theory itself is precise.

In the literature, there are two kinds of justification for fuzzy systems theory:The real world is too complicated for precise descriptions to be obtained,therefore approximation (or fuzziness) must be introduced in order to obtaina reasonable, yet trackable, model.

As we move into the information era, human knowledge becomes increasingly important.We need a theory to formulate human knowledge in a systematic manner and put it into engineering systems, together with other information like mathematical models and sensory measurements.

The first justification is correct, but does not characterize the unique nature of fuzzy systems theory. In fact, almost all theories in engineering characterize the real world in an approximate manner. For example, most real systems are nonlinear,but we put a great deal of effort in the study of linear systems. A good engineering theory should be precise to the extent that it characterizes the key features of the real world and, at the same time, is trackable for mathematical analysis. In this aspect, fuzzy systems theory does not differ from other engineering theories.

The second justification characterizes the unique feature of fuzzy systems theory and justifies the existence of fuzzy systems theory as an independent branch in engineering. As a general principle, a good engineering theory should be capable of making use of all available information effectively. For many practical systems,important information comes from two sources: one source is human experts who describe their knowledge about the system in natural languages; the other is sensory measurements and mathematical models that are derived according to physical laws.

An important task, therefore, is to combine these two types of information into system designs. To achieve this combination, a key question is how to formulate human knowledge into a similar framework used to formulate sensory measurements and mathematical models. In other words, the key question is how to transform a human knowledge base into a mathematical formula. Essentially, what a fuzzy system does is to perform this transformation. In order to understand how this transformation is done, we must first know what fuzzy systems are.

I.1.2 What Are Fuzzy Systems?

Fuzzy systems are knowledge-based or rule-based systems. The heart of a fuzzy System is a knowledge base consisting of the so-called fuzzy IF-THEN rules. A fuzzy IF-THEN rule is an IF-THEN statement in which some words are characterized by continuous membership functions. For example, the following is a fuzzy IF-THEN rule:

IF the speed of a car is high, THEN apply less force to the accelerator (1.1)

Where the words "high" and "less" are characterized by the membership functions shown in Figs.1 and 2, respectively.' A fuzzy system is constructed from a collection of fuzzy IF-THEN rules. Let us consider **Example1.1**.

Example 1.1. Suppose we want to design a controller to automatically control the speed of a car. Conceptually, there are two approaches to designing such a controller: the first approach is to use conventional control theory, for example, designing a PID controller; the second approach is to emulate human drivers, that is, converting the rules used by human drivers into an automatic controller. We now consider the second approach. Roughly speaking, human drivers use the following three types of rules to drive a car in normal situations:

IF speed is low, THEN apply more force to the accelerator (1.2)

IF speed is medium, THEN apply normal force to the accelerator (1.3)

IF speed is high, THEN apply less force to the accelerator (1.4)

where the words "low," "more," "medium," "normal," "high," and "less" are characterized by membership functions similar to those in Figs.I.1-2. Of course, more rules are needed in real situations. We can construct a fuzzy system based on these rules. Because the fuzzy system is used as a controller, it also is called a fuzzy controller.

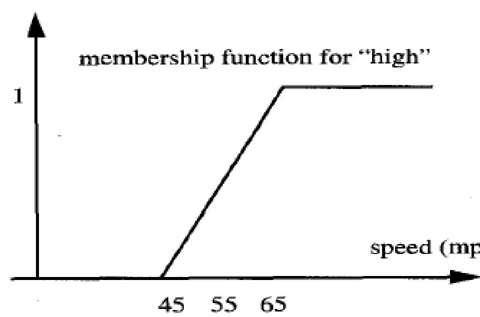

Figure I.1 horizontal axis represents the speed of the car and the vertical axis represents the membership value for "high."

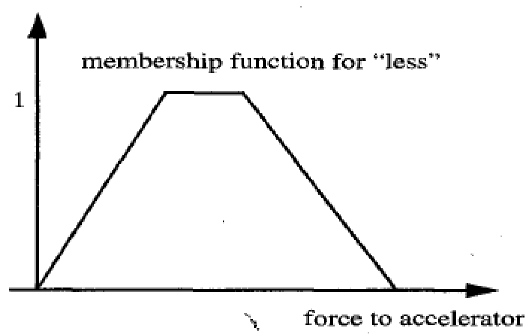

Figure. I. 2 horizontal axis represents the force applied to the accelerator and the vertical axis represents the membership value for "less."

In summary, the starting point of constructing a fuzzy system is to obtain a collection of fuzzy IF-THEN rules from human experts or based on domain knowledge.

The next step is to combine these rules into a single system. Different fuzzy systems use different principles for this combination. So the question is: what are the commonly used fuzzy systems?

There are three types of fuzzy systems that are commonly used in the literature: (i) pure fuzzy systems, (ii) Takagi-Sugeno-Kang (TSK) fuzzy systems, and (iii) fuzzy systems with fuzzifier and defuzzifier. We now briefly describe these three types of fuzzy systems.

The basic configuration of a pure fuzzy system is shown in Fig.I. 3. The fuzzy rule base repr-esents the collection of fuzzy IF-THEN rules. For examples, for the car controller in Example 1.1, the fuzzy rule base consists of the three rules (1.2)-(1.4), The fuzzy inference engine combines these fuzzy IF-THEN rules into a mapping from fuzzy sets in the input space $U \subset R^n$to fuzzy sets in the output space $U \subset R$ based on fuzzy logic principles. If the dashed feedback line in Fig. I. 1.3 exists, the system becomes the so-called fuzzy dynamic system.

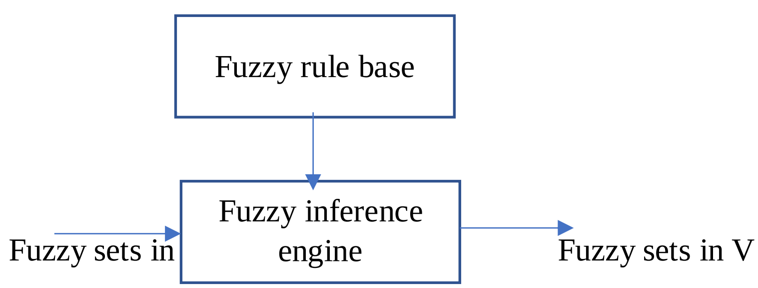

Figure . I.3 Basic configuration of pure fuzzy systems.

fuzzy sets (that is, words in natural languages), whereas in engineering systems the inputs and outputs are real-valued variables. To solve this problem, Takagi, Sugeno, and Kang (Takagi and Sugeno [I9851 and Sugeno and Kang [1988]) proposed another fuzzy system whose inputs and outputs are real-valued variables.

Instead of considering the fuzzy IF-THEN rules in the form of (1.1), the Takagi-Sugeno-Kang (TSK) system uses rules in the following form:

IF the speed x of a car is high, THEN the force to the accelerator is $y = cx$ (1.9)

where the word "high" has the same meaning as in (1.1), and c is a constant. Comparing (1.9) and (1.1) we see that the THEN part of the rule changes from a description using words in natural languages into a simple mathematical formula. This change makes it easier to combine the rules. In fact, the Takagi-Sugeno-Kang fuzzy system is a weighted average of the values in the THEN parts of the rules.
The basic configuration of the Takagi-Sugeno-Kang fuzzy system is shown in Fig. 4.

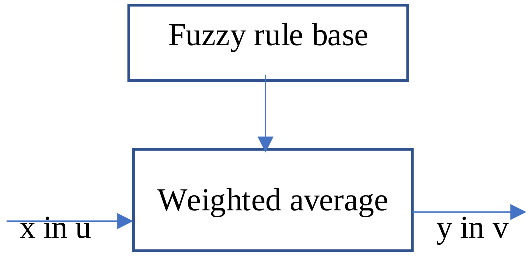

Figure . I.4 Basic configuration of Takagi-Sugeno-Kang fuzzy system.

The main problems with the Takagi-Sugeno-Kang fuzzy system are: (i) its THEN part is a mathe-matical formula and therefore may not provide a natural framework to represent human know-ledge, and (ii) there is not much freedom left to apply different principles in fuzzy logic, so that the versatility of fuzzy systems is not well-represented in this framework. To solve these pro-blems, we use the third type of fuzzy systems-fuzzy systems with fuzzifier and defuzzifier.
In order to use pure fuzzy systems in engineering systems, a simple method is to add a fuzzifier, which transforms a real-valued variable into a fuzzy set, to the input, and a defuzzifier, which transforms a fuzzy set into a real-valued variable, to the output. The result is the fuzzy system with fuzzifier and defuzzifier, shown in Fig. I.5. This fuzzy system overcomes the disadvantages of the pure fuzzy systems and the Takagi-Sugeno-Kang fuzzy systems. Unless otherwise specified, from now on when we refer fuzzy systems we mean fuzzy systems with fuzzifier and defuzzifier.

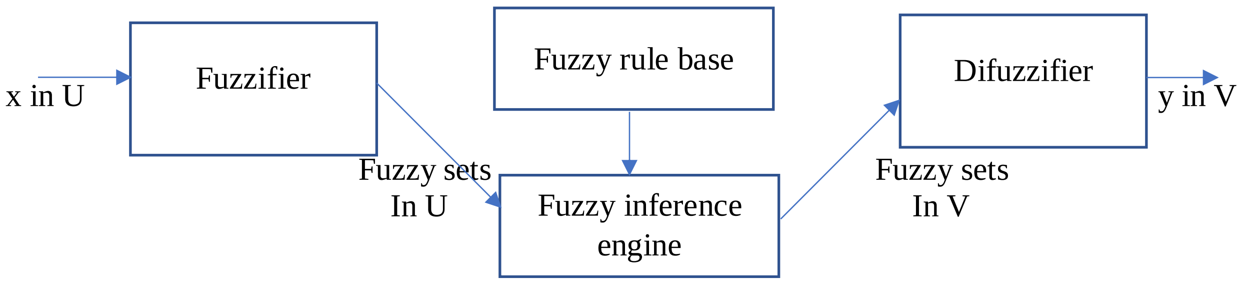

Figure. I. 5 Basic configuration of fuzzy systems with fuzzifier and defuzzifier.

I.1.3 Where Are Fuzzy Systems Used and How?

Fuzzy systems have been applied to a wide variety of fields ranging from control,signal processing, communications, integrated circuit manufacturing, and expert systems to business, medicine, psychology, etc. However, the most significant applications have concentrated on control problems. Therefore, instead of listing the applications of fuzzy systems in the different fields, we concentrate on a number of control problems where fuzzy systems play a major role.

I.2.The Mathematics of Fuzzy Systems and Control

I.2.1.Introduction

Fuzzy mathematics provide the starting point and basic language for fuzzy systems and fuzzy control. Fuzzy mathematics by itself is a huge field, where fuzzy mathematical principles are developed by replacing the sets in classical mathematical theory with fuzzy sets. In this way, all the classical mathematical branches may be "fuzzified." We have seen the birth of fuzzy measure theory, fuzzy topology, fuzzy algebra, fuzzy analysis, etc. Understandably, only a small portion of fuzzy mathematics has found applications in engineering. In the next five chapters, we will study those concepts and principles in fuzzy mathematics that are useful in fuzzy systems and fuzzy control.

In this section **,** we will introduce the most fundamental concept in fuzzy theory the concept of fuzzy,set-theoretical operations on fuzzy sets such as complement, union, and intersection will be studied in detail. We will study fuzzy relations and introduce an important principle in fuzzy theory the extension principle. Linguistic variables and fuzzy IF-THEN rules.

I.2.2.Fuzzy Sets and Basic Operations on Fuzzy Sets

I.2.2.1 From Classical Sets to Fuzzy Sets:

Let U be the unaverse of discourse, or universal set, which contains all the possible elements of concern in each particular context or application. Recall that a classical (crisp) set ***A,*** or simply a set A, in the universe of discourse U can be defined by listing all of its members (the last method) or by specifying the properties that must be satisfied by the members of the set (the rule method). The list method can be used only for finite sets and is therefore of limited use. The rule method is more general. In the rule method, a set A is represented as

$$\mathrm{A} = \{x \in U | x\ meets\ some\ condition\} \qquad (2.1)$$

There is yet a third method to define a set A the membership method, which introduces a zero one membership function (also called characteristic function, discrimination function, or indicator function) for ***A,*** denoted by $\mu_A(x)$, such that:

$$\mu_A = \begin{cases} 1\ if\ x \in A \\ 0\ if\ x \notin A \end{cases} \qquad (2.2)$$

The set ***A*** is mathematically equivalent to its membership function$\mu_A(x)$ onn the sense that knowing $\mu_A(x)$ is the same **as** knowing A itself.

Definition 1. A fuzzy *set* in a universe of discourse U is characterized by a membership function $\mu_A(x)$ that takes values in the interval [0, 11. Therefore, a fuzzy set is a generalization of a classical set by allowing the membership function to take any values in the interval [0, *11*. In other words, the membership function of a classical set can only take two values-zero and one, whereas the membership function of a fuzzy set is a continuous function with range [0, 11.
We see from the definition that there is nothing ‘’fuzzy’’ about a fuzzy set; it is simply a set with a continuous membership function.
A fuzzy set ***A*** in ***U*** may be represented as a set of ordered pairs of a generic element *x* and its membership value, that is,

$$\text{A} = \{(x, \mu_A(x))|x \in U\} \qquad (2.3)$$

When *U* is continuous (for example, ***U*** = R), ***A*** is commonly written as

$$\text{A} = \int_U \mu_A(x)/x \qquad (2.4)$$

where the integral sign does not denote integration; it denotes the collection of all points $x \in U$ with the associated membership function $\mu_A(x)$. When *U* is discrete,***A*** is commonly written as

$$\text{A} = \sum_U \mu_A(x)/x \qquad (2.5)$$

where the summation sign does not represent arithmetic addition; it denotes the collection of all points $x \in U$ with the associated membership function $\mu_A(x)$.

I .2.2.2 Basic Concepts Associated with Fuzzy Set

We now introduce some basic concepts and terminology associated with a fuzzy set. Many of them are extensions of the basic concepts of a classical (crisp) set, but some are unique to the fuzzy set framework.
Definition 2. The concepts of support, fuzzy singleton, center, crossover point, height, normal fuzzy set, a-cut, convex fuzzy set, and projections are defined as follows.
The support of a fuzzy set A in the universe of discourse U is a crisp set that contains all the elements of U that have nonzero membership values in A, that is,

$$\text{supp(A)} = \{x \in U|\mu_A(x) > 0\} \qquad (2.6)$$

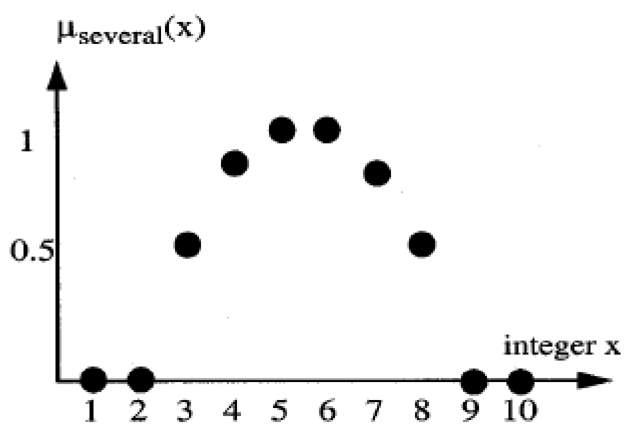

Figure .I.6 Membership function for fuzzy set "several."

where supp(A) denotes the support of fuzzy set A. For example, the support of fuzzy set "several" in Fig. 6 is the set of integers {3,4,5,6,7,8). If the support of a fuzzy set is empty, it is called an empty fuzzy set. A fuzzy singleton is a fuzzy set whose support is a single point in U.

The center of a fuzzy set is defined as follows: if the mean value of all pointsat which the membership function of the fuzzy set achieves its maximum value is finite, then define this mean value as the center of the fuzzy set; if the mean value equals positive (negative) infinite, then the center is defined as the smallest (largest) among all points that achieve the maximum membership value. Fig. I.7 shows the centers of some typical fuzzy sets. The crossover point of a fuzzy set is the point in U whose membership value in A equals 0.5.

The height of a fuzzy set is the largest membership value attained by any point.

An a-cut of a fuzzy set A is a crisp set A, that contains all the elements in U that have membership values in A greater than or equal to a, that is,

$$A_a = \{x \in U \mid \mu_A(x) \geq \alpha\} \qquad (2.7)$$

For example, for a = 0.3, the a-cut of the fuzzy set (2.11) (Fig. 7) is the crisp set[-0.7,0.7], and for a = 0.9, it is [-0.1,0.1].

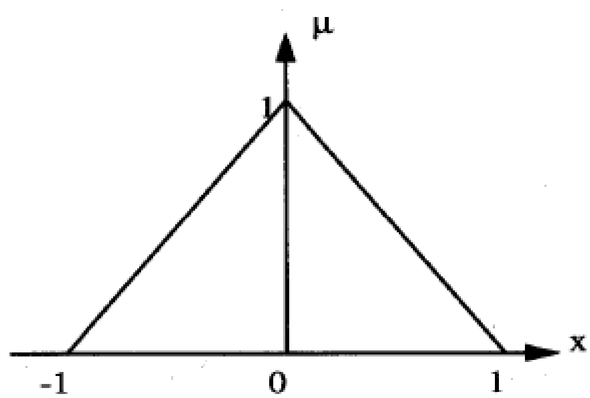

Figure . I.7 possible membership function to

When the universe of discourse U is the n-dimensional Euclidean space $\mathbb{R}^n$, the concept of set convexity can be generalized to fuzzy set. A fuzzy set A is said to be convex if and only if its a-cut A, is a convex set for any a in the interval (0, 1].

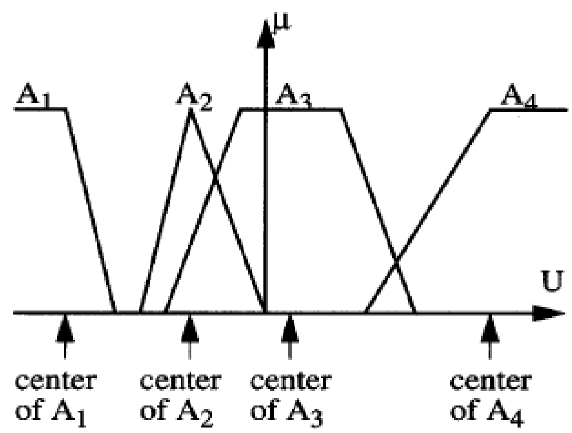

Figure . I.8 Centers of some typical fuzzy sets

I .2.2.3 Operations on Fuzzy Sets

The basic concepts introduced in Sections 2.2.1 and 2.2.2 concern only a single fuzzy set. In this section, we study the basic operations on fuzzy sets. In the sequel, we assume that A and B are fuzzy sets defined in the same universe of discourse U.

Definition 2.3. The equality, containment, complement, union, and intersection of two fuzzy sets A and B are defined as follows.

We say A and B are equal if and only if $\mu_A(x) = \mu_B(x)\ for\ all\ x \in U$ .We say B contains A, denoted by $A \subset B$, if and only if $\mu_A(x) \leq \mu_B(x)\ for\ all\ x \in U$ The complement of A is a fuzzy set $\bar{A}$ in U whose membership function is defined as

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \qquad (2.8)$$

The union of A and B is a fuzzy set in U , denoted by A ∪ B with the logical operator OR .Its membership function is defined as

$$\mu_{A \cup B}\ (x) = max[\mu_A(x), \mu_B(x)] \qquad (2.9)$$

The intersection of A and B is a fuzzy set A ∩ B in U with the logical operator AND. Its membership function is defined as

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] \qquad (2.10)$$

The reader may wonder why we use "max" for union and "min" for intersection; we now give an intuitive explanation. An intuitively appealing way of defining the union is the following: the union of A and B is the smallest fuzzy set containing both A and B. More precisely, if C is any fuzzy set that contains both A and B, then it also contains the union of A and B. To show that this intuitively appealing definition is equivalent to (2.9), we note, first, that A ∪ B as defined by (2.9) contains both A and B because $max[\mu_A, \mu_B] \geq \mu_A$and $max[\mu_A, \mu_B] \geq \mu_B$. Furthermore, if C is any fuzzy set containing both A and B, then $\mu_C \geq \mu_A$ and $\mu_C \geq \mu_B$. Therefore,

$\mu_C \geq max[\mu_A, \mu_B] = \mu_{A \cup B}(x)$, which means that A ∪ B as defined by (2.9) is the smallest fuzzy set containing both A and B. The intersection as defined by (2.10) can be justified in the same manner.

I .3. Further Operations on Fuzzy Sets

I .3.1.introduction:

In the previous section we introduced the following basic operators for complement, union, and intersection of fuzzy sets:

$$\mu_{\bar{A}}\ (x) = 1 - \mu_A\ (x) \qquad (3.1)$$

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] \qquad (3.2)$$

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \qquad (3.3)$$

We explained that the fuzzy set A U B defined by (3.2) is the smallest fuzzy set containing both A and B, and the fuzzy set A n B defined by (3.3) is the largest fuzzy set contained by both A and B. Therefore, (3.1)-(3.3) define only one type of operations on fuzzy sets. Other possibilities exist. For example, we may define A U B as any fuzzy set containing both A and B (not necessarily the smallest fuzzy set). In this section, we study other type of operators for complement.

I .3.2 Fuzzy Complement

Let c : [0,1] + [0,1] be a mapping that transforms the membership function of fuzzy set A into the membership function of the complement of A, that is,

$$c[\mu_A(x)] = \mu_{\bar{A}}(x) \qquad (3.4)$$

In the case of (3.1), $c[\mu_A(x)] = 1 - \mu_A(x)$. In order for the function c to be qualified as a complement, it should satisfy at least the following two requirements.

Axiom c1. c(0) = 1 and c(1) = 0 (boundary condition)

Axiom c2. For all $, b \in [0,1]$, if $a < b$,then $c(a) \geq c(b)$ (nonincreasing condition), where a and b denote membership functions of some fuzzy sets, say, $a = \mu_A(x)$ and $b = \mu_b(x)$.

Axiom c1 shows that if an element belongs to a fuzzy set to degree zero (one), then it should belong to the complement of this fuzzy set to degree one (zero).

Axiom c2 requires that an increase in membership value must result in a decrease or no change in membership value for the complement. Clearly, any violation of these two requirements will result in an operator that is unacceptable as complement

Definition 3. Any function c : $[0,1] \to [0,1]$ that satisfies Axioms c1 and c2 is called a fuzzy complement.

One class of fuzzy complements is the Sugeno class (Sugeno 119771) defined by

$$c_\lambda(a) = \frac{1-a}{1+\lambda a} \qquad (3.5)$$

where $\lambda \in (-1, \infty)$. For each value of the parameterλ, we obtain a particular fuzzy complement. It is a simple matter to check that the complement defined by (3.5)satisfies Axioms c1 and c2. Note that when λ = 0 it becomes the basic fuzzy complement (3.1)

Another type of fuzzy complement is the Yager class (Yager [1980]) defined by

$$c_w(a) = (1 - a^w)^{1/w} \qquad (3.6)$$

where $w \in (0, \infty)$. For each value of$w$, we obtain a particular fuzzy complement. It is easy to verify that (3.6) satisfies Axioms cl and c2. When w = 1, (3.6) becomes (3.1).

I .4.Linguistic Variables and Fuzzy IF-THEN Rules

I .4.1.From Numerical Variables to Linguistic Variables

In our daily life, words are often used to describe variables. For example, when we say "today is hot," or equivalently, "today's temperature is high," we use the word "high" to describe the variable "today's temperature." That is, the variable "today's temperature' takes the word "high" as its value. Clearly, the variable "today's temperature" also can take numbers like 25^o,35^o , etc., as its values. When a variable takes numbers as its values, we have a well-established mathematical framework to formulate it. But when a variable takes words as its values, we do not have a formal framework to formulate it in classical mathematical theory. In order to provide such a formal framework, the concept of linguistic variables was introduced.

Roughly speaking, if a variable can take words in natural languages as its values, it is called a linguistic variable. Now the question is how to formulate the words in mathematical terms? Here we use fuzzy sets to characterize the words. Thus, we have the following definition.

Definition 4. If a variable can take words in natural languages as its values, it is called a linguistic variable, where the words are characterized by fuzzy sets defined in the universe of discourse in which the variable is defined.

Example 4.1. The speed of a car is a variable x that takes values in the interval $[0, V_{max}]$, where V, is the maximum speed of the car. We now define three fuzzy sets "slow," "medium," and "fast" in $[0, V_{max}]$ as shown in Fig.9. If we view x as a linguistic variable, then it can take "slow," "medium" and "fast" as its values.

That is, we can say "x is slow," "x is medium," and "x is fast." Of course, x also can take numbers in the interval $[0, V_{max}]$ as its values, for example, x = 50mph,35mph, etc.

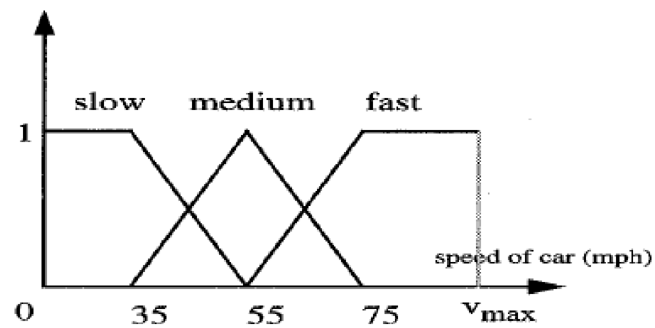

Figure .I.9 The speed of a car as a linguistic variable

Definition 4 gives a simple and intuitive definition for linguistic variables. In the fuzzy theory literature.

I .4.2.Linguistic Hedges:

With the concept of linguistic variables, we are able to take words as values of (linguistic) variables. In our daily life, we often use more than one word to describe a variable. For example, if we view the speed of a car as a linguistic variable, then its values might be "not slow," "very slow," "slightly fast," "more or less medium,"etc. In general, the value of a linguistic variable is a composite term $x = x_1 x_2 x_3 \ldots \ldots x_n$ that is a concatenation of atomic terms $x_1, x_2, x_3 \ldots \ldots x_n$. These atomic terms may be classified into three groups:

- Primary terms, which are labels of fuzzy sets; in Example 1, they are "slow," "medium," and "fast."
- Complement "not" and connections "and" and "or."
- Hedges, such as "very," slightly," "more or less," etc.

The terms "not," "and," and "or" were studied in sections **2** and **3.** Our task now is to characterize hedges.

Although in its everyday use the hedge very does not have a well-defined meaning, in essence it acts as an intensifier. In this spirit, we have the following definition for the two most commonly used hedges: very and more or less.

Definition 5. Let A be a fuzzy set in U, then very A is defined as a fuzzy set in U with the memb-ership function

$$\mu_{very}A(x) = [\mu_A(x)]^2 \qquad (3.7)$$

and more or less A is a fuzzy set in U with the membership function

$$\mu_{more\ or\ less}\ A(x) = [\mu_A(x)]^{1/2} \qquad (3.8)$$

I .4.3 Fuzzy IF-THEN Rules

I .4.3.1.Introduction:

In section 1 we mentioned that in fuzzy systems and control, human knowledge is represented in terms of fuzzy IF-THEN rules. A fuzzy IF- THEN rule is a conditional statement expressed as

IF < fuzzy proposition >, THEN < fuzzy proposition > (3.9)

Therefore, in order to understand fuzzy IF-THEN rules, we first must know what are fuzzy propositions.

I .4.3.2.Fuzzy propositions:

There are two types of fuzzy propositions: atomic fuzzy propositions, and compound fuzzy propositions. An atomic fuzzy proposition is a single statement

$$x\ is\ A \qquad (3.10)$$

where x is a linguistic variable, and A is a linguistic value of x (that is, A is a fuzzy set defined in the physical domain of x). A compound fuzzy proposition is a composition of atomic fuzzy propositions using the connectives "and," "or," and "not" which represent fuzzy intersection, fuzzy union, and fuzzy complement, respectively. For example, if x represents the speed of the car in Example 4.1, then the following are fuzzy propositions (the first three are atomic fuzzy propositions and the last three are compound fuzzy propositions):

$$x\ is\ S \qquad (4.1)$$

$$x\ is\ M \qquad (4.2)$$

$$x\ is\ F \qquad (4.3)$$

$$x\ is\ S\ or\ x\ is\ not\ M \qquad (4.4)$$

$$x\ is\ not\ S\ and\ x\ is\ not\ F \qquad (4.5)$$

$$(x\ is\ S\ and\ x\ is\ not\ F) or\ x\ is\ M \ (4.6)$$

where S, M and F denote the fuzzy sets "slow," "medium," and "fast," respectively.

Note that in a compound fuzzy proposition, the atomic fuzzy propositions are independent, that is, the x's in the same proposition of (4.4)-(4.5) can be different variables. Actually, the linguistic variables in a compound fuzzy proposition are in general not the same. For example, let x be the speed of a car and $y = \dot{x}$ be the acceleration of the car, then if we define fuzzy set large(L) for the acceleration, the following is a compound fuzzy proposition

$$x\ is\ F\ and\ y\ is\ L$$

Therefore, compound fuzzy propositions should be understood as fuzzy relations. How to determine the membership functions of these fuzzy relations?

- For connective "and" use fuzzy intersections. Specifically, let x and y be linguistic variables in the physical domains U and V, and A and B be fuzzy sets in U and V, respectively, then the compound fuzzy proposition

$$x \text{ is } A \text{ and } y \text{ is } B \qquad (4.7)$$

is interpreted as the fuzzy relation $A \cap B$ $in\ U \times V$ $with\ membership\ function$

$$\mu_{A \cap B}(x, y) = t[\mu_A(x), \mu_B(y)] \qquad (4.8)$$

Where $t : [0,1] \times [0,1] \rightarrow [0,1$ $is\ any\ t - norm$.

- For connective "or" use fuzzy unions. Specifically, the compound fuzzy proposition

$$x \text{ is } A \text{ or } Y \text{ is } B \qquad (4.9)$$

is interpreted as the fuzzy relation $A \cap B$ $in\ U \times V$ with membership function

$$\mu_{A \cup B}(x, y) = s[\mu_A(x), \mu_B(y)] \qquad (4.10)$$

Where $s : [0,1] \times [0,1] \rightarrow [0,1$ $is\ any\ s - norm$.

- For connective "not" use fuzzy complements. That is, replace not$A$ by $\bar{A}$, which is defined according to the complement operators in section 3.

I .4.3.3. Interpretations of Fuzzy IF-THEN Rules

Because the fuzzy propositions are interpreted as fuzzy relations, the key question remaining is how to interpret the IF-THEN operation. In classical propositional calculus, the expression IF p THEN q is written as p → q with the implication → regarded as a connective defined by Table 4.1, where p and q are propositional variables whose values are either truth (T) or false (F). From Table 4.1 we see that if both p and q are true or false, then p → q is true; if p is true and q is false, then p → q is false; and, if p is false and q is true, then p → q is true. Hence, p → q is equivalent to

$$\bar{p} \vee q \qquad (4.11)$$

And

$$(p \wedge q) \vee \bar{p} \qquad (4.12)$$

in the sense that they share the same truth table (Table 4.1) as p → q, where; ∨ and ∧ represent (classical) logic operations "not," "or," and "and," respectively.

Table I.4.1 Interpretations of Fuzzy IF-THEN Rules

p q	p → q
T F T T F T F F	T F T T

Because fuzzy IF-THEN rules can be viewed as replacing the p and q with fuzzy propositions, we can interpret the fuzzy IF-THEN rules by replacing the $^{-}$,∨ and ∧ operators in (4.11) and (4.12) with fuzzy complement, fuzzy union, and fuzzy intersection, respectively. Since there are a wide variety of fuzzy complement, fuzzy union, and fuzzy intersection operators, a number of diff-erent interpretations of fuzzy IF-THEN rules were proposed in the literature. We list some of them below.

In the following, we rewrite (5.7) as IF < FP1 > THEN < FP2 > and replace the p and q in (4.11) and (4.12) by FP1 and FP2, respectively, where FP1 and FP2 are fuzzy propositions. We assume that FP1 is a fuzzy relation defined in $U = U_1 \times U_2 \dots\dots \times U_n,$ FP2 is a fuzzy relation defined in

$V = V_1 \times \dots . \times V_m$,and x and y are linguistic variables (vectors) in U and V, respectively.

- **Dienes-Rescher Implication**: If we replace the logic operators $^{-}$and V in (4.11) by the basic fuzzy complement (3.1) and the basic fuzzy union (3.2), respectively, then we obtain the so-called Dienes-Rescher implication. Specifically, the fuzzy IF-THEN rule IF < FP1 > THEN < FP2 > is interpreted as a fuzzy relation $Q_D$in U × V with the membership function

$$\mu_{Q_D}(x, y) = \max[1 - \mu_{FP1}(x), \mu_{FP2}(y)] \qquad (4.13)$$

- **Zadeh Implication**: Here the fuzzy IF-THEN rule IF < FP1 > THEN < FP2 > is interpreted as a fuzzy relation Q_Z in U x V with the membership function

$$\mu_{Q_Z}(x, y) = \max\left[min\left(\mu_{FP1}(x), \mu_{FP2}\ (y)\right), 1 - \mu_{FP1}(x)\right] \qquad (4.14)$$

Clearly, (5.4) is obtained from (5.2) by using basic fuzzy complement (3.1), basic fuzzy union (3.2), and basic fuzzy intersection (3.3) for $^{-}$,V and A, respectively.

- **Mamdani Implications:** The fuzzy IF-THEN rule IF < FP1 > THEN < FP2 > is interpreted as
a fuzzy relation Q_{MM} or Q_{MP} in U x V with the membership function

$$\mu_{Q_{mm}}(x, y) = \min[\mu_{FP1}(x), \mu_{FP2}(y)] \qquad (4.15)$$

Mamdani implications are the most widely used implications in fuzzy systems and fuzzy control. They are supported by the argument that fuzzy IF-THEN rules are local. However, one may not agree with this argument. For example, one may argue that when we say "IF speed is high, THEN resistance is high," we implicitly indicate that "IF speed is slow, THEN resistance is low." In this sense, fuzzy IF-THEN rules are nonlocal. This kind of debate indicates that when we repress-ent human knowledge in terms of fuzzy IF-THEN rules, different people have different interpret-ations. Consequently, different implications are needed to cope with the diversity of interpret-ations. For example, if the human experts think that their rules are local, then the Mamdani implications should be used; otherwise, the global implications Zadeh,Diences,Lukasiewics should be considered.

- **Takagi sugono implication:** a widely used implication which we will go through in the chapter 2

I .5.Fuzzy Rule Base and Fuzzy Inference Engine

I .5.1.Introduction:

Consider the fuzzy system shown in Fig. I.5, where $U = U_1 \times U_2 \dots \times U_n \subset R^n$, and$V \subset R$. We consider only the multi-input-single-output case, because a multioutput system can always be decomposed into a collection of single-output systems.

For example, if we are asked to design a 4-input-boutput fuzzy system, we can first design three 4-input-1-output fuzzy systems separately and then put them together as in Fig. 10.

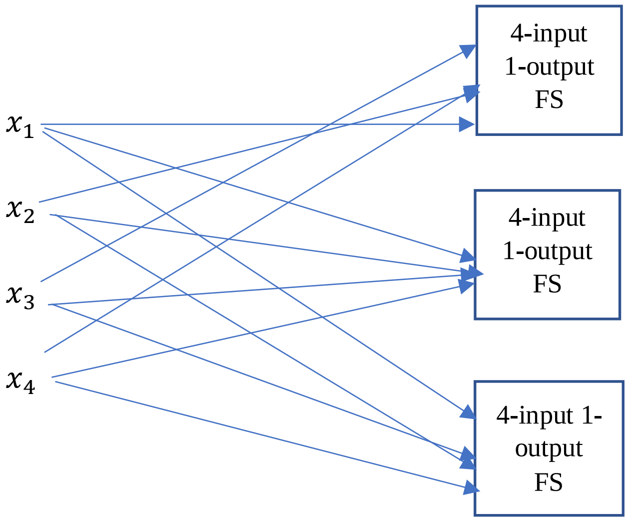

Figure . I.10 A multi-input-multi-output fuzzy system can be decomposed into a collection of multi-input-singleoutput fuzzy systems.

In this section, we will study the details inside the fuzzy rule base and fuzzy inference engine; fuzzifiers and defuzzifiers will be studied in the next section.

I .5.2 Fuzzy Rule Base:

I .5.2.1 Structure of Fuzzy Rule Base:

A fuzzy rule base consists of a set of fuzzy IF-THEN rules. It is the heart of the fuzzy system in the sense that all other components are used to implement these rules in a reasonable and efficient manner. Specifically, the fuzzy rule base comprises the following fuzzy IF-THEN rules:

$$Ru^{(l)}: IF\ x_1\ is\ A_1^l\ and \dots \dots and\ x_n is\ A_n^l\ , THEN\ y\ is\ B^l \qquad (5.1)$$

Where A_1^l and B^l are fuzzy sets in $U_i \subset R\ and\ V \subset R$,respectively, and $x = (x_1, x_2, x_3, \dots, x_n)^T \in U\ and\ y \in V$are inputs and output (linguistic) variables of the fuzzy system, respectively. Let M be the number of rules in the fuzzy rule base; that is, l= 1,2, ..., M in (7.1). We call the rules in the form of (7.1) canonical fuzzy IF-THEN rules because they include many other types of fuzzy rules and fuzzy propositions as special cases.

I .5.2.2 Properties of Set of Rules:

Because the fuzzy rule base consists of a set of rules, the relationship among these rules and the rules as a whole impose interesting questions. For example, do the rules cover all the possible situations that the fuzzy system may face? Are there any conflicts among these rules? To answer these sorts of questions, we introduce the following concepts

Definition 5.1. A set of fuzzy IF-THEN rules is complete if for any $x =\in U$, there exists at least one rule in the fuzzy rule base, say rule $Ru^{(l)}$(in the form of (7.1)), such that

$$\mu_{A_1^l}(x_i) \neq 0 \qquad (5.2)$$

Intuitively, the completeness of a set of rules means that at any point in the input space there is at least one rule that "fires"; that is, the membership value of the IF part of the rule at this point is non-zero.

Definition 5.2. A set of fuzzy IF-THEN rules is consistent if there are no rules with the same IF parts but different THEN parts.

For nonfuzzy production rules, consistence is an important requirement because it is difficult to continue the search if there are conflicting rules. For fuzzy rules, however, consistence is not critical because we will see later that if there are conflicting rules, the fuzzy inference engine and the defuzzifier will automatically average them out to produce a compromised result. Of course, it is better to have a consistent fuzzy rule base in the first place.

Definition 5.3. A set of fuzzy IF-THEN rules is continuous if there do not exist such neighboring rules whose THEN part fuzzy sets have empty intersection.

Intuitively, continuity means that the input-output behavior of the fuzzy system should be smooth. It is difficult to explain this concept in more detail at this point, because we have not yet derived the complete formulas of the fuzzy systems, but it will become clear as we move into section 6.

I .5.3.Fuzzy lnference Engine:

In a fuzzy inference engine, fuzzy logic principles are used to combine the fuzzy IFTHEN

rules in the fuzzy rule base into a mapping from a fuzzy set A' in U to a fuzzy set B' in V. THE fuzzy IF-THEN rule is interpreted as a fuzzy relation in the input-output product space U x V, and there is a number implications that specify the fuzzy relation. Because any practical fuzzy rule base constitutes more than one rule, the key question here is how to infer with a set of rules. There are two ways to infer with a set of rules: composition based inference and individual-rule based inference, which we will discuss next.

I .5.3.1 Composition Based lnference:

In composition based inference, all rules in the fuzzy rule base are combined into a single fuzzy relation in U x V, which is then viewed as a single fuzzy IF-THEN rule.

So the key question is how to perform this combination. We should first understand what a set of rules mean intuitively, and then we can use appropriate logic operators to combine them.

There are two opposite arguments for what a set of rules should mean. The first one views the rules as independent conditional statements. If we accept this point of view, then a reasonable operator for combining the rules is union. The second one views the rules as strongly coupled conditional statements such that the conditions of all the rules must be satisfied in order for the whole set of rules to have an impact. If we adapt this view, then we should use the operator intersection to combine the rules.

I .5.3.2 Individual-Rule Based lnference:

In individual-rule based inference, each rule in the fuzzy rule base determines an output fuzzy set and the output of the whole fuzzy inference engine is the combination of the M individual fuzzy sets. The combination can be taken either by union or by intersection.

I .5.3.3 The Details of Some lnference Engines:

From the previous two subsections we see that there are a variety of choices in the fuzzy inference engine. Specifically, we have the following alternatives: (i) composition based inference or individual-rule based inference, and within the composition based inference, Mamdani inference or Godel inference, (ii) Dienes-Rescher implication Lukasiewicz, implication , Zadeh implication , Godel implication , or Mamdani implica-tions , and (iii) different operations for the t-norms and s-norms in the various formulas. So a natural question is: how do we select from these alternatives?

In general, the following three criteria should be considered:

- **Intuitive appeal:** The choice should make sense from an intuitive point of view. For example, if a set of rules are given by a human expert who believes that these rules are independent of each other, then they should be combined by union.
- **a Computational eficiency:** The choice should result in a formula relating B' with A', which is simple to compute.
- **Special properties:** Some choice may result in an inference engine that has special properties. If these properties are desirable, then we should make this choice.

We now show the detailed formulas of a two of fuzzy inference engines that are commonly used in fuzzy systems and fuzzy control.

- **Product Inference Engine:** In product inference engine, we use: (i) individualrule based inference with union combination , (ii) Mamdani's product implication , and (iii)algebraic product for all the t-norm operators and max for all the s-norm operators.We obtain the product inference engine as

$$\mu_{B'}(y) = \max_{i=1}^{M}[\sup_{X \in U} (\mu_{A'}(x), \prod_{i=1}^{n} , \mu_{A_1^l}(x_i), \mu_{B'}(y))] \qquad (5.3)$$

That is, given fuzzy set A' in U, the product inference engine gives the fuzzy set B' in V according to (5.3).

- Minimum Inference Engine: In minimum inference engine, we use: (i) individual-rule based inference with union combination, (ii) Mamdani's minimum implication and (iii) min for all the t-norm operators and max for all the s-norm operators.

$$\mu_{B'}(y) = max_{i=1}^{M} [\sup_{X\in U} \min(\mu_{A'}(x), \mu_{A_1^l}(x_1), \ldots\ldots, \mu_{A_n^l}(x_n), \mu_{B'}(y))] \qquad (5.4)$$

That is, given fuzzy set A' in U, the minimum inference engine gives the fuzzy set B' in V according to (5.4).

I .6.Fuzzifiers and Defuzzifiers:

I .6.1.Introduction:

We learned from section 5 that the fuzzy inference engine combines the rules in the fuzzy rule base into a mapping from fuzzy set A' in U to fuzzy set B' in V. Because in most applications the input and output of the fuzzy system are realvalued numbers, we must construct interfaces between the fuzzy inference engine and the environment. The interfaces are the fuzzifier and defuzzifier in Fig. 1.5.

I .6.2.Fuzzifiers:

The fuzzifier is defined as a mapping from a real-valued point $x^* \in U \subset R^n$ to a fuzzy set A' in U. What are the criteria in designing the fuzzifier? First, the fuzzifier should consider the fact that the input is at the crisp point x*, that is, the fuzzy set A' should have large membe-rship value at x*. Second, if the input to the fuzzy system is corrupted by noise, then it is desireable that the fuzzifier should help to suppress the noise. Third, the fuzzifier should help to simplify the computations involved in the fuzzy inference engine From (5.3), (5.4 we see that the most complicated computation in the fuzzy inference engine is the $sup_{x\in U}$, therefore our objective is to simplify the computations involving $sup_{x\in U}$.
We now propose three fuzzifiers:

- **Singleton fuzzifier:** The singleton fuzzifier maps a real-valued point x* ∈ U into a fuzzy singleton A' in U, which has membership value 1 at x* and 0 at all other points in U; that is,

$$\mu_{A'}(x) = \begin{cases} 1 & if\ x = x^* \\ 0 & otherwise \end{cases} \qquad (6.1)$$

- **Gaussian fuzzifier**: The Gaussian fuzzifier maps x* ∈ U into fuzzy set A' in U, which has the following Gaussian membership function:

$$\mu_{A'}(x) = e^{-(\frac{x_1 - x_1^*}{a_1})^2} * \ldots \ldots * e^{-\left(\frac{(x_n - x_n^*)}{a_n}\right)^2} \qquad (6.2)$$

where a_i are positive parameters and the t-norm * is usually-chosen algebraic product or min.

- **Triangular fuzzifier:** The triangular fuzzifier maps x* ∈ U into fuzzy set A' in U, which has the following triangular membership function

$$\mu_{A'}(x) = \begin{cases} \left(1-\frac{|x_1-x_1^*|}{b_1}\right) * \dots\dots * \left(1-\frac{|x_n-x_n^*|}{b_n}\right) if\ |x_i-x_i^*| \le b_i, i=1,2,\dots\dots,n \\ 0\ otherwise \end{cases} \quad (6.3)$$

where $b_i$are positive parameters and the t-norm * is usually chosen as algebraic product or min.

I .6.3.Defuzzifiers:

The defuzzifier is defined as a mapping from fuzzy set B' in V ⊂R (which is the output of the fuzzy inference engine) to crisp point y* ∈ V. Conceptually, the task of the defuzzifier is to specify a point in V that best represents the fuzzy set B'. This is similar to the mean value of a random variable. However, since the B' is constructed in some special ways (see section 5), we have a number of choices in determining this representing point. The following three criteria should be considered in choosing a defuzzification scheme:

- **Plausibility:** The point y* should represent B' from an intuitive point of view; for exa-mple, it may lie approximately in the middle of the support of B' or has a high degree of membership in B'.
- **Computational simplicity:** This criterion is particularly important for fuzzy control because fuzzy controllers operate in real-time.
- **Continuity:** A small change in B' should not result in a large change in y*.

We now propose three types of defuzzifiers. For all the defuzzifiers, we assume that the fuzzy set B' is obtained from one of the five fuzzy inference engines in Chapter 7, that is, B' is given by (5.3),or(5.4). From these equations we see that B' is the union or inter-section of M individual fuzzy sets.

I .6.3.1 Center of gravity Defuzzifier:

The center of gravity defuzzifier specifies the y" as the center of the area covered by the membership function of B', that is,

$$y^* = \frac{\int_V y\mu_{B'}(y)dy}{\int_V \mu_{B'}(y)dy} \quad (6.4)$$

where $\int_V$ is the conventional integral. Fig. 11 shows this operation graphically.

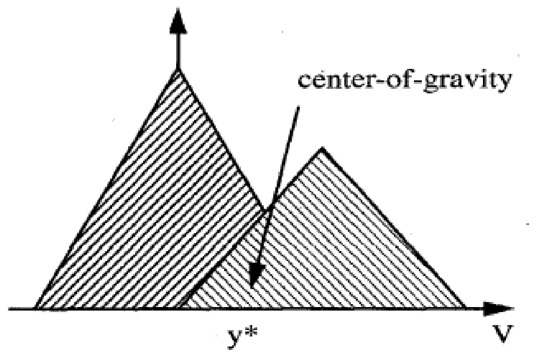

Figure I. .11. A graphical representation of the center of gravity defuzzifier.

If we view $\mu_{B'}(y)$ as the probability density function of a random variable, then the center of gravity defuzzifier gives the mean value of the random variable.Sometimes it is desirable to eliminate the y E V, whose membership values in B' are too small; this results in the indexed center of gravity defuzzifier, which gives

$$y^* = \frac{\int_{V_a} y\mu_{B'}(y)dy}{\int_{V_a} \mu_{B'}(y)dy} \qquad (6.5)$$

where V_a is defined as

$$V_a = \{y \in V | \mu_{B'}(y) \geq a\} \qquad (6.6)$$

and a is a constant.

The advantage of the center of gravity defuzzifier lies in its intuitive plausibility.The dis-advantage is that it is computationaily intensive. In fact, the membership function $\mu_{B'}(y)$ is usually irregular and therefore the integrations in (6.5) and(6.4) are difficult to compute. The next defuzzifier tries to overcome this disadvantage by approximating (6.4) with a simpler formula.

I .6.3.2 Center Average Defuzzifier:

Because the fuzzy set B' is the union or intersection of M fuzzy sets, a good approximation of (6.4) is the weighted average of the centers of the M fuzzy sets, with the weights equal the heights of the corresponding fuzzy sets. Specifically, let y^{-l} be the center of the l'th fuzzy set and w_l be its height, the center average defuzzifier determines y^*as

$$y^* = \frac{\sum_{l=1}^{M} y^{-l} w_l}{\sum_{l=1}^{M} w_l} \qquad (6.7)$$

Fig. 12 illustrates this operation graphically for a simple example with M = 2.

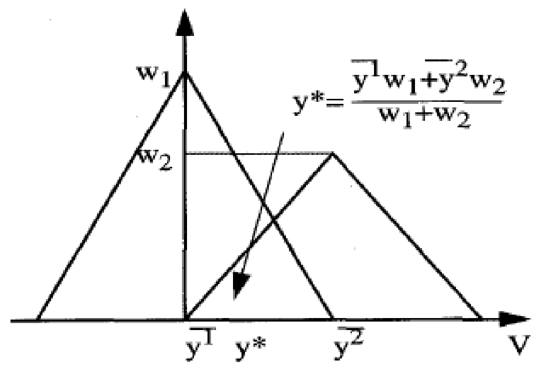

Figure I. .12. A graphical representation of the center average defuzzifier

The center average defuzzifier is the most commonly used defuzzifier in fuzzy systems and fuzzy control. It is computationally simple and intuitively plausible.Also, small changes in $y^{-l}$and w_l result in small changes in y^*

I .6.3.3 Maximum Defuzzifier

Conceptually, the maximum defuzzifier chooses the y^* as the point in V at which $\mu_{B'}(y)$ achieves its maximum value. Define the set

$$hgt(B') = \left\{ y \in V \middle| \mu_{B'}(y) = \sup_{y \in V} \mu_{B'}(y) \right\} \tag{6.8}$$

that is, $hgt(B')$ is the set of all points in V at which $\mu_{B'}(y)$ achieves its maximum value. The maximum defuzzifier defines y^*as an arbitrary element in $hgt(B')$, that is,

$$y^* = any\ point\ in\ hgt(B') \tag{6.9}$$

If $hgt(B')$ contains a single point, then y* is uniquely defined. If $hgt(B')$ contains more than one point, then we may still use (6.9) or use the smallest of maxima, largest of maxima, or mean of maxima defuzzifiers. Specifically, the smallest of maxima defuzzifier gives

$$y^* = inf\{y \in hgt(B')\} \tag{6.10}$$

the largest of maxima defizzifier gives

$$y^* = sup\{y \in hgt(B')\} \tag{6.11}$$

and the mean of maxima defuzzifier is defined as

$$y^* = \frac{\int_{hgt(B')} y dy}{\int_{hgt(B')} dy} \tag{6.12}$$

where $\int_{hgt(B')}$ is the usual integration for the continuous part of $hgt(B')$ and is summation for the discrete part of $hgt(B')$.We feel that the mean of maxima defuzzifier may give results which are contradictory to the intuition of maximum membership. For example, the y* from the mean of maxima defuzzifier may have very small membership value in B'; see Fig. 6.3 for an example. This problem is due to the nonconvex nature of the membership function $\mu_{B'}(y)$.

The maximum defuzzifiers are intuitively plausible and computationally simple. But small changes in B' may result in large changes in y*; see Fig.6.4 for an example. If the situation in Fig.I.6.4 is unlikely to happen, then the maximum defuzzifiers are good choices.

I .6.3.4. Comparison of the Defuzzifiers:

Table I.6.1 compares the three types of defuzzifiers according to the three criteria:plausibility, computational simplicity, and continuity. From Table I.8.2 we see that the center average defuzzifier is the best.

Finally, we consider an example for the computation of the defuzzifiers with some particular membership functions.

Table I.6.1Comparison of the center of gravity, center average, and maximum defuzzifiers with respect to plausibility, computational simplicity, and continuity.

	Center of gravity	Center average	maximum
Plausibility	Yes	Yes	Yes
Computational simplicity	No	Yes	Yes
continuity	Yes	Yes	Yo

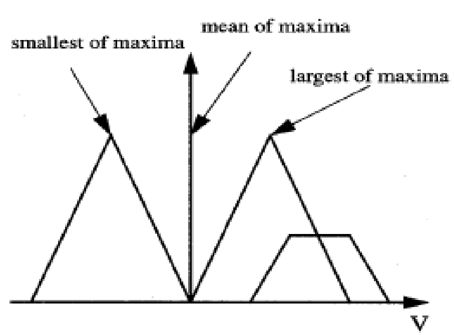

Figure .I.13. A graphical representation of the maximum defuzzifiers. In this example, the mean of maxima defuzzifier gives a result that is contradictory to the maximummembership intuition

I .7.Summary and further readings:

In this chapter we have demonstrated the following:

- The goal of using fuzzy systems is to put human knowledge into engineering systems in a systematic, efficient, and analyzable order.
- The fuzzy IF-THEN rules used in certain industrial processes and consumer products.
- The definitions of fuzzy set, basic concepts associated with a fuzzy set (support,a-cut, convexity, etc.) and basic operations (complement, union, intersection,etc.) of fuzzy sets.
- The intuitive meaning of membership functions and how to determine intuitively appealing membership functions for specific fuzzy descriptions.
- The concept of linguistic variables and the characterization of hedges.
- The concept of fuzzy propositions and fuzzy IF-THEN rules.
- Different interpretations of fuzzy IF-THEN rules, including Dienes-Rescher,Lukasiewicz, Zadeh, Godel and Mamdani implications.
- The structure of the canonical fuzzy IF-THEN rules and the criteria for evaluating a set of rules.
- The computational procedures for the composition based and individual-rule based inferences.
- The detailed formulas of some fuzzy inference engines: product, minimum, ….
- The definitions and intuitive meanings of the singleton, Gaussian and triangular fuzzifiers, and the center of gravity, center average and maximum defuzzifiers.
- Computing the outputs of the fuzzy systems for different combinations of the fuzzifiers, defuzzifiers, and fuzzy inference engines for specific example

Chapter II

Fuzzy modeling

II .1.Introduction:

The concepts of fuzzy-set theory and fuzzy logic can be employed in the modeling of systems in a number of ways. Examples of fuzzy systems are rule-based fuzzy systems (Zadeh, 1973; Driankov, et aI., 1993), fuzzy linear regression models (Tanaka, et aI., 1982), or fuzzy models using cell structures (Smith, et aI., 1994). This chapter focuses only on rule-based fuzzy systems, i.e., systems where the relationships between variables are represented by a means of fuzzy if-then rules of the form:

If antecedent proposition **then** consequent proposition.

Depending on the particular structure of the consequent proposition, three types of models are distinguished:

- Linguistic fuzzy model (Zadeh, 1973; Mamdani, 1977), where both the antecedent and consequent are fuzzy propositions.
- Fuzzy relational model (Pedrycz, 1984; Yi and Chung, 1993), which can be regarded as a generalization of the linguistic model, allowing one particular antecedent proposition to be associated with several different consequent propositions via a fuzzy relation.
- Takagi-Sugeno (TS) fuzzy model (Takagi and Sugeno, 1985), where the consequent is a crisp function of the antecedent variables rather than a fuzzy proposition.

II.2. Fuzzy Clustering:

II.2.1.Cluster analysis

Cluster analysis is the classification of objects according to the existing similarities between them,and organizing the data into groups. Clustering techniques can be applied to quantitative (numerical) data, qualitative (categorical) data or a mixture of them.

II.2.1.1 classic partition and fuzzy partition

- **classic partition** :

The goal of clustering is to partition the Z data set into clusters. Using classical sets, a classical partition of Z can be defined as a family of subsets$\left\{\frac{A_i}{1} \leq i \leq c\right\} \complement P(Z) with$:

$$\bigcup_{i=1}^{c} A_i = Z$$

$$A_i \cap A_j = \emptyset \quad 1 \leq i \neq j \leq c \qquad (6.9)$$

$$\emptyset \complement A_i \complement Z \quad 1 \leq i \leq c$$

In terms of membership function, equation (1) can be rewritten as:

$$\bigvee_{i=1}^{c} \mu_{A_i} = 1$$

$$\mu_{A_i} \cap \mu_{A_j} = 0 \quad 1 \leq \mathrm{i} \neq \mathrm{j} \leq \mathrm{c} \tag{6.10}$$

$$0 < \mu_{A_i} < 1 \quad 1 \leq i \leq c$$

here, 0 and 1 represent the functions zero and one, respectively and μ_{A_i} the characteristic function of A_j.

The matrix U=[μ_{ik}]of the ordre c x N represents The classical partition and only if these Elements satisfy the following conditions:

$$\mu_{ik} \in \{0,1\} \quad 1 \leq i \leq c, 1 \leq k \leq N$$

$$\sum_{i=1}^{c} \mu_{ik} = 1 \quad 1 \leq k \leq N \tag{6.11}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N \quad 1 \leq i \leq c$$

The ith line of the matrix U contains the values of the membership function of the ith subset A_i of the set Z.

- fuzzy partition:

The generalization of the classical score to the fuzzy score is obtained directly by allowing μ_{ik} to belong to the real interval [0,1]. Hence the following conditions for such a partition:

$$\mu_{ik} \in [0,1] \quad 1 \leq i \leq c, 1 \leq k \leq N$$

$$\sum_{i=1}^{c} \mu_{ik} = 1 \quad 1 \leq k \leq N \tag{6.12}$$

$$0 < \sum_{k=1}^{N} \mu_{ik} < N \quad 1 \leq i \leq c$$

II.2.2. Fuzzy classification algorithms

II.2.2.1. Clustering algorithm of the type C-means (FCM)

The algorithm (FCM) was resulted from the work of Dunn [Dunn, 1974] then improved by Bezdek [Bezdek, 1981]] .

Most fuzzy clustering algorithms are based on the optimization of the objective function C-means given by:

$$J_{FCM}(Z; U, V) = \sum_{i=1}^{c} \sum_{k=1}^{N} (\mu_{ik})^m \|Z_k - V_i\|_A^2 \tag{6.13}$$

Où

$$\mathrm{U} = [\mu_{ik}] \tag{6.14}$$

is the fuzzy partition matrix of Z

$$V = [v1;v2;\ldots\ldots\ldots.. ;vc],\ v_i \in R^n \qquad (6.15)$$

is the vector of the prototypes (centers) of the clusters, which must be determined, 1 $<i<c$.

$$D_{ika}^2 = \|Z_k - V_i\|_A^2 = (Z_k - V_i)^T A (Z_k - V_i) \qquad (6.17)$$

is a distance standard, and

$$m \in]1, \infty]$$

is a weighting exponential that determines the level of fuzziness of the resulting clusters.

Therefore, the dissimilarity measure will be the square of a distance between each element k z and the center of the cluster v_i. This distance will be weighted by the power of the degree to which this element belongs $(\mu_{ik})^m$:

The minimization of the objective function (6) gives us:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} (\frac{D_{ika}}{D_{jka}})^{\frac{2}{m-1}}} 1 \leq i \leq c, \quad 1 \leq k \leq N \qquad (6.18)$$

$$V_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m z_k}{\sum_{k=1}^{N} (\mu_{ik})^m} 1 \leq i \leq c \qquad (6.19)$$

The principale of the algorithm is given by::

Given the data set Z, choose a number of class1 <c <N, the exponent m>1, the stop tolerance ε> 0 and the standard matrix A. Randomly initialize the matrix of partition U :

Repeat for l = 0,1,2,……..

Step 1 :Calculate the centre of the clusters (means) :

$$V_i^l = \frac{\sum_{k=1}^{N} (\mu_{ik}^{(l)})^m z_k}{\sum_{k=1}^{N} (\mu_{ik}^{(l)})^m} \qquad 1 \leq i \leq c$$

Step 2 :calculate the distances :

$$D_{ikA_i}^2 = \left(Z_k - V_i^{(l)}\right)^T A \left(Z_k - V_i^{(l)}\right) \qquad 1 \leq i \leq c, 1 \leq k \leq N$$

Step 3 :update the matrix of partition : If $D_{ikA_i} > 0\ for\ 1 \leq i \leq c,\ 1 \leq k \leq N$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (\frac{D_{ikA}}{D_{jkA}})^{\frac{2}{m-1}}}$$

Otherwise

$$\mu_{ik}^{(l)} = 0\ si\ D_{ikA_i} < 0\ et\ \mu_{ik}^{(l)} \in [0,1] avec \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$$

Until $\left\|U^{(l)} - U^{(l-1)}\right\| < \varepsilon$

The shape of the clusters is determined by the choice of matrix A in the equation. The particular choice *A* = *I*, induces the standard Euclidean norm:

$$D_{ikA}^2 = \left(Z_k - V_i^{(l)}\right)^T \left(Z_k - V_i^{(l)}\right) \qquad (6.20)$$

in this case the clusters detected are in spherical shapes.

MATLAB application :

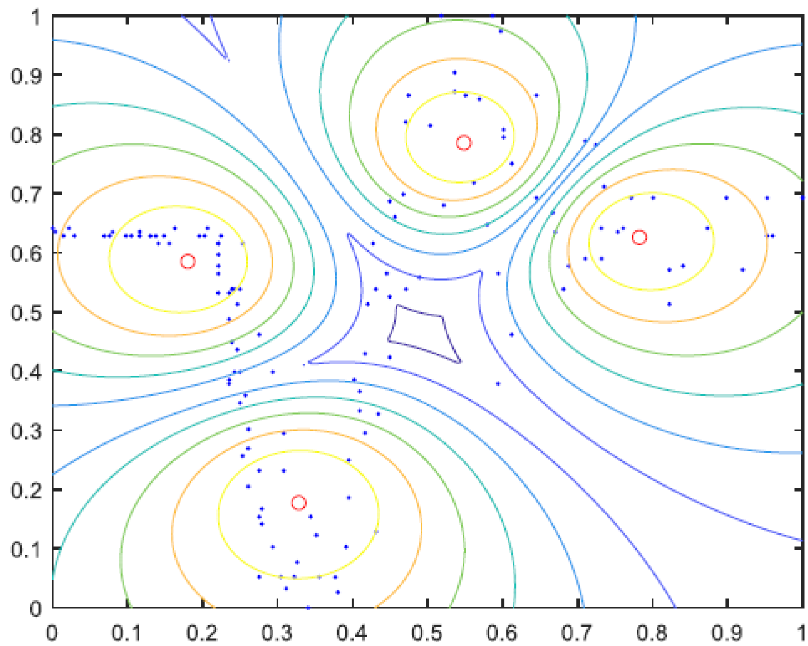

Figure .II.1.The clusters and its centers

II.2.2.2.Algorithm of Gustafson-Kessel (GK)

In 1979, Gustafson and Kessel [1978] generalized the FCM algorithm by employing an adaptive distance standard in order to detect clusters of different geometric shapes in a dataset. In this case, each class has its own standard matrix, which results in:

$$D_{ika}^2 = \|Z_k - V_i\|_A^2 = (Z_k - V_i)^T A_i (Z_k - V_i) \qquad (6.21)$$

We assume that the matrix A_i satisfies the hypothesis:

$$|A_i| = \rho_i, \rho_i > 1 \qquad (6.22)$$

Where ρ_i is fixed for each cluster.In this case, the optimization of (6) gives us the following expression for A_i:

$$A_i = [\rho_i det(F_i)]^{\frac{1}{n}} F_i^{-1} \qquad (6.23)$$

where F_i is the fuzzy covariance matrix of the i-th class given by :

$$F_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^{N}(\mu_{ik})^m} \tag{6.24}$$

which leads us to the following Gastafson-Kessel (GK) algorithm:

Given the data set Z, choose a number of class1 <c <N, the exponent m>1, the stop tolerance ε> 0 and the standard matrix A. Randomly initialize the matrix of partition U :

Repeat for l = 0,1,2,……..

Step 1 :calculate the centers of the clusters :

$$v_i^l = \frac{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m} \qquad 1 \le i \le c$$

Step 2 : Calculate clusters covariance matrices:

$$F_i = \frac{\sum_{k=1}^{N}(\mu_{ik}^{l-1})^m \left(z_k - v_i^l\right)\left(z_k - v_i^l\right)^T}{\sum_{k=1}^{N}(\mu_{ik})^m} \qquad 1 \le i \le c$$

Step 3 :Calculate the distances :

$$D_{ikA_i}^2 = \left(Z_k - V_i^{(l)}\right)^T \left[\left(\rho_i det(F_i)\right)^{\frac{1}{n}} {F_i}^{-1}\right]\left(Z_k - V_i^{(l)}\right) \qquad 1 \le i \le c, 1 \le k \le N$$

Step 4 :Update the partition matrix :

If $D_{ikA_i} > 0\ for\ 1 \le i \le c,\ 1 \le k \le N$

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c}\left(\frac{D_{ikA_i}}{D_{ikA_i}}\right)^{\frac{2}{m-1}}}$$

otherwise

$$\mu_{ik}^{(l)} = 0\ si\ D_{ikA_i} < 0\ et\ \mu_{ik}^{(l)} \in [0,1] avec \sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$$

Until $\left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$

MATLAB application

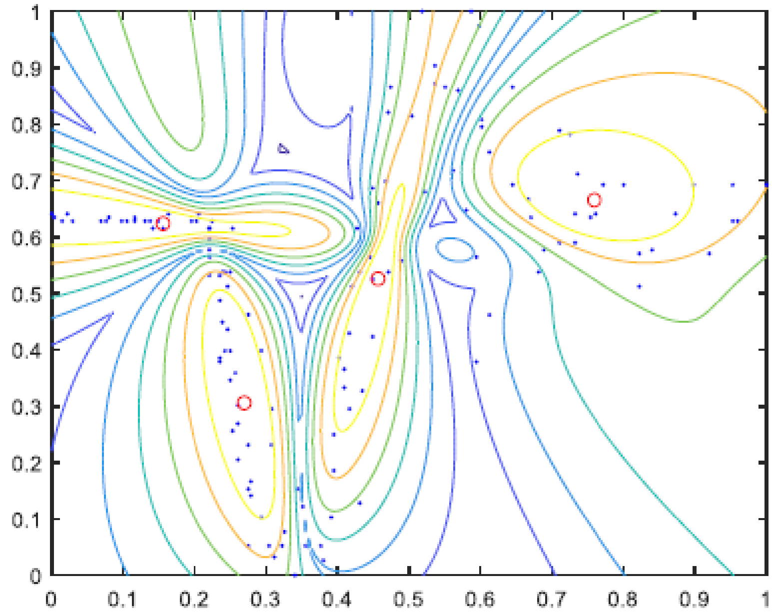

Figure. II.2. The clusters and its centers

The advantage of the GK algorithm over the FCM algorithm is its ability to detect clusters with different shapes and orientations in a single set of data as shown in the figure 16

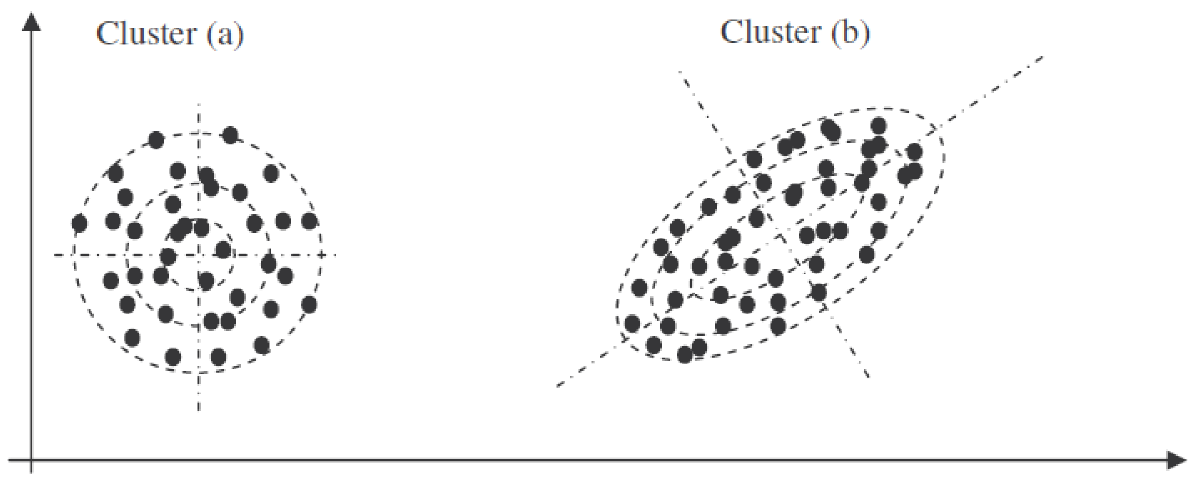

Figure .II..3.(b) Elliptical cluster detectable only by the GK algorithm

(a) Spherical cluster detectable by the FCM and GK algorithms

II.3. Fuzzy modeling of the Takagi-sugono type:

II.3.1. Introduction:

Almost all of the physical dynamical systems in real life cannot be represented by linear diffr-ential equations and have a nonlinear nature. At the same time, linear control methods rely on the key assumption of small range of operation for the linear model, acquired from linearizing the nonlinear system, to be valid. When the required operation range is large, a linear control-ler is prone to be unstable, because the nonlinearities in the plant cannot be properly dealt with.

One way to cope with such difficulty is to develop a nonlinear model (MIMO)composing of a number of sub-models which are simple(linear) or (MISO), understandable, and responsible for respective subdo-mains. The idea of multi-model approach is not new, but the idea of fuzzy modeling using the concept of the fuzzy sets theory offers a new technique to build multi-models of the process based on the input-output data or the original mathematical model of the system.

The fuzzy model proposed by Takagi and Sugeno is described by fuzzy IF-THEN rules which represents local input-output relations of a nonlinear system. And it is proved that Takagi-Sugeno fuzzy models are universal approximators of any smooth nonlinear system.

II.3.2. Product space clustering for identification

This section addresses the decomposition of a nonlinear identification problem into a set of locally linear models by means of product-space fuzzy clustering. The identification procedure is first outlined in Section 2.2.1. Structure selection and the choice of regressors in the modeling of dynamic systems are discussed in Section 2.2.2. Section 2.2.3 describes the principle of identification of nonlinear systems by product-space clustering. The choice of clustering algorithms is discussed in Section 2.2.4. Section 2.2.5 deals with the determination of the number of clusters by means of validity measures and compatible cluster merging.

II.3.2.1 Outline of the Approach

Figure .II.4. outlines the individual steps of the identification procedure, which is iterative in its nature. In a typical modeling session, some of the steps may be repeated for different choices of the various parameters. The purpose and the different steps and the related methods are outlined below.

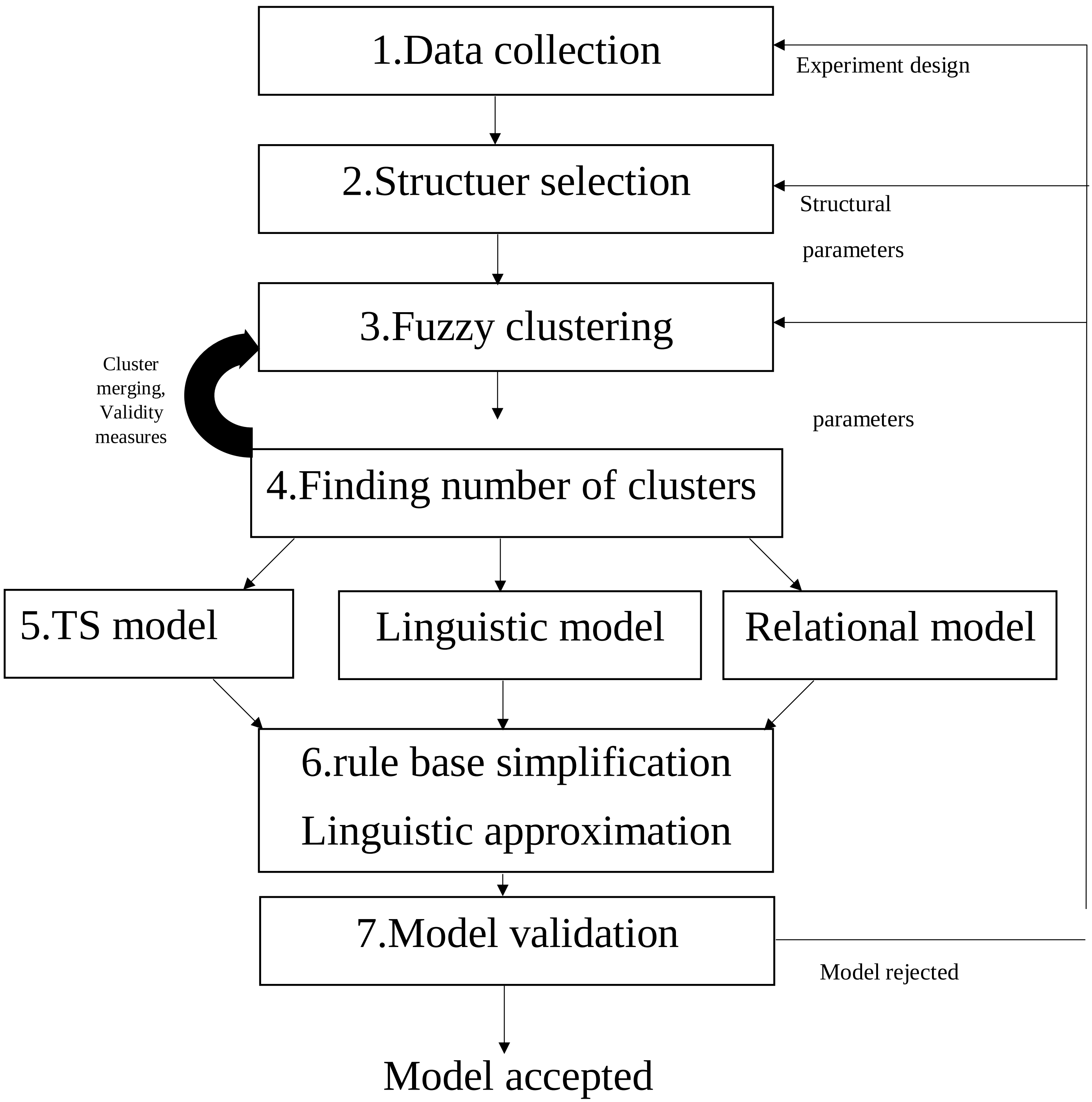
1.Data collection
Experiment design
2.Structuer selection
Structural
parameters
3.Fuzzy clustering
Cluster
merging,
Validity
measures
parameters
4.Finding number of clusters
5.TS model
Linguistic model
Relational model
6.rule base simplification
Linguistic approximation
7.Model validation
Model rejected
Model accepted

Figure . II.4...Overview of the identification approeach based on fuzzy clustering

Step 1: Design of id[…] an important initial step for any ident[…]ion content of the identification data set. As opposed to linear techniques,pseudo-random binary excitation signals are not suitable for nonlinear identification in general, and for fuzzy clustering in particular. Although the choice of the excitation signal may be problem dependent, the input data should preferably excite the system in the entire range of the considered variables both in amplitude and in frequency. The pseudo-random binary signal is not suitable, since it only contains two amplitude levels. Typical choices are a multi-sinusoidal signal or a step-wise signal with random amplitude and random width (Godfrey, 1993). White noise of small amplitude is often added to these signals in order to guarantee proper excitation of the process dynamics. The choice of a suitable sampling period, the design of (anti-aliasing) filters, the duration of the experiments, etc., are other important issues of the experiment design.

Step 2: Structure selection. The purpose of this step is to determine the relevant input and output variables with respect to the aim of the modeling exercise. When identifying dynamic systems, the structure and the order of the model dynamics must be chosen. Structure selection allows us to translate the identification of a dynamic system into a regression problem that can be solved in a static or quasi-static manner. The structure can be selected in an automated way by comparing different candidate structures in terms of some performance measures. In most cases, a reasonable choice can be made by the user, based on the prior knowledge about the process.

Step 3: Clustering of the data. Structure selection leads to a nonlinear static regression problem, which is then approximated by a collection of local linear submodels.The location and the parameters of the submodels are found by partitioning the available data into hyperplanar or hypereUipsoidal clusters. Each of the clusters defines a fuzzy region in which the system can be approximated locally by a linear submodel.

Step 4: Selection of the number of clusters. By applying cluster validity measures, compatible cluster merging, or a combination of the two techniques, an appropriate number of clusters can be found. This step typically involves several repetitions of Step 3 for a different number of clusters and a different initial partition matrix.

Step 5: Generation of an initial fuzzy model. Fuzzy clustering divides the available data into groups in which local linear relations exists between the inputs and the output. In order to obtain a model suitable for prediction or controller design, a rule-based fuzzy model of a selected structure is derived from the available fuzzy partition matrix and from the cluster prototypes. The rules, the membership functions and other parameters that constitute the fuzzy model are extracted in an automated way. The exact procedure applied at this step depends on the type of fuzzy model required and on the purpose of modeling (prediction, analysis, control design, etc.).

Step 6: Simplification and reduction of the initial model. Initial fuzzy models obtained from data may be redundant in the sense that they contain more membership functions than are necessary to describe the system. Fuzzy similarity measures can be applied to simplify or reduce the initial fuzzy rule base and to obtain linguistic interpretation of the membership functions. Chapter 5 presents the corresponding techniques.

Step 7: Model validation. By means of validation, the final model is either accepted as appropriate for the given purpose, or it is rejected. In the latter case, some steps of the identification loop shown in Figure.II.4. may be repeated with a different setting, as it is usual also in other approaches to linear and nonlinear system identification (Ljung, 1987; Johansen, 1994). In addition to the usual numerical validation by means of simulation, interpretation of fuzzy models plays an important role in the validation step. The coverage of the input space by the rules can be analyzed, and, for an incomplete rule base, additional rules can be provided based on prior knowledge, local linearization, or first-principle models.

II.3.2.2 Structure Selection :

In fuzzy modeling, the problem of structure selection! can be divided into three subproblems: 1) choice of input and output variables; 2) representation of the system's dynamics, and 3) choice of the fuzzy model's granularity.

Choice of input and output variables. Although most identification methods assume that the input and output variables of the process are known (Ljung, 1987), in reality, especially for multivariable and closed-loop systems, it is often not clear which variables should be considered as the model inputs. The selection of the input and output variables is based on the aim of the modeling exercise, on the prior knowledge related to the (expected) process dynamics, and on additional variables that may cause the nonlinearity of the system. Statistical techniques, such as correlation analysis, can be used in combination with prior knowledge. This step can also be partially automated. Several candidate models with different input variables can be compared in terms of some performance measure, and the best one is then selected.

Representation of the system's dynamics. A common approach is to transform the identification of a dynamic system into a static regression problem (Leonaritis and Billings, 1985; Chen and Billings, 1989; Sjoberg, et aI., 1995). The choice of this particular transformation is usually based on a combination of a priori knowledge with intuition, insights, and understanding of the process behavior. Mechanistic (physical, first-principle) modeling of the well-understood relationships and physical laws can guide the selection of the relevant variables, and of the model's order. This transformation can be regarded as a mapping from the domain of time signals into a space of variables that fully determine the state of the system. These variables are called the regressors. The system's behavior can be predicted by means of a static mapping from the space of regressors to the space of the model output (regressand).

A major distinction can be made between input-output models, state-space models, and hybrid (semi-mechanistic) models. The choice of the regressors is a crucial step, as an inappropriate choice may hamper the modeling effort. Choosing too poor a structure (too few regressors) results in inaccurate modeling of the process dynamics and nonlinearities. Choosing a structure richer than necessary (too many regressors) leads to badly conditioned estimation problems and to "overfitting" the data.

II.3.2.2.1 The Nonlinear Regression Problem

This problem consists of transforming the identification of a dynamic system into a static regression problem(NARX model for example). This transformation can be seen as an application of a domain of time signals to the space of state signals which are called regressors. The behavior of the system can then be predicted using a static transformation from the regr-essor space to the model output space. In this chapter we will use the fuzzy logic to approxim-ate this transformation.

A nonlinear regression is a modeling of a static relationship between an output variable $y \in Y \subset \mathbb{R}$ called regressand and a regression vector $x = \left[x_1, x_2, \ldots\ldots, x_p\right]^T$ in a domain where $X \subset \mathbb{R}^p$. The elements of the regression vector are called regressors and the domain X is called the regression space. The system generating the data will be described by:

$$y = f(x) \qquad (6.25)$$

The goal of regression is to use the data to construct a function F (x) that can serve as a reasonable approximation of $f(x)$, not just for the available data, But also on all the elements of domain X. The prediction error will then be characterized in a continuous domain by:

$$I = \int_x \|f(x) - F(x)\|\, dx \qquad (6.26)$$

or in a discrete domain by:

$$J = \frac{1}{N}\sum_{i=1}^{N} \|f(x_i) - F(x_i)\| \qquad (6.27)$$

where N denotes the number of data samples. The minimum of I or J will give the best model for the selected structure.

II.3.2.2.2. Input-output Black-box Models':

As in the case of linear systems, there are several possibilities to choose the regressors in a black box identification. The NARX model (Nonlinear AutoRegressive with eXogenous input) is frequently used in the identification of nonlinear systems. This model establishes a relationship between the past input-output values of the data and the prediction of the output :

$$\hat{y}(k+1) = F(y(k), \ldots.., y\big(k - n_y + 1\big), u(k), \ldots\ldots u(k - n_u + 1)) \qquad (6.28)$$

where k denotes the discrete time sample, and n_u, n_y, are integers related to the order of the system.

In NARX models, the regression vector is a collection of the past inputs and outputs, $x(k) = [y(k), y(k-1, \dots\dots, y(k-n_y+1), u(k), u(k-1), \dots\dots]^T$.The regression is the predicted output $\check{y}(k+1)$, therefore, from an input / output measurement set of an unknown dynamic system.

$$S = \{(u(j), y(j))/j \; = \; 1, \dots\dots, N\} \quad (6.29)$$

The function F (.) in (20) can be approximated using nonlinear static regression. A rule of a NARX model can be represented by the following form:

$$R_i: IF \quad y(k) \; is \; A_{i1} \; \; and \dots\dots and \quad y(k-n_y+1) is \; A_{in_y} \quad (6.30)$$

$$and \; u(k) is \; B_{i1} and \dots\dots. and \; \; u(k-n_u+1) \; is \; B_{in_u} \; \; THEN \; \; y(k+1) \; \; is \; C_i$$

The quadratic prediction error in this case will then be given by:

$$J = \frac{1}{N}\sum_{i=1}^{N}(y(i) - \hat{y}(i))^2 \quad (6.31)$$

II.3.2.2.3 State-space Framework:

In addition to the most frequently used input-output structures, fuzzy models can also represent nonlinear systems in the state-space form:

$$x(k+1) = g(x(k), u(k))$$
$$y(k) = h(x(k))$$

where state transition function 9 maps the current state $x(k)$ and the input $u(k)$ into a new state $x(k+1)$. The output function h maps the state $x(k)$ into the output $y(k)$. An example of a rule-based representation of a state-space model is the following Takagi-Sugeno model:

$$if \; x(k) \; is \; A_i \; and \; u(k) \; is \; B_i \; then \begin{cases} x(k+1) = A_i x(k) + \; B_i u(k) \\ y(k) = \; C_i x(k) \end{cases} \quad (6.32)$$

for i = 1, ... ,K. Here A_i, B_i, C_i are matrices of appropriate dimensions, associated

with the ith rule.

II.3.2.2.4 Semi-mechanistic Modeling:

With physical insights in the system, nonlinear transformations of the measured signals can be involved. When modeling, for instance, the relation between the room temperature and the voltage applied to an electric heater, the power signal is computed by squaring the voltage, since it is the heater power rather than the voltage that causes the temperature to change (Lindskog and Ljung, 1994). This new variable is then used in a linear black-box model instead of the voltage itself. The motivation for using nonlinear regressors in nonlinear models is not to waste effort (rules, parameters, etc.)

on estimating facts that are already known.Another approach is based on a combination of white-box and black-box models.In many systems, such as chemical and biochemical processes, the modeling task can be divided into two subtasks: modeling of well-understood mechanisms based on mass and energy balances (first-principle modeling), and approximation of partially known relationships such as specific reaction rates. A number of hybrid modeling approaches have been proposed that combine first principles with nonlinear black-box models, e.g., neural networks (Psichogios and Ungar, 1992; Thompson and Kramer, 1994) or fuzzy models (Babuska, et al., 1996). A neural network or a fuzzy model is typically used as a general nonlinear function approximator that "learns" the unknown relationships from data and serves as a predictor of unmeasured process quantities that are difficult to model from first principles.

II.3.2.3 Identification by Product-space Clustering:

The principle of identification by product-space clustering is to approximate a nonlinear regression problem by decomposing it into several local linear subproblems. This approach has a number of advantages in comparison with global nonlinear models, such as neural networks. The model structure is easy to understand and interpret, both qualitatively and quantitatively. Various types of knowledge can be integrated in the model, including empirical knowledge, measured data and available mathematical models. In addition, the approach has computational advantages and lends itself to straightforward adaptive and learning algorithms (Murray-Smith and Johansen, 1997). Fuzzy clustering is applied in the product space of the regressors and the regressand: X x Y. Let X denote the matrix in, having the regression vectors x_k^Tin its rows, and let y denote the column vector in $\mathbb{R}^{N\times p}$, containing the regressands y_k:

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \qquad (6.33)$$

N denotes the number of data samples, p is the dimension of the regression vector. For an input-output model of a dynamic system, the matrix X contains shifted versions of the input and output data. As an example, assume a second-order NARX model

$$y(k+1) = F(y(k), y(k-1), u(k), u(k-1)).$$

With the set of available measurements, $S = \{(u(j), y(j))/j \;=\; 1, \dots\dots, N_d\}$, the regressor matrix and the regressand vector are:

$$X = \begin{bmatrix} y(2) & y(1) & u(2) & u(1) \\ y(3) & y(2) & u(3) & u(2) \\ \vdots & \vdots & \vdots & \vdots \\ y(N_d-1) & y(N_d-2) & y(N_d-1) & y(N_d-2) \end{bmatrix}, \qquad y = \begin{bmatrix} y(3) \\ y(4) \\ \vdots \\ y(N_d) \end{bmatrix}$$

In this example,$N = N_d - 2$. The decomposition of a global nonlinear mapping into a set of locally linear models is based on a geometrical interpretation of the regression problem. The unknown nonlinear function $y = f(x)$represents a nonlinear (hyper)surface in the product space: $(X \times Y) \subset R^{p+1}$• This surface is called the regression surface.

II.3.2.4.Choice of Clustering Algorithms:

Two of the Clustering algorithms suitable for locating clusters have been reviewed in Sections 1.2.1 (c-means) and 1.2.2 (GK-means). All the Clustering algorithms that are suitable for locating clusters that are linear subspaces of the data are based on extensions of the c-means function-al, which is a least-squares criterion for minimizing the variance of the data from the cluster means. These algorithms differ in the definition of the distance measure and of the prototypical structure for the clusters. Because of these differences, each algorithm performs in a different way for the same data set. With regard to the choice of a suitable clustering algorithm for system identification, it is required that:

- The clusters represent local linear models of the function being approximated.This facilitates the analysis and control design based on the obtained model.
- The projection of the partition matrix onto the regressors results in a semantically interpretable partition with unimodal fuzzy sets.
- The algorithm is robust with respect to the initialization, and does not suffer from convergence to local optima representing unsatisfactory solutions of the approximation problem.
- The algorithm is able to reveal clusters of different sizes, since some regions in the regression space can be easily represented as a single linear model, while other regions may require finer partitioning.

The advantages and drawbacks of some clustering algorithms are discussed in the following sections.

II.3.2.4.1.Clustering with Adaptive Distance Measure:

Two algorithms are considered in this section, the Gustafson-Kessel (GK) algorithm and the fuzzy maximum likelihood estimates (FMLE) algorithm.

Gustafson-Kessel Algorithm. The GK algorithm appears to be a suitable method for identific-ation purposes, because of the following properties:

- The size of the clusters is limited by the definition of the distance measure (6.21).
- In comparison with the other considered algorithms, the GK algorithm is relatively insensitive to the initialization of the partition matrix (or cluster prototypes).
- As the GK algorithm is based on an adaptive distance measure, it is not so sensitive to scaling (normalization, standardization) of the data
- The GK algorithm can detect clusters of different shapes, not only linear subspaces.

The GK algorithm has, however, also some drawbacks:

- The calculation of the inverse and of the determinant of the covariance matrix in each iteration slows down the algorithm considerably for a large data dimension n and a large number of clusters c.

- When only a small number of data samples is available, or when the data are linearly dependent, numerical problems occur when the covariance matrix becomes close to singular.
- Without any prior knowledge, the volumes, Pi, of the clusters are set equal to each other. The GK algorithm then cannot detect clusters that differ largely in their volumes.

Fuzzy Maximum Likelihood Clustering. As suggested by Gath and Geva (1989), the FMLE cl-ustering algorithm should be able to automatically detect clusters of varying volumes, con-trary to the GK algorithm. A drawback of the FMLE algorithm is that it generates almost cris-p partitions due to the exponential distance measure(6.22), and, consequently, it is also more se-nsitive to the initial conditions. It is thus useful to generate the initial partition, for instance, by the GK algorithm, and then initialize the FMLE algorithm with that partition.

$$D_{ik\sum_i} = \frac{[det \sum_i]^{1/2}}{P_i} exp\left[\frac{1}{2}(Z_k - V_i)^T \textstyle\sum_i^{-1} \ (Z_k - V_i)\right] \qquad (6.22)$$

II.3.2.4.2.Fuzzy c-lines and c-elliptotypes:

The fuzzy c-lines (FCL) and fuzzy c-elliptotypes (FCE) algorithms are designed to detect linear clusters, see Section 3.5.2. The cluster prototypes are defined as linear varieties (lines in $\mathbb{R}^2$,

Planes in $\mathbb{R}^3$, and hyperplanes in a general multidimensionalspace) and the distance metric measures the distance of data points from the linear varieties.

II.3.2.4.3.Fuzzy c-regression Models:

The fuzzy c-regression (FCR) algorithm yields simultaneous estimates for the parameters of the local regression models together with the partitioning of the data.

Contrary to the previous methods, with this approach, the cluster prototypes are not geom-etrical objects in the data space, but are defined explicitly by functional relationships in terms of regression equations.

II.3.2.5.Determining the Number of Clusters:

Before fuzzy clustering can be applied, the number of clusters must be specified. Two met-hods to determine the number of clusters are considered in this section: cluster validity me-asures, and compatible cluster merging. Validity measures assess the goodness of the obtai-ned partition by using criteria like the within-cluster distance, the partition density, the ent-ropy, etc. Cluster merging approaches start with a higher number of clusters than are expe-cted for the particular problem. The initial number of clusters is then reduced by successive-ely merging compatible clusters until some threshold is reached and no more clusters can be merged.

II.3.2.5.1.Cluster Validity Measures:

The use of cluster validity measures is a standard approach to determining an appropriate number of clusters in a data set. Clustering algorithms generally aim at locating well-separ-ated and compact clusters. When the number of clusters is chosen equal to the number of groups that actually exist in the data, it can be expected that the clustering algorithm will identify them correctly. When this is not the case, misclassifications appear, and the clusters are not likely to be well separated and compact. Hence, most cluster validity measures are designed to quantify the separation and the compactness of the clusters.

II.3.2.5.2.Compatible Cluster Merging:

A compatible cluster merging (CCM) algorithm was proposed by Krishnapuram and Freg (1992) for finding an appropriate number oflinear or planar clusters in 2D or 3D image data. The algorithm starts with C = Cmax , which is greater than the maximum number of clusters expected for the particular problem. The number of clusters is then reduced by successively merging compatible clusters until some threshold is reached and no more clusters can be merged. Figure.II.5 depicts the principal idea of compatible cluster merging.

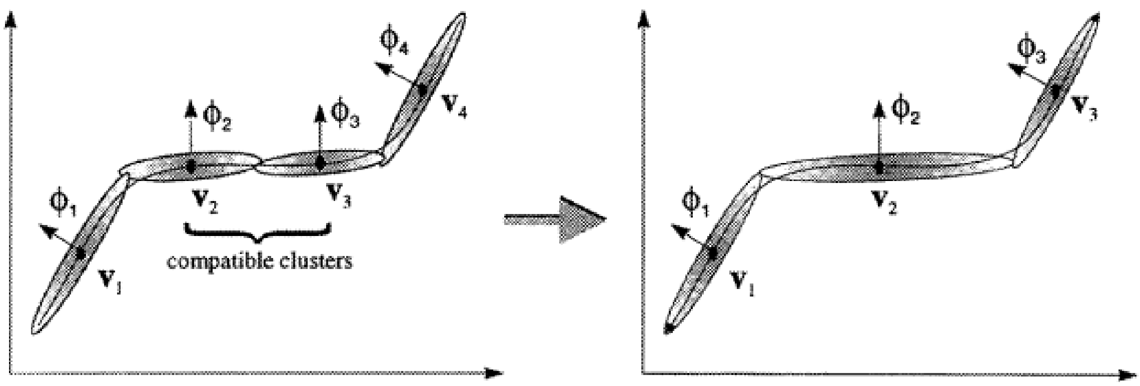

Figure . II.5. Merging of compatible clusters.

II.3.2.6. Takagi-sugono modeling:

To approximate a non linear problam we need a concatenation of local linear models, in this segment we will see how to obtain these models. In this case, each sub-model will be represe-nted by a fuzzy relation. Such a model can be represented either as a rule base or as a fuzzy relation. In this chapter, we propose to use fuzzy models of the Takagi-Sugeno (TS) type to perf-orm fuzzy identification. For this, each cluster obtained in the Cartesian product can be consi-dered as a local linear approximation of the regression surface. The global model can be suita-bly represented by the use of the affine rules of Takagi-Sugeno (TS):

$$R_i: IF\ x\ is\ A_i\ THEN\ y_i = a_i^T x + b_i, \qquad i = 1,2, \dots\dots, K$$

The defuzzification will be obtained using the following general formula:

$$y = \frac{\sum_{i=1}^{K} \omega_i(x) y_i}{\sum_{i=1}^{K} \omega_i(x)}$$

The membership functions defined in the rows of the partition matrix are multidimensional membership functions.

In the projection method, we project the fuzzy partition matrix on the axes of the premise variablesx_j, $1 \le j \le p$. The rules of TS then become:

$$R_i: IF\ x_1 is\ A_{i1} and\ \dots\dots x_p\ is\ A_{ip}\ THEN\ y_i = a_i^T x + b_i, \qquad i = 1,2, \dots\dots, K$$

The degree of activation of a rule is calculated by the product of the individual memberships:

$$\omega_i(x) = \prod_{j=1}^{p} \mu_{A_{ij}}(x)$$

Where $\mu_{A_{ij}}(x)$ is the membership function of the fuzzy set A_{ij} .

To obtain the premiseof the membership functions A_{ij}, 1 ≤ j ≤ p,l, the multidimensional fuzzy set discretely defined in the partition matrix must be projected onto the regressors x_j:

$$\mu_{A_{ij}}(x_{jk}) = proj_j(\mu_{ik})$$

Note that the resulting membership functions are discrete in nature, and are defined for identification data only.

Fig.II.6. shows: (a) Data to be identified (b) Cluster centers detected (c) Clusters detected (d) The small lines in bold represent the smallest eigenvectors of the covariance matrices, large thin lines represent linear models premises given by the largest eigenvectors

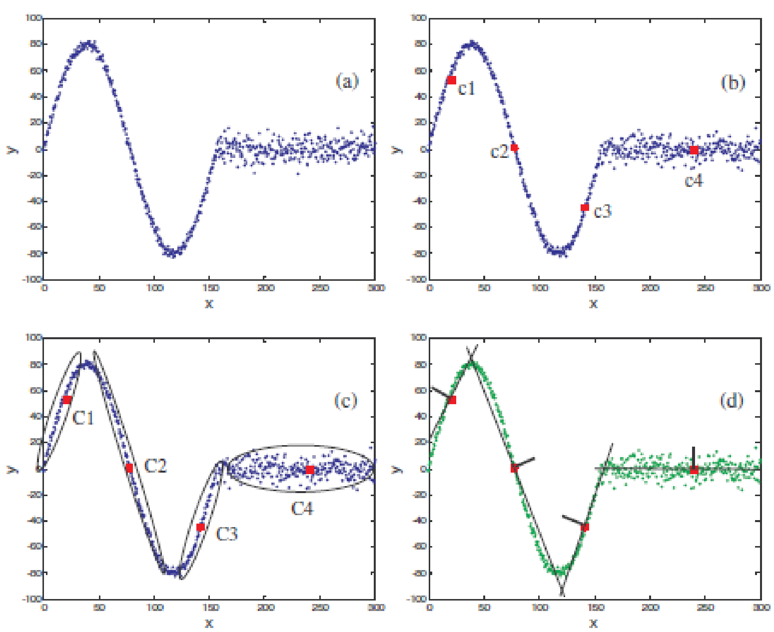

Figure . II.6

II.3.2.7.Rule Base Simplification Algorithm.

The simplification algorithm merges similar fuzzy sets iteratively using two thresholds: $\eta \in (0,1)$for merging fuzzy sets that are similar to one another, and $\eta_r \in (0,1)$ for removing fuzzy sets similar to the universal set. In each iteration, the similarity between all fuzzy sets for each antecedent variable is considered. The pair of membership functions with the highest similarity $s \geq \eta$ are merged. The rule base is updated by substituting the new fuzzy set for the ones merged. The algorithm repeatedly evaluates the similarities in the updated rule base, until there are no more fuzzy sets for which $\geq \eta$. Finally, fuzzy sets similar to the universal set are removed from the antecedents of the rules in which they occur. The algorithm only merges one pair of fuzzy sets per iteration. Merging two fuzzy sets $A_{lq} and\ A_{mq}$ is accomplished by taking the support of the new fuzzy set A as the support of $A_{lq} \cup A_{mq}$. This guarantees preservation of the coverage of the antecedent space. The kernel of A is given by averaging the kernels of $A_{lq} and\ A_{mq}$, making a trade-off between the two rules l and m. The procedure is summarized in the Algorithm below.

Given a fuzzy rule base obtained by clustering, select the thresholds $\eta_r, \eta \in (0,1)$.

Repeat

Step 1: Select the two most similar fuzzy sets in the rule base. Calculate $S_{ijk} = (A_{ij}, A_{kj})$,$j = 1,2 \ldots\ldots, p, i, k = 1,2, \ldots., K$. Select $A_{lq} and\ A_{mq}$,such that $s_{lmq} = \max_{i,j,k,i\neq k}\{s_{ijk}\}$.

Step 2 : Merge the two most similar fuzzy sets and update the rule base.

$If\ S(A_{lq}, A_{mq}) \geq \eta\ merge\ A_{lq} and\ A_{mq}$ to create a new fuzzy set A and replace

$$A_{lq} = A\ and\ A_{mq} = A$$

Until :no more fuzzy sets have similarity $s_{ijk} \geq \eta$.

Step 3: Remove fuzzy sets similar to the universal set. For each fuzzy set A_{ij} calculate $S(A_{ij}, U)$,

Where $\mu_U = 1, \forall_{x_j}$. If $S(A_{ij}, U) \geq \eta_r$ remove $A_{ij}\ from\ the\ antecedent\ of\ the\ ith\ rule$.

II.3.2.8.Model Validation.

The validation of fuzzy models has several facets, namely the standard validation through numerical simulations and comparisons with the process data, analysis of the linear consequent models (stability, gains, time constants, nonminimum phase behavior, etc.) and analysis of the co-verage of the input space by the rules. For an incomplete rule base, additional rules can be ob-tained from prior knowledge, or by local linearization of first-principle models. The antecedents of these rules can be created from unused combinations of membership functions in the initial model.

The identification data usually cover only a fraction of the entire product space of the model variables. Therefore, the antecedents of the obtained rules include only those combinations of the linguistic terms which were identified from the data. However, in simulation or prediction, it is possible that regions not covered by any rules are entered. This situation can be detected by observing the degrees of fulfillment in the rule base. If no rule is activated above a specified threshold, an additional rule may be added to the rule base. The antecedent of this rule is given by the combination of linguistic terms which give the highest degree of fulfillment for the given data point, see 139 for an example. This is a major difference from many other purely black-box techniques, such as neural networks, CMAC, splines, etc., where the validation relies entirely on numerical simulation.

II.4.Summary and further readings:

In this chapter we have learned the following:

- By means of product-space fuzzy clustering, a data set generated by a nonlinear system can be partitioned into fuzzy subsets of data that are locally described by linear submodels.
- Prior to clustering, the regression structure of the model must be selected,in order to properly represent the system's dynamics.
- Problems where little prior knowledge is available are usually represented in an input-output form, using the NARX structure.
- When the structure of the system is (partly) known and when the state variables can be measured or reconstructed from other variables, state-space or hybrid semi-mechanistic modeling approaches should be preferred to black-box input-output models.

Chapitre 3:

Fuzzy modeling of SISO and MIMO systems via C-means and GK-means methods of clustering

III.1.Introduction.

In this chapter we're going to simulate two different systems (SISO,MIMO),Both with two different ways of clustering (C-means,GK-means),and compare the two different results of each system individually.

III.2.Application of SISO system.

III.2.1.description.

In this example, we develop a simple dynamic model for the relationship between the throttle angle and the speed of an engine.

III.2.2.simulation.

The input Data (Valve position) and the output data (Pressure) figures:

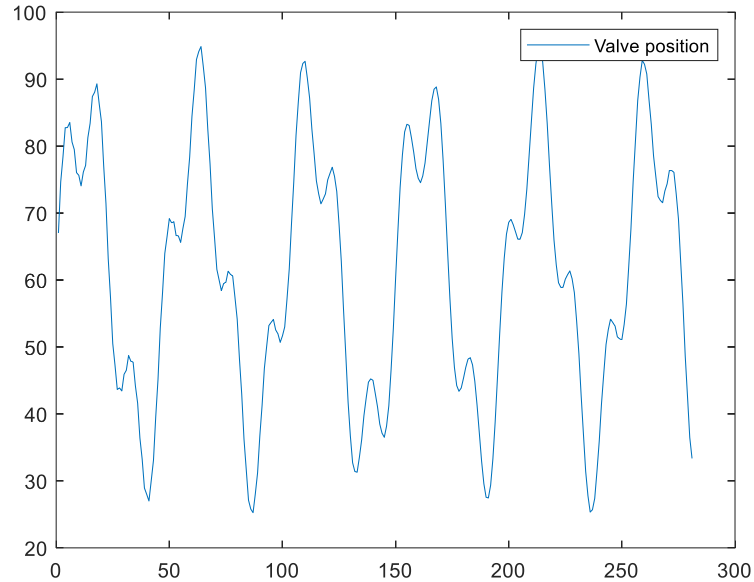

Figure .III.1.Input data:Valve position

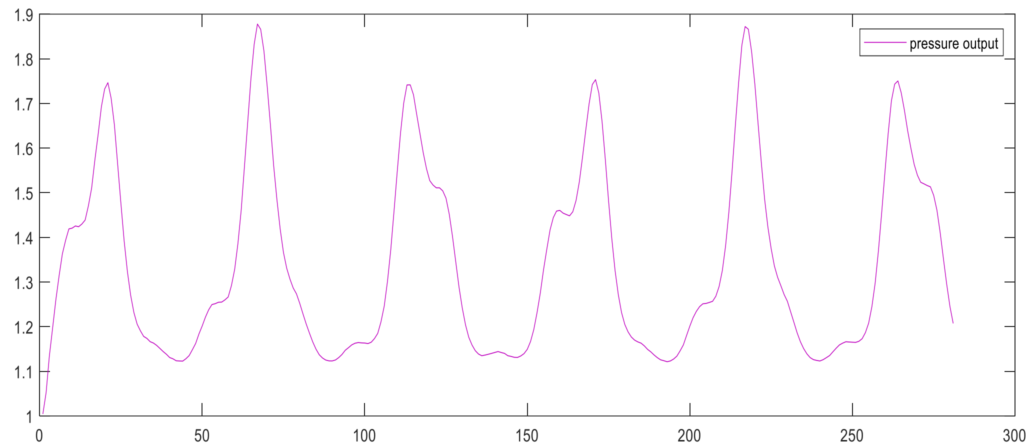

Figure . III.2. Output data (pressure)

The simulation results of the system fuzzy modeling:

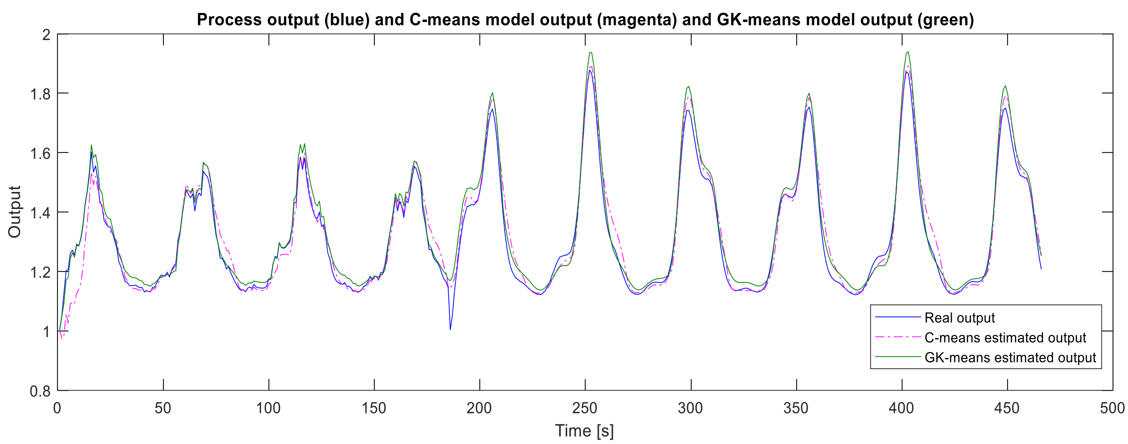

Figure .III.3. the real output and the estimated outputs

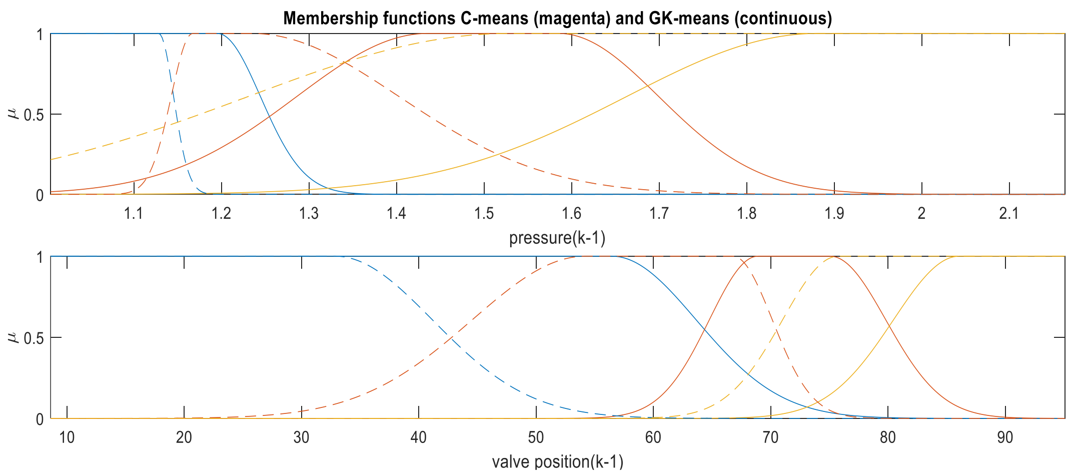

Figure . III.4. Membership functions of siso system with C-means and GK-means

III.2.3. Results comparison between C-means and GK-means.

The following table presents the comparison of some valuable results of this SISO system fuzzy modeling

	VAF	RMS
C-means	94.9535	0.0430
GK-means	98.1362	0.0336

III.3.Application of MIMO system.

III.3.1.description.

Consider a MIMO process consisting of four cascaded tanks as shown in Fig. III.5.. The inputs are the two flow rates $u = [Q_1, Q_2]^T$, and the outputs are the four levels $y = [h_1, h_2, h_3, h_4]^T$.

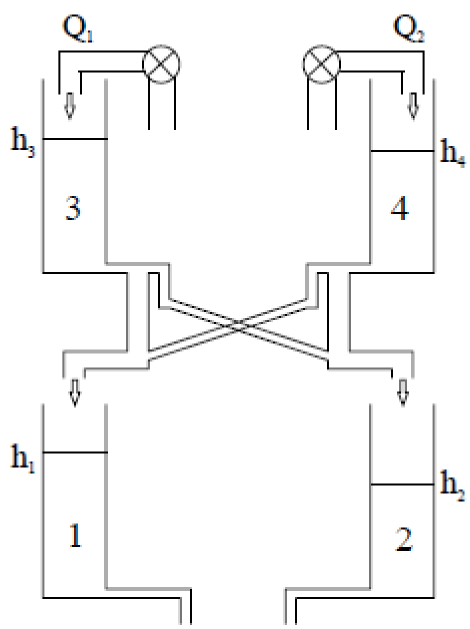

Figure . III.5. Four cascaded tanks.

The measured outputs are the levels in the four tanks.These signals are given in Fig. III.7. The number of samples available for identication is 1000 and the sample time is 10 s.

III.3.2.simulation.

The input Data (Water flow rate) and the output data (The warer levels of the four tanks) figures:

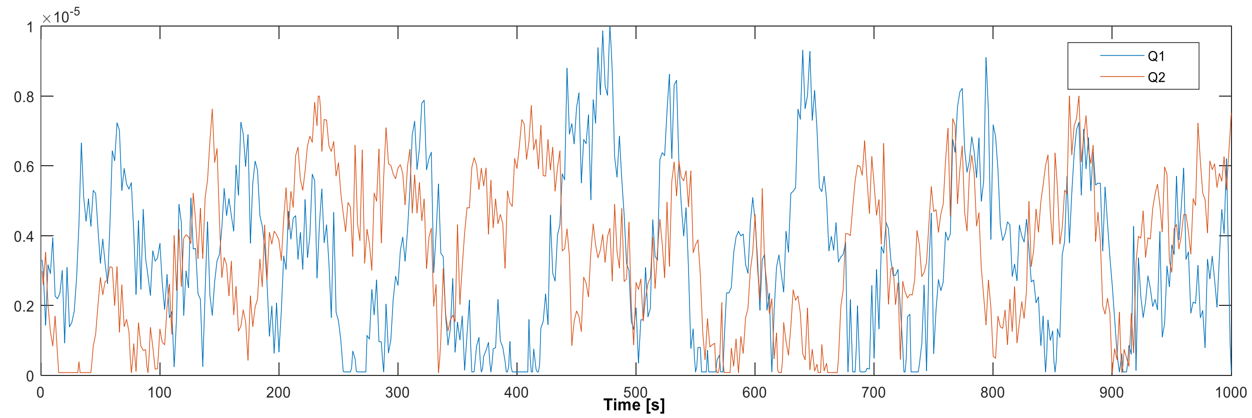

Figure . III.6. Input data (Water flow rate)

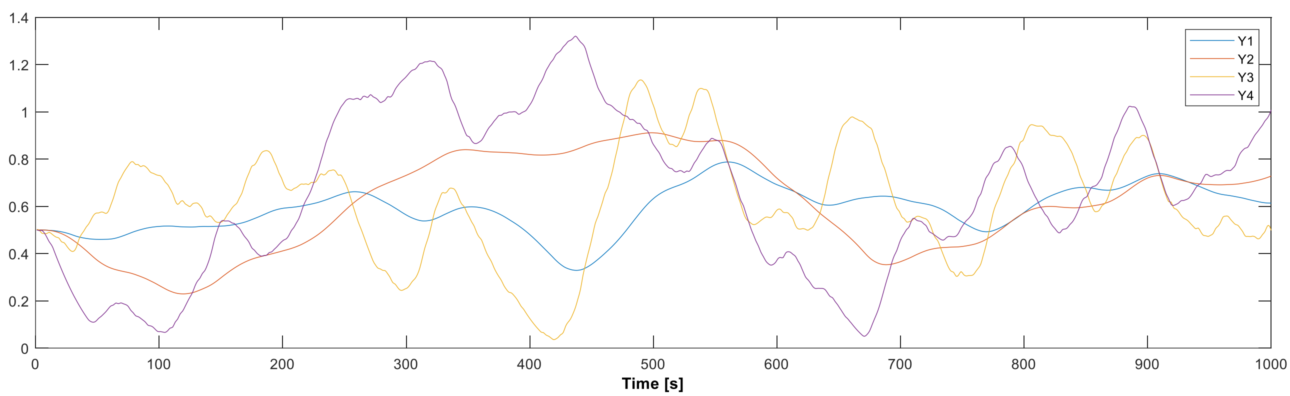

Figure . III.7. Ouput data (Tanks water levels)

The simulation results of the system fuzzy modeling:

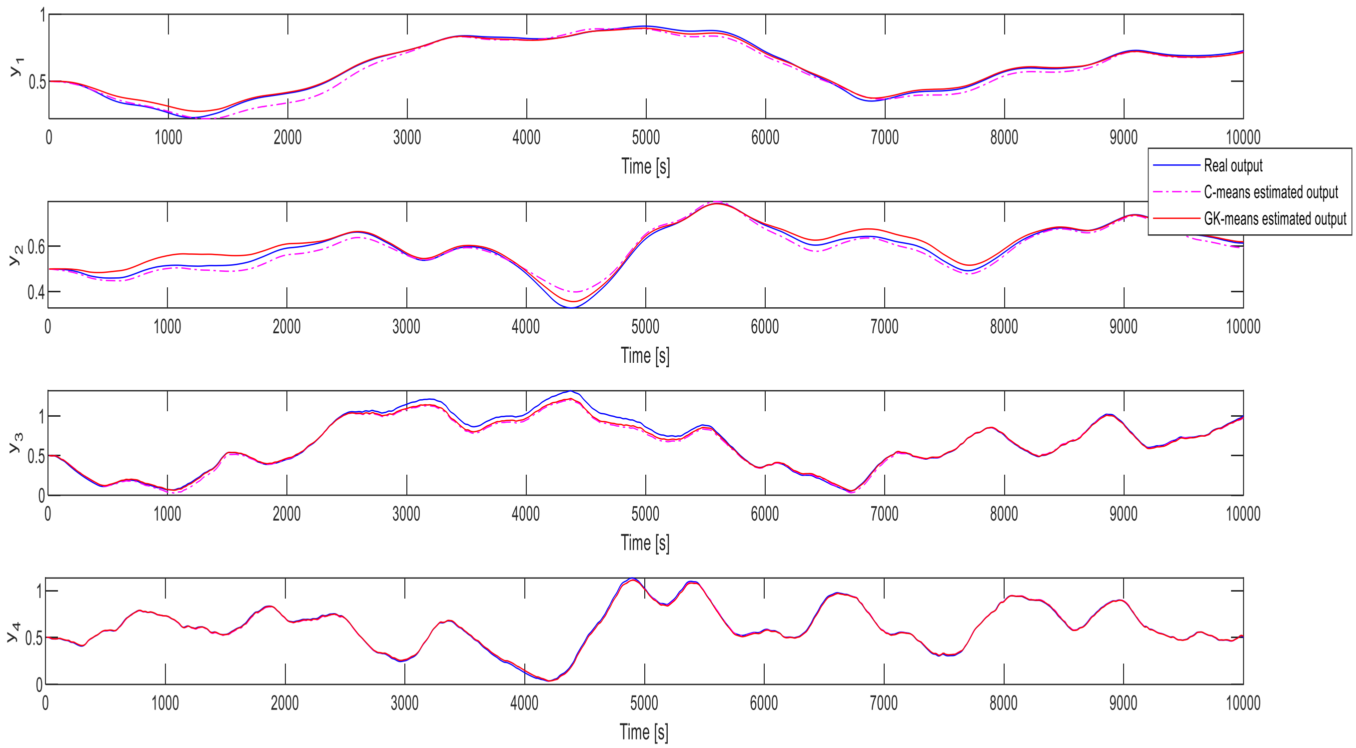

Figure . III.8. Real and estimated outputs

III.3.3. Results comparison between C-means and GK-means.

The following table presents the comparison of some valuable results of this MIMO system fuzzy modeling

	VAF				RMS			
C-means	98.5965	95.6373	98.6678	99.8172	0.0307	0.0215	0.0483	0.0099
GK-means	99.0253	97.7123	99.0167	99.7082	0.0216	0.0214	0.0374	0.0115

III.4.Conclusion.

The comparison of the results of the fuzzy modeling with the two clustering methods individually shows that the GK-means method provides a slightly better precise estimation (a higher percentile variance accounted for (VAF) between two signals) and (a less Root-mean-squared (RMS) error between two signals)

General conclusion.

In this memoire we went through the fundamentals of the fuzzy modeling including the mathematics and the base concepts of the fuzzy sets theory and fuzzy logic, the detailed approach of the fuzzy modeling of the takagi sugono type including (data selection, structure selection ,fuzzy clustering, choice of the number of clusters and the clustering algorithm, rule base simplification Linguistic approximation, modal validation…), then we explained The different ways of data clustering and in the end by analyzing some examples we learned how the usage of the GK-means clustering can provide a better and more precise results compared to the less complicated method of clustering C-means

Bibliographical references

[1] Li-Xin Wang, “A Course in Fuzzy Systems and Control”,China, 1997.
[2] Prof. Dr. Dr. h.c. Hans-Jiirgen Zimmermann, Editor European Laboratory for Intelligent Techniques Engineering Aachen, Gennany, December 2016.
[3] Tanaka, K., and M. Sugeno [1992], "Stability analysis and design of fuzzy control systems," Fuzzy Sets and Systems, 45, no. 2, pp. 135-156
[4] Sugeno, M., and M. Nishida [1985], "Fuzzy control of model car," Fuzzy Sets and Systems, China, pp. 103-113.
[5] Shafer, G. [1976], A Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ.
[6] Krishnapuram, R (1994). Generation of membership functions via possibilistic clustering. In IEEE World Congress on ComputationalIntelligence, Volume 2, Orlando,U.S.A., pp. 902-908.
[7] Spooner, J.T., and K.M. Passino [1995], "Stable adaptive control using fuzzy systems and neural networks," IVHS OSU Report 95-01.
[8] Vandegrift, M.W., F.L. Lewis, S. Jagannathan, and K. Liu [1995], "Adaptive fuzzy control of discrete-time dynamical systems," personal communication.
[9] Wang, L. X. [1992], "Fuzzy systems are universal approximators," Proc. IEEE International Conf. on Fuzzy Systems, San Diego, pp. 1163-1170.
[10] Onnen, C., R. Babuska, U. Kaymak, J. Sousa, H. Verbruggen and R Isermann (1997).Genetic algorithms for optimization in predictive control. Control Engineering Practice 5(10), 1363-13

Annex A: MATLAB code of Gustafson-Kessel (GK) Algorithm

```
function [U,V,F] = gk(Z,U0,m,tol) %
Z= [1 2 ; 2 1 ; 2 3 ; 2 1.5 ; 4 3 ; 3 4 ; 4 2 ; 7 9 ; 8 10 ; 8 7 ; 9 8 ; 10
7.5; 9 7 ;8.5 9.5] ;
U0= [1 0 ; 1 0 ; 1 0 ; 1 0 ; 1 0 ; 1 0 ; 1 0 ; 0 1 ; 0 1 ; 0 1; 0 1 ; 0 1 ; 0
1 ; 0 1 ] ;
m= 2
tol=10^(-3)
 [mz,nz] = size(Z);
% data matrix size
c = size(U0,2);% number of clusters
mZ1 = ones(mz,1);% auxiliary variable
nZ1 = ones(nz,1); % auxiliary variable
V1c = ones(1,c);% auxiliary variable
U = zeros(mz,c);% partition matrix
d = U;% distance matrix
F = zeros(c*nz,nz);% covariance matrix
%----------------- iterate ------------------------------
while max(max(U0-U)) > tol % iteration loop
    U = U0; Um = U.^m; sumU = sum(Um); % auxiliary vars
    V = (Um'*Z)./(nZ1*sumU)'; % cluster centers
    for j = 1 : c, % for all clusters
        ZV = Z - mZ1*V(j,:); % auxiliary var
        f = nZ1*Um(:,j)'.*ZV'*ZV/sumU(j); % cov. matrix
        d(:,j)=sum((ZV*(det(f)^(1/nz)*inv(f)).*ZV)')';
    end; d = (d+1e-10).^(-1/(m-1)); % distances
    U0 = (d ./ (sum(d')'*V1c)); % partition matrix
end
%----------------- update final F and U -----------------
U = U0;
Um = U.^m;
sumU = nZ1*sum(Um);
for j = 1 : c,
    ZV = Z - mZ1*V(j,:);
    F((j-1)*nz+(1:nz),:) = nZ1*Um(:,j)'.*ZV'*ZV/sumU(1,j);
end;
%----------------- end of function ----------------------
U
F
V
hold on
plot(Z(:,1),Z(:,2),'rx')
plot(V(:,1),V(:,2),'gx')
figure;
hold off
plot(U(:,1),U(:,2),'rx') figure;
```

Annex B: MATLAB code of C-means Algorithm

```
function [U,V] = ck(Z,U0,m,tol) %
Z= [1 1 ; 1 1.5 ; 1 2.5 ; 1.5 2.5 ; 4 1.5 ; 2 0 ; 4 1 ; 7 5 ; 7 5.5 ; 7.5 6 ;
9 5 ; 9 5.5; 9 4.5 ;8.5 6] ;
U0= [1 0 ; 1 0 ; 1 0 ; 1 0 ; 1 0 ; 1 0 ; 1 0 ; 0 1 ; 0 1 ; 0 1; 0 1 ; 0 1 ; 0
1 ; 0 1 ] ;
m= 2
tol=10^(-3)
 [mz,nz] = size(Z);
% data matrix size
c = size(U0,2);% number of clusters
mZ1 = ones(mz,1);% auxiliary variable
nZ1 = ones(nz,1); % auxiliary variable
V1c = ones(1,c);% auxiliary variable
U = zeros(mz,c);% partition matrix
d = U;% distance matrix
F = zeros(c*nz,nz);% covariance matrix
%------------------ initialize U ------------------
if size(U0,2) == 1,
minZ = V1c'*min(Z); maxZ = V1c'*max(Z);
V = minZ + (maxZ-minZ).*rand(c,nz);
for j = 1 : c,
ZV = Z - mZ1*V(j,:);
d(:,j) = sum((ZV.^2)')';
end;
d = (d+1e-100).^(-1/(m-1));
U0 = (d ./ (sum(d')'*V1c));
end;
%------------------ iterate ------------------------------
while max(max(U0-U)) > tol % iteration loop
    U = U0; Um = U.^m; sumU = sum(Um); % auxiliary vars
    V = (Um'*Z)./(nZ1*sumU)'; % cluster centers
    for j = 1 : c, % for all clusters
        ZV = Z - mZ1*V(j,:); % auxiliary var
        d(:,j)=sum((ZV*(1).*ZV)')';
    end; d = (d+1e-10).^(-1/(m-1)); % distances
    U0 = (d ./ (sum(d')'*V1c)); % partition matrix
end
%------------------ update final F and U -----------------
U = U0;
Um = U.^m;
sumU = nZ1*sum(Um);
%------------------ end of function ----------------------
U
V
hold on
plot(Z(:,1),Z(:,2),'rx')
plot(V(:,1),V(:,2),'gx') figure;holdoff plot(U(:,1),U(:,2),'rx')figure;
```